Rethinking Learning Approaches for Long-Term Action Anticipation

Megha Nawhal[1], Akash Abdu Jyothi[1], and Greg Mori[1,2]

[1] Simon Fraser University, Burnaby, Canada
[2] Borealis AI, Vancouver, Canada

Abstract. Action anticipation involves predicting future actions having observed the initial portion of a video. Typically, the observed video is processed as a whole to obtain a video-level representation of the ongoing activity in the video, which is then used for future prediction. We introduce ANTICIPATR which performs long-term action anticipation leveraging segment-level representations learned using individual segments from different activities, in addition to a video-level representation. We propose a two-stage learning approach to train a novel transformer-based model that uses these two types of representations to directly predict a set of future action instances over any given anticipation duration. Results on Breakfast, 50Salads, Epic-Kitchens-55, and EGTEA Gaze+ datasets demonstrate the effectiveness of our approach.

Keywords: Action Anticipation; Transformer; Long-form videos

1 Introduction

The ability to envision future events is a crucial component of human intelligence which helps in decision making during our interactions with the environment. We are naturally capable of anticipating future events when interacting with the environment in a wide variety of scenarios. Similarly, anticipation capabilities are essential to practical AI systems that operate in complex environments and interact with other agents or humans (*e.g.*, wearable devices [55], human-robot interaction systems [28], autonomous vehicles [36, 62]).

Existing anticipation methods have made considerable progress on the task of near-term action anticipation [9,10,13,14,16,18,37,59] that involves predicting the immediate next action that would occur over the course of a few seconds. While near-term anticipation is a valuable step towards the goal of future prediction in AI systems, going beyond short time-horizon prediction has applicability in a broader range of tasks that involve long-term interactions with the environment. The ability to anticipate actions over long time-horizons is imperative for applications such as efficient planning in robotic systems [8, 15] and intelligent augmented reality systems.

In this paper, we focus on long-term action anticipation. Figure 1 illustrates the problem – having observed an initial portion of an untrimmed activity video, we predict *what* actions would occur *when* in the future.

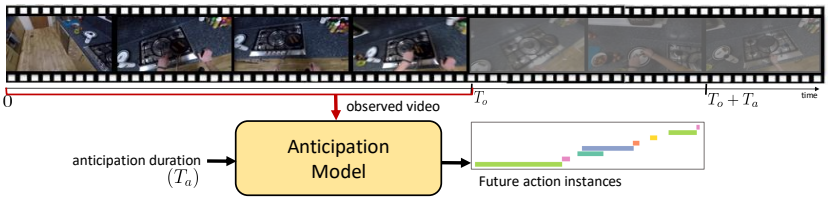


Fig. 1. Long-Term Action Anticipation. Given the initial portion of an activity video $(0, \ldots, T_o)$ and anticipation duration T_a, the task is to predict the actions that would occur from time $T_o + 1$ to $T_o + T_a$. Our proposed anticipation model receives the observed video and the anticipation duration as inputs and directly predicts a set of future action instances. Here, the action anticipation is *long-term* – both the observed duration T_o and the anticipation duration T_a are in the order of minutes.

Long-term anticipation methods [2,12,15,25,51] predict future actions based on the information in the observed video (*i.e.*, an initial portion of an untrimmed activity video) that partially depicts the activity in the video. Current approaches rely on encoding the observed video (input) as a whole to obtain *video-level representations* to perform action anticipation.

We propose a novel approach that leverages segment-level and video-level representations for the task of long-term action anticipation. Consider the example in Figure 1. The video depicts the activity *person making pasta* spanning several minutes. This activity has segments with actions such as *slice onion*, *put pesto*, *put courgette*, *add cheese*. One of these segments such as *put pesto* tends to co-occur with actions involving objects such as *courgette*, *onion*, or *cheese* in a specific order. However, other videos with a different activity, say, *person making pizza*, could potentially have a similar set and/or sequence of actions in a different kitchen scenario. As such, while a specific sequence of actions (*i.e.,* segments of a video) help denote an activity, an individual video segment (containing a single action) alone contains valuable information for predicting the future. Based on this intuition, we introduce an approach that leverages segment-level representations in conjunction with video-level representations for the task of long-term action anticipation. In so doing, our approach enables reasoning beyond the limited context of the input video sequence.

In this work, we propose ANTICIPATR that consists of a two-stage learning approach employed to train a transformer-based model for long-term anticipation (see Fig. 2 for an overview). In the first stage, we train a *segment encoder* to learn segment-level representations. As we focus on action anticipation, we design this training task based on co-occurrences of actions. Specifically, we train the segment encoder to learn *which future actions are likely to occur after a given segment*? Intuitively, consider a video segment showing a pizza pan being moved towards a microwave. Irrespective of the ongoing activity in the video that contains this segment, it is easy to anticipate that certain actions such as *open microwave*, *put pizza* and *close microwave* are more likely to follow than the actions *wash spoon* or *close tap*.

In the second stage, we utilize both the segment-level and video-level representations for long-term action anticipation. We design a transformer-based model that contains two encoders: (1) the segment encoder to derive representations corresponding to segments in the observed video, and (2) a *video encoder* to derive the video-level representations of the observed video. These encoded representations are then fed into an *anticipation decoder* that predicts actions that would occur in the future. Our model is designed to directly predict a set of future action instances, wherein, each element of the set (*i.e.*, an action instance) contains the start and end timestamps of the instance along with the action label. Using direct set prediction, our approach predicts the actions at all the timestamps over a given anticipation duration in a single forward pass.

To summarize, this paper makes the following contributions: (1) a novel learning approach for long-term action anticipation that leverages segment-level representations and video-level representations of the observed video, (2) a novel transformer-based model that receives a video and anticipation duration as inputs to predict future actions over the specified anticipation duration, (3) a direct set prediction formulation that enables single-pass prediction of actions, and (4) state-of-the-art performance on a diverse set of anticipation benchmarks: Breakfast [29], 50Salads [56], Epic-Kitchens-55 [9], and EGTEA Gaze+ [31]. Code is available at https://github.com/Nmegha2601/anticipatr

Overall, our work highlights the benefits of learning representations that capture different aspects of a video, and particularly demonstrates the value of such representations for action anticipation.

2 Related Work

Action Anticipation. Action anticipation is generally described as the prediction of actions before they occur. Prior research efforts have used various formulations of this problem depending on three variables: (1) anticipation format, *i.e.*, representation format of predicted actions, (2) anticipation duration, *i.e.*, duration over which actions are anticipated, and (3) model architectures.

Current approaches span a wide variety of anticipation formats involving different representations of prediction outcomes. They range from pixel-level representations such as frames or segmentations [5, 33, 34, 39] and human trajectories [3, 10, 20, 24, 27, 38] to label-level representations such as action labels [12, 13, 14, 16, 25, 30, 46, 48, 49, 51, 53, 59, 64, 65] or temporal occurrences of actions [2, 15, 32, 37, 40, 57] through to semantic representations such as affordances [28] and language descriptions of sub-activities [52]. We focus on label-level anticipation format and use 'action anticipation' to refer to this task.

Existing anticipation tasks can be grouped into two categories based on the anticipation duration: (1) near-term action anticipation, and (2) long-term action anticipation. In this paper, we focus on long-term action anticipation.

Near-term anticipation involves predicting label for the immediate next action that would occur in the range of a few seconds having observed a short video segment of duration of a few seconds. Prior work propose a variety of

temporal modeling techniques to encode the observed segment such as regression networks [59], reinforced encoder-decoder network [16], TCNs [63], temporal segment network [9], LSTMs [13,14,45], VAEs [40,61] and transformers [18].

Long-term anticipation involves predicting action labels over long time-horizons in the range of several minutes having observed an initial portion of a video (observed duration of a few minutes). A popular formulation of this task involves prediction of a sequence of action labels having observed an initial portion of the video. Prior approaches encode the observed video as a whole to obtain a video-level representation. Using these representations, these approaches either predict actions recursively over individual future time instants or use time as a conditional parameter to predict action label for the given single time instant. The recursive methods [2,12,15,46,51] accumulate prediction error over time resulting in inaccurate anticipation outcomes for scenarios with long anticipation duration. The time-conditioned method [25] employs skip-connections based temporal models and aims to avoid error accumulation by directly predicting an action label for a specified future time instant in a single forward pass. However, this approach still requires multiple forward passes during inference as the task involves predicting actions at all future time instants over a given anticipation duration. Additionally, sparse skip connections used in [25] do not fully utilize the relations among the actions at intermediate future time instants while predicting action at a given future time instant. In contrast to these approaches based on video-level representations, our approach leverages segment-level representations (learned using individual segments across different activities) in conjunction with video-level representations. Both these representations are utilized to directly predict action instances corresponding to actions at all the time instants over a given anticipation duration in a single forward pass.

An alternate formulation of long-term anticipation proposed in [42] focuses on predicting a set of future action labels without inferring when they would occur. [42] extracts a graph representation of the video based on frame-level visual affordances and uses graph convolutional network to encode the graph representation to predict a set of action labels. In contrast, our approach leverages both the segment-level and video-level representations of the input video and a transformer-based model to predict action instances - both action labels and their corresponding timestamps.

Other methods design approaches to model uncertainty in predicting actions over long time horizons [1,44,46] and self-supervised learning [47].

Early action detection. The task of early action detection [21,35,50,54] involves recognizing an ongoing action in a video as early as possible given an initial portion of the video. Though the early action detection task is different from action anticipation (anticipation involves prediction of actions *before* they begin), the two tasks share the inspiration of future prediction.

Transformers in computer vision. The transformer architecture [58], originally proposed for machine translation task, has achieved state-of-the-art performance for many NLP tasks. In recent years, there has been a flurry of work on transformer architectures designed for high-level reasoning tasks on images and

videos. Examples include object detection [6], image classification [11], spatio-temporal localization in videos [17], video instance segmentation [60], action recognition [4, 66], action detection [43], multi-object tracking [41], next action anticipation [18], human-object interaction detection [26, 67]. DETR [6] is a transformer model for object detection, wherein, the task is formulated as a set prediction problem. This work has since inspired transformer designs for similar vision tasks – video instance segmentation [60] and human-object interaction detection [67]. Inspired by these works, we propose a novel transformer architecture that uses two encoder to encode different representations derived from the input video and a decoder to predict the set of future action instances in a single pass. Our proposed decoder also receives anticipation duration as an input parameter to control the duration over which actions are predicted.

3 Action Anticipation with ANTICIPATR

In this section, we first describe our formulation of long-term action anticipation and then describe our approach.

Problem Formulation. Let $\mathbf{v}_o$ be an observed video containing T_o frames. Our goal is to predict the actions that occur from time T_o+1 to T_o+T_a where T_a is the anticipation duration, *i.e.*, the duration over which actions are predicted. Specifically, we predict a set $\mathcal{A} = \{a^i = (c^i, t_s^i, t_e^i)\}$ containing future action instances. The i-th element denotes an action instance a^i depicting action category c^i occurring from time t_s^i to t_e^i where $T_o < t_s^i < t_e^i \leq T_o + T_a$. Here, $c^i \in \mathcal{C}$ where $\mathcal{C}$ is the set of action classes in the dataset.

Intuitively, for action anticipation, the observed video as a whole helps provide a broad, video-level representation of the ongoing activity depicted in the video. However, the observed video is composed of several segments that individually also contain valuable information about future actions and provide an opportunity to capture the video with segment-level representations. Using this intuition, in this paper, we propose ANTICIPATR that leverages these two types of representations of the observed video for the task of long-term anticipation.

ANTICIPATR employs a two-stage learning approach to train a transformer-based model that takes an observed video as input and produces a set of future action instances as output. See Fig. 2 for an overview. In the first stage, we train a *segment encoder* that receives a segment (sequence of frames from a video) as input and predicts the set of action labels that would occur at any time in the future after the occurrence of the segment in the video. We refer to this stage as segment-level training (described in Sec. 3.1). As the segment encoder only operates over individual segments, it is unaware of the broader context of the activity induced by a specific sequence of segments in the observed video.

In the second stage, we train a *video encoder* and an *anticipation decoder* to be used along with the segment encoder for long-term action anticipation. The video encoder encodes the observed video to a video-level representation. The segment encoder (trained in the first stage) is fed with a sequence of segments from the observed video as input to obtain a segment-level representation of

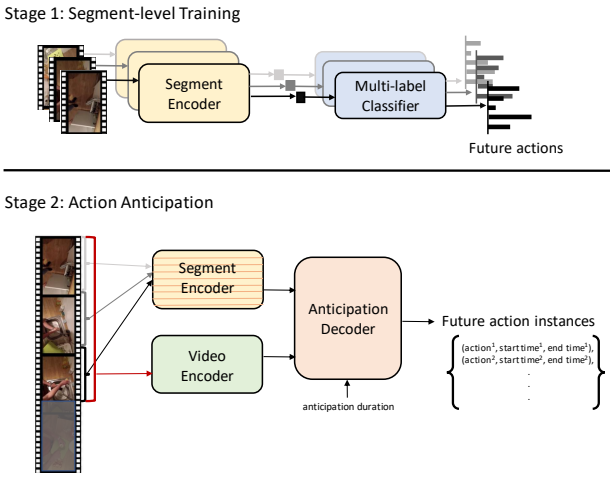


Fig. 2. Learning Approach. ANTICIPATR uses a two-stage learning approach. In the first stage, we perform segment-level training (refer to Sec 3.1). Given a segment as input, we train a segment encoder to predict the set of action labels that would occur at any time after the occurrence of the segment in the activity video. In the second stage, we perform long-term action anticipation (refer to Sec 3.2). We use video encoder to obtain video-level representation and segment encoder (trained in the first stage) is used to obtain segment-level representation. The anticipation decoder receives these two representations of the observed video to directly predict a set of action instances that would occur in the future over a given anticipation duration.

the video. The anticipation decoder receives the two representations along with the anticipation duration to predict a set of future action instances over the given anticipation duration in a single pass. The video encoder and anticipation decoder are trained using classification losses on the action labels and two temporal losses (L_1 loss and temporal IoU loss) on the timestamps while the segment encoder is kept unchanged. We refer to this second stage of training as action anticipation (see Sec. 3.2).

3.1 Stage 1: Segment-level Training

In this stage, the segment encoder is trained on a segment-level prediction task to learn representations for individual segments. See Fig. 3 (*left*) for an overview.
Segment Encoder. We design the segment encoder network E_s as a sequence of ℓ_s transformer blocks containing a multi-head self-attention module followed by layernorm and a feed forward network [58]. This network is trained on the task of segment-level action anticipation.
Training. During training, the segment encoder receives a segment (sequence of frames from a video) as input and predicts the set of action labels that would occur at any time in the future (starting from the temporal boundary, *i.e.,* end of the segment until the end of that video) without inferring when they would

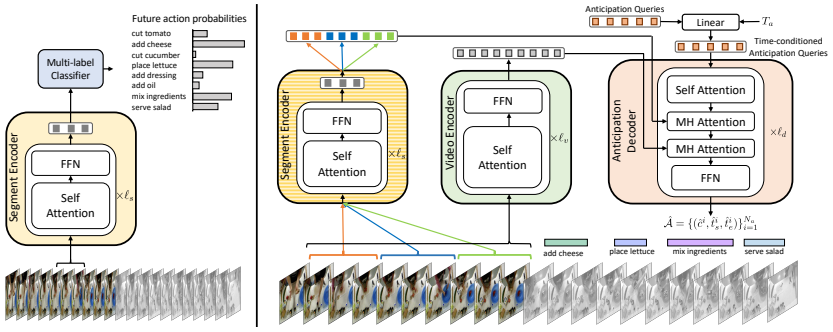


Fig. 3. Model Architecture. Our model comprises three networks: *segment encoder*, *video encoder* and *anticipation decoder* and is trained for long-term action anticipation in two stages. (*left*) Segment-level training (Sec. 3.1): The segment encoder receives a segment as input and predicts a set of action labels that would occur at any time in the future (after the occurrence of segment in the video). (*right*) Action Anticipation (Sec. 3.2): The video encoder encodes the observed video to a video-level representation. Concurrently, the video is divided into a sequence of segments and each segment is fed into the segment encoder (trained in first stage)The anticipation decoder receives the two representations along with an anticipation duration as inputs to directly predict a set of future action instances over the given anticipation duration. [MH Attention: Multi-head Attention, FFN: Feed Forward Network.]

occur. Depending on the segment, there could be multiple actions occurring between the end of segment and end of video. Thus, we formulate this training task as a multi-class multi-label classification.

The training data for the segment encoder is derived from the training set in the original video dataset containing videos with action annotations. These input segments are obtained using the action boundaries provided in the training set. We do not require any additional annotations. Formally, given a video $\mathbf{v}$ containing T frames, a segment $\mathbf{v}_s^{(t',t'')}$, spanning time indices t' to t'' where $0 \leq t' < t'' < T$, is taken as input. For this segment, the target is a binary vector $\mathbf{c}_s$ (dimension $|\mathcal{C}|$) corresponding to the action labels that occur after the temporal boundary of the segment until the end of the video ($[\mathbf{v}^{t''+1}, \ldots, \mathbf{v}^{T}]$).

The segment encoder E_s receives the segment $\mathbf{v}_s^{(t',t'')}$ along with positional encodings $\mathbf{p}_s^{(t',t'')}$ (details in supplementary). The output of the encoder is an embedding $\mathbf{h} = [\mathbf{h}^1, \ldots, \mathbf{h}^{t''-t'+1}]$ of dimension $(t'' - t' + 1) \times d_s$ where d_s is the channel dimension. The output embeddings are then averaged along time dimension and fed into a linear layer F followed by a sigmoid activation σ to obtain future action probabilities $\hat{\mathbf{c}}_s$ of dimension $|\mathcal{C}|$, expressed as:

$$\begin{aligned} \mathbf{h} &= E_s\left(\mathbf{v}_s^{(t',t'')}, \mathbf{p}_s^{(t',t'')}\right) \\ \hat{\mathbf{c}}_s &= \sigma\left(F\left(\frac{1}{t''-t'+1}\sum_{i=1}^{t''-t'+1}\mathbf{h}^i\right)\right). \end{aligned} \tag{1}$$

Here, $\hat{\mathbf{c}}_s$ is the output of a multi-label classifier where each element c_s^j of $\hat{\mathbf{c}}_s$ denotes probability of corresponding action category $j \in \mathcal{C}$. This network is trained using binary cross entropy loss between the prediction vector $\hat{\mathbf{c}}_s$ and target vector $\mathbf{c}_s$. Once trained, the linear layer F is discarded and the segment encoder E_s is used to obtain segment-level representations for the action anticipation stage.

3.2 Stage 2: Action Anticipation

In the second stage of our approach, we use an encoder-decoder model that contains two encoders: (i) the segment encoder from the first stage, and (ii) a video encoder that encodes the observed video as a whole. The outputs of these two encoders along with an anticipation duration are fed into an anticipation decoder which uses the representations from the two encoders to predict a set of future action instances over the given anticipation duration. See Fig. 3 (*right*).

Video Encoder. The video encoder receives an observed video containing T_o frames. We denote the input as $\mathbf{v}_o = [\mathbf{v}^1, \ldots, \mathbf{v}^{T_o}]$. We design the encoder network E_v as a sequence of ℓ_v transformer blocks [58] containing a multi-head self-attention module followed by layernorm and feed forward network. The encoder receives the features corresponding to the observed video $\mathbf{v}_o$ as input. As the self-attention module is permutation-invariant, we provide additional information about the sequence in the form of sinusoidal positional encodings [58] $\mathbf{p}_o = [\mathbf{p}^1, \ldots, \mathbf{p}^{T_o}]$ (see supplementary for additional explanation). Here, each element in the positional encoding sequence is added to the corresponding element in the video features and then fed into the encoder block. The encoder models temporal relationships in the observed video and transforms the input sequence to a contextual representation $\mathbf{h}_v = [\mathbf{h}_v^1, \ldots, \mathbf{h}_v^{T_o}]$, expressed as:

$$\mathbf{h}_v = E_v(\mathbf{v}_o, \mathbf{p}_o). \tag{2}$$

Encoding Video Segments. Concurrent to the video encoder, the input video is divided into a sequence of segments using temporal sliding windows. Specifically, a temporal window of size k starting from frame index i obtains a segment $[\mathbf{v}^i, \ldots, \mathbf{v}^{i+k-1}]$, which is fed to the segment encoder to obtain the outputs $\mathbf{h}_s^i, \ldots, \mathbf{h}_s^{i+k-1}$. The starting index i slides across time with $i \in \{1, k+1, 2k+1, \ldots, (T_o - k + 1)\}$ generating the temporal windows, where the window size k is a hyperparameter. The outputs of the segment encoder for all temporal windows are concatenated to obtain $\mathbf{h}_s = [\mathbf{h}_s^1, \ldots, \mathbf{h}_s^{T_o}]$. During implementation, the representations can still be obtained in one forward pass of the segment encoder by stacking segments along the batch dimension of the input. This segment-level representation of the video is complementary to the video-level representation that encodes the ongoing activity in the video.

Anticipation Decoder. Given the video-level and the segment-level representations, the decoder aims to predict a set of future action instances over a given anticipation duration. The predicted set contains action instances of the form (label, start time, end time). The anticipation decoder receives the following inputs: (i) *anticipation queries* $\mathbf{q}_0$, (ii) anticipation duration T_a over which actions

are to be predicted, (iii) encoded representation $\mathbf{h}_v$ from video encoder E_v, and (iv) encoded representation $\mathbf{h}_s$ from segment encoder E_s.

The anticipation queries contain N_a elements, *i.e.*, $\mathbf{q}_0 = [\mathbf{q}_0^1, \ldots, \mathbf{q}_0^{N_a}]$, wherein each query is a learnable positional encoding (more details in supplementary). We consider N_a as a hyperparameter that is constant for a dataset and is sufficiently larger than the maximum number of action instances to be anticipated per video in the overall dataset. Each query $\mathbf{q}_0^i$ is then fed into a linear layer (weights shared for all values of i) along with the anticipation duration T_a to obtain time-conditioned anticipation queries $\mathbf{q}_a^i$ for $i = 1, \ldots, N_a$. This time conditioning enables the anticipation decoder to predict actions over any specified anticipation duration.

The decoder network D consists of ℓ_d blocks, wherein, each block contains a cascade of attention layers. The first attention layer is the multi-head self-attention block which models relations among the anticipation queries. The second attention layer is a multi-head encoder-decoder attention layer that maps the queries and the segment-level representations from the segment encoder. And, the third attention layer is another multi-head encoder-decoder attention layer that maps the output of previous layer to the video-level representation corresponding to the input. This third attention layer is followed by a feedforward network. The output of the decoder $\mathbf{y} = [\mathbf{y}^1, \ldots, \mathbf{y}^{N_a}]$ serves as a latent representation of the action instances in the videos, expressed as:

$$\mathbf{y} = D(\mathbf{q}_a, \mathbf{h}_v, \mathbf{h}_s) \tag{3}$$

The decoder output is used to predict the set of action instances $\hat{\mathcal{A}} = \{\hat{a}^i = (\hat{c}^i, \hat{t}_s^i, \hat{t}_e^i)\}_{i=1}^{N_a}$. Each element in decoder output $\mathbf{y}^i$ is fed into a linear layer followed by softmax to obtain prediction probabilities $\hat{p}^i(c)$ where $c = 1, \ldots, |\mathcal{C}|+1$ and $\hat{c}^i$ is the class corresponding to maximum probability. The number of queries N_a is larger than the maximum number of action instances per video in the dataset. Thus, we introduce an additional class label $\varnothing$ indicating no action. $\mathbf{y}^i$ is also fed into another feedforward network with ReLU to obtain corresponding start timestamps $\hat{t}_s^i$ and end timestamps $\hat{t}_e^i$.

Training. To compute the loss, we first align the predictions with the groundtruth set of action instances. This alignment is necessary as there is no fixed prior correspondence between the predicted and the groundtruth set of action instances. Here, the predicted set for any video contains N_a action instances, but the size of groundtruth set $\mathcal{A}$ varies based on the video and is smaller than the predicted set. Thus, we first pad the groundtruth set to make it the same size as the predicted set by adding $N_a - |\mathcal{A}|$ elements with label $\varnothing$ indicating no action. Then, we use a pair-wise greedy correspondence algorithm to align the groundtruth and predicted sets. Starting with the groundtruth instance having the longest duration, we match each groundtruth instance with the unmatched predicted instance that has the maximum temporal overlap with the groundtruth instance. This results in a one-to-one mapping for loss computation (more details in supplementary).

Consider the output of the set correspondence module as γ denoting the permutation of the predicted set of instances, *i.e.*, the groundtruth action instance

a^i is matched to predicted instance $\hat{a}^{\gamma(i)}$ for $i = 1, \ldots, N_a$. Given this alignment, we compute loss $\mathcal{L}$ over all the matched pairs as a weighted combination of cross-entropy loss for classification, and two temporal losses: $L1$ loss and IoU loss ($\mathcal{L}_{iou}$) for prediction of segment timestamps, defined as:

$$\mathcal{L} = \sum_{i=1}^{N_a} \Big[-\log(\hat{p}^{\gamma(i)}(c^i)) + \mathbb{1}_{\{c^i \neq \varnothing\}} \lambda_{L1} ||s^i - \hat{s}^{\gamma(i)}||_1 + \mathbb{1}_{\{c^i \neq \varnothing\}} \lambda_{iou} \mathcal{L}_{iou}(s^i, \hat{s}^{\gamma(i)}) \Big], \tag{4}$$

where $\lambda_{iou}, \lambda_{L1} \in \mathbb{R}^+$ are hyperparameters, $s^i = [t_s^i, t_e^i]$, $\hat{s}^{\gamma(i)} = [\hat{t}_s^{\gamma(i)}, \hat{t}_e^{\gamma(i)}]$ and $\hat{p}^{\gamma(i)}(c^i)$ is the probability of the groundtruth class c^i for prediction $\gamma(i)$. The video encoder and anticipation decoder are jointly trained to minimize this loss. We do not fine-tune the segment encoder in this stage.

Inference. During inference, the video encoder takes the observed video as input and the segment encoder takes the chunked video (*i.e.*, non-overlapping segments of fixed length) as input. The inputs to the decoder are: (i) anticipation queries $\mathbf{q}_0 = 1, \ldots, N_a$ (a constant, regardless of input), (ii) anticipation duration T_a (varies based on the input video and the anticipation requirement), (iii) output representation from the video encoder, and (iv) output representation from the segment encoder. The decoder predicts a set of action instances. Thus, our approach allows us to build a model that can anticipate actions over any future duration in a single pass by simply controlling the input T_a to the decoder as shown by results in Table 1.

In summary, ANTICIPATR uses a two-stage learning approach to train a transformer-based model (consisting of two encoders and one decoder) to predict a set of future action instances over any given anticipation duration. Our approach aims to perform action anticipation with segment-level representations learned using individual video segments in conjunction with video-level representations learned by encoding input video as a whole. Our model anticipates actions at all time instants over a given anticipation duration in a single forward pass by directly predicting a set of future action instances.

4 Experiments

We conducted extensive experiments and analysis to demonstrate the effectiveness of our proposed approach.

Datasets. We evaluate on four established benchmarks for this task. These datasets of untrimmed videos vary in scale, diversity of labels and video duration.

Breakfast [29] contains 1,712 videos each depicting one of 10 breakfast activities and annotated with action instances spanning 48 different action classes. On average, a video contains 6 action instances and has a duration of 2.3 minutes. For evaluation, we report the average across 4 splits from the original dataset.

50Salads [56] contains 50 videos, each showing a person preparing a salad. On average, there are 20 action instances per video spanning 17 action classes

Table 1. Results (Breakfast and 50Salads). We report the mean over classes accuracy for different observation/anticipation durations. Higher values indicate better performance. Note that "Sener *et al.* [51] (features+labels)" use action labels from a segmentation algorithm as additional input. Baseline results are from respective papers.

	Observation (β_o) →	20%				30%			
	Anticipation (β_a) →	10%	20%	30%	50%	10%	20%	30%	50%
Breakfast	RNN [2]	18.1	17.2	15.9	15.8	21.6	20.0	19.7	19.2
	CNN [2]	17.9	16.3	15.3	14.5	22.4	20.12	19.7	18.7
	RNN [2] + TCN	5.9	5.6	5.5	5.1	8.9	8.9	7.6	7.7
	CNN [2] + TCN	9.8	9.2	9.1	8.9	17.6	17.1	16.1	14.4
	Ke *et al.* [25]	18.4	17.2	16.4	15.8	22.7	20.4	19.6	19.7
	Farha *et al.* [12]	25.9	23.4	22.4	21.5	29.7	27.4	25.6	25.2
	Qi *et al.* [47]	25.6	21.0	18.5	16.0	27.3	23.6	20.8	17.3
	Sener *et al.* [51] (features)	24.2	21.1	20.0	18.1	30.4	26.3	23.8	21.2
	Sener *et al.* [51] (features+labels)	**37.4**	31.8	30.1	27.1	39.8	34.2	31.9	27.9
	ANTICIPATR **(Ours)**	**37.4**	**32.0**	**30.3**	**28.6**	**39.9**	**35.7**	**32.1**	**29.4**
50Salads	RNN [2]	30.1	25.4	18.7	13.5	30.8	17.2	14.8	9.8
	CNN [2]	21.2	19.0	15.9	9.8	29.1	20.1	17.5	10.9
	RNN [2] + TCN	32.3	25.5	19.1	14.1	26.1	17.7	16.3	12.9
	CNN [2] + TCN	16.0	14.7	12.1	9.9	19.2	14.7	13.2	11.2
	Ke *et al.* [25]	32.5	27.6	21.3	15.9	35.1	27.1	22.1	15.6
	Farha *et al.* [12]	34.8	28.4	21.8	15.2	34.4	23.7	18.9	15.9
	Sener *et al.* [51](features)	25.5	19.9	18.2	15.1	30.6	22.5	19.1	11.2
	Sener *et al.* [51](features+labels)	34.7	26.3	23.7	15.7	34.5	26.1	22.7	17.1
	Qi *et al.* [47]	37.9	28.8	21.3	11.1	37.5	24.1	17.1	09.1
	Piergiovanni *et al.* [46]	40.4	33.7	25.4	20.9	40.7	40.1	26.4	19.2
	ANTICIPATR **(Ours)**	**41.1**	**35.0**	**27.6**	**27.3**	**42.8**	**42.3**	**28.5**	**23.6**

and duration is 6.4 minutes. Following the original dataset, we report the average across 5-fold cross-validation in our evaluation.

EGTEA Gaze+ (EGTEA+) [31] contains egocentric videos of 32 subjects following 7 recipes in a single kitchen. Each video depicts the preparation of a single dish. Each video is annotated with instances depicting interactions (*e.g.*, open drawer), spanning 53 objects and 19 actions.

EPIC-Kitchens-55 (EK-55) [9] contains videos of daily kitchen activities. It is annotated for interactions spanning 352 objects and 125 actions. It is larger than the aforementioned datasets, and contains unscripted activities.

We represent the input videos by feature representations used in the benchmarks (see supplementary for details).

Evaluation. To measure the performance of our model, we adopt the evaluation protocol followed by state-of-the-art methods for these benchmark datasets.

For Breakfast and 50Salads, we report the mean over classes accuracy averaged over all future timestamps in the specified anticipation duration, *i.e.*, dense prediction evaluation as defined in [2, 12, 25]. We use β_o% of a full video as observation duration and predict the actions corresponding to following β_a% of the remaining video. As per the benchmarks, we sweep the values of $\beta_o \in \{20, 30\}$ and $\beta_a \in \{10, 20, 30, 50\}$ denoting different observation and anticipation durations respectively. Note that a single trained model is used for predicting at all these values of β_o and β_a by just varying the anticipation duration input to the decoder. Since the metric is computed over a dense anticipation timeline, we

Table 2. Results (EK-55 and EGTEA+). We report mAP values for ALL classes, FREQUENT classes (> 100 action instances) and RARE class (< 10 action instances). Following [42], we report the mAP values averaged over different observation durations. Higher values implies better performance. Baseline results are from respective papers.

Method	EK-55			EGTEA+		
	ALL	FREQ	RARE	ALL	FREQ	RARE
RNN	32.6	52.3	23.3	70.4	76.6	54.3
I3D [7]	32.7	53.3	23.0	72.1	79.3	53.3
ActionVLAD [19]	29.8	53.5	18.6	73.3	79.0	58.6
Timeception [22]	35.6	55.9	26.1	74.1	79.7	59.7
VideoGraph [23]	22.5	49.4	14.0	67.7	77.1	47.2
EGO-TOPO [42]	38.0	56.9	**29.2**	73.5	80.7	54.7
ANTICIPATR**(Ours)**	**39.1**	**58.1**	29.1	**76.8**	**83.3**	**55.1**

first convert our model predictions (set of action instances) into a timeline and then compute mean over classes accuracy (details in supplementary).

For EK-55 and EGTEA+, we compute a multi-label classification metric (mAP) over the target action classes as defined in [42]. α_o% of each untrimmed video is given as input to predict all action classes in the future $(100 - \alpha_o)$% of the video, *i.e.*, until the end of the video. We sweep values of $\alpha_o \in \{25, 50, 75\}$ representing different observation durations. Since the metric is computed only over the future action classes, we take the union of the class labels of predicted action instances to compute mAP.

Comparison with state-of-the-art. Table 1 shows the results for Breakfast and 50Salads datasets in the *'no groundtruth labels'* setting [25, 51]. The results show that our approach outperforms existing methods by a considerable margin for different observation/anticipation durations. For these benchmarks, the most similar approach to ours is Sener *et al.* [51] where they propose self-attention methods for temporal aggregation for long-term video modeling. In the setting similar to ours where they use only visual features as input, our approach outperforms [51] with up to 13% improvement. Moreover, when they also use action labels from a segmentation algorithm as input, our approach is still competitive despite not using such additional inputs. In addition, the benefit of our approach is more apparent when the anticipation duration is longer.

Table 2 shows results on the long-term action anticipation benchmarks for EK-55 and EGTEA+ datasets, as defined by [42]. The results show that our model achieves competitive results with the state-of-the-art method [42]. While this benchmark only considers prediction of future action labels, our results demonstrate that the segment prediction in our model acts as a beneficial auxiliary task for label prediction.

Impact of Segment-level Training. Our two-stage learning approach separately learns video-level representations and segment-level representations. To analyze the impact of such two-stage training, we design following experiments.

(i) **Fine-tuned Segment Encoder.** In this experiment, we also fine-tune the segment encoder while training video encoder and decoder during the anticipation stage (Sec 3.2). The results in Fig. 4 ('Fine-tuned SE') indicate that

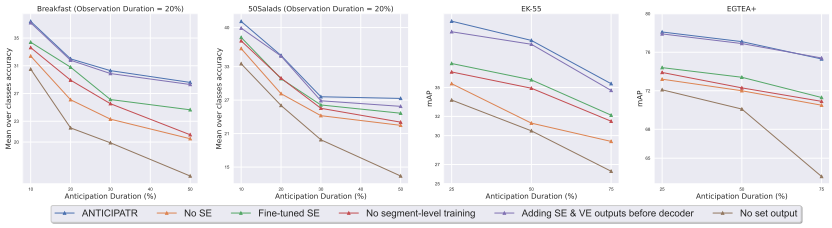


Fig. 4. Analysis. Quantitative evaluation of the anticipation performance of ablated versions of ANTICIPATR. [SE: segment encoder; VE: video encoder].

fine-tuning the segment encoder hurts the anticipation performance. We believe fine-tuning the segment encoder with anticipation loss (Eq. 4) perturbs the segment-level representation learned during first stage of training.

(ii) **No Segment-level Training.** In this experiment, we do not train the segment encoder network in a separate stage. Instead, we train all three networks (*i.e.*, segment encoder, video encoder and anticipation decoder) jointly for the task of long-term action anticipation using the anticipation loss function (Eq. 4). Here, the segment encoder receives videos chunked into short segments (same as the proposed two-stage training). However, it is directly tasked with solving a more difficult problem of simultaneously encoding segment-level representation and inferring its usage for long-term anticipation. The results for all datasets presented in Fig. 4 ('No Segment-level Training') illustrate that eliminating training of the segment encoder worsens the anticipation performance. This shows the value of learning the segment-level representations independently without being influenced by the overall activity in the input video.

In summary, these experiments demonstrate the importance of the two-stage learning approach and suggest that the two representations should be learned separately to serve their individual purposes during anticipation.

Impact of Segment Encoder. To evaluate the impact of learning segment-level representation, we conducted experiments without the segment encoder network. This ablated version only contains the video encoder and the anticipation decoder and is trained in a single-stage using the anticipation loss (Eq. 4). The results in Fig. 4 ('No SE') show that removing the segment-level representations considerably hurts the anticipation performance. This performance degradation is worse than just removing the segment-level training stage ('No segment-level training' in Fig. 4). Thus, this experiment validates the benefit of the segment-level stream of information for action anticipation.

Impact of Set-based Output Representation. In our approach, we model the anticipation output as a set of action instances. We empirically validate this design by comparing with an alternative approach where the output is a sequence of action labels corresponding to the individual future time instants. We implement this by changing the anticipation queries (decoder input) during the anticipation stage – we provide positional encodings corresponding to each time instant over anticipation duration and directly predict the labels corresponding

Fig. 5. Visualizations from Breakfast (left) and 50salads (right) where 20% of the video is observed and actions are anticipated over 50% of the remaining video.

to these time instants. While the prediction for all time instants still happens in a single pass, the decoder is required to transform a large number of anticipation queries. The results in Fig. 4 ('No Set Output') show poor performance that worsen further as anticipation duration increases. This is largely because the number of queries is too high for the decoder for effective modeling.

Fusion of Encoder Outputs. To combine the representation from segment encoder and video encoder, our model uses two encoder-decoder attention layers in the decoder blocks. We tested an alternative approach wherein we fused the representations using a simple addition along temporal dimension before feeding into the decoder. Here, we modify the decoder blocks to contain a single encoder-decoder attention layer. The results in Fig. 4 ('Adding SE & VE before decoder') indicate that this fusion approach leads to a slight decrease in anticipation performance. We believe adding the representations before decoder forces the computation of encoder-decoder attention weights by considering both information streams at once. In contrast, our ANTICIPATR approach of computing attention one-by-one enables it to first filter out the relevant information from segment-level representations learned across different activities and then contextualize them into the specific context of the input video.

Visualizations. The examples in Fig. 5 shows that our model effectively anticipates future actions. Please refer to supplementary material for additional visualizations and analysis of failure cases.

5 Conclusion

We introduced a novel approach for long-term action anticipation to leverage segment-level representations learned from individual segments across different activities in conjunction with a video-level representation that encodes the observed video as a whole. We proposed a novel two-stage learning approach to train a transformer-based model that receives a video and an anticipation duration as inputs and predicts a set of future action instances over the given anticipation duration. Results showed that our approach achieves state-of-the-art performance on long-term action anticipation benchmarks for Breakfast, 50Salads, Epic-Kitchens-55, and EGTEA Gaze+ datasets. Overall, our work highlights the benefits of learning representations that capture information across different activities for action anticipation.

References

1. Abu Farha, Y., Gall, J.: Uncertainty-aware anticipation of activities. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
2. Abu Farha, Y., Richard, A., Gall, J.: When will you do what?-anticipating temporal occurrences of activities. In: Proceedings of the IEEE International Conference on Computer Vision (CVPR) (2018)
3. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
4. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
5. Bhattacharyya, A., Fritz, M., Schiele, B.: Bayesian prediction of future street scenes using synthetic likelihoods. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019)
6. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
7. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
8. Chang, C.Y., Huang, D.A., Xu, D., Adeli, E., Fei-Fei, L., Niebles, J.C.: Procedure planning in instructional videos. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
9. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Scaling egocentric vision: The epic-kitchens dataset. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
10. Dang, L., Nie, Y., Long, C., Zhang, Q., Li, G.: Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)
12. Farha, Y.A., Ke, Q., Schiele, B., Gall, J.: Long-term anticipation of activities with cycle consistency. In: Proceedings of the German Conference on Pattern Recognition (GCPR) (2020)
13. Furnari, A., Farinella, G.: Rolling-unrolling lstms for action anticipation from first-person video. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
14. Furnari, A., Farinella, G.M.: What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In: Proceedings of the IEEE International Conference on Computer Vision (CVPR) (2019)
15. Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Forecasting future action sequences with neural memory networks. Proceedings of the British Machine Vision Conference (BMVC) (2019)

16. Gao, J., Yang, Z., Nevatia, R.: Red: Reinforced encoder-decoder networks for action anticipation. Proceedings of the British Machine Vision Conference (BMVC) (2017)
17. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
18. Girdhar, R., Grauman, K.: Anticipative video transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
19. Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: Actionvlad: Learning spatio-temporal aggregation for action classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
20. Hernandez, A., Gall, J., Moreno-Noguer, F.: Human motion prediction via spatio-temporal inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
21. Hoai, M., De la Torre, F.: Max-margin early event detectors. International Journal of Computer Vision (IJCV) (2014)
22. Hussein, N., Gavves, E., Smeulders, A.W.: Timeception for complex action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
23. Hussein, N., Gavves, E., Smeulders, A.W.: Videograph: Recognizing minutes-long human activities in videos. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshop (2019)
24. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
25. Ke, Q., Fritz, M., Schiele, B.: Time-conditioned action anticipation in one shot. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
26. Kim, B., Lee, J., Kang, J., Kim, E.S., Kim, H.J.: Hotr: End-to-end human-object interaction detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
27. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: Proceedings of the European Conference on Computer Vision (ECCV) (2012)
28. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2015)
29. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
30. Lan, T., Chen, T.C., Savarese, S.: A hierarchical representation for future action prediction. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014)
31. Li, Y., Liu, M., Rehg, J.M.: In the eye of beholder: Joint learning of gaze and actions in first person video. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
32. Liang, J., Jiang, L., Niebles, J.C., Hauptmann, A.G., Fei-Fei, L.: Peeking into the future: Predicting future person activities and locations in videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

33. Liang, X., Lee, L., Dai, W., Xing, E.P.: Dual motion gan for future-flow embedded video prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2017)
34. Luc, P., Neverova, N., Couprie, C., Verbeek, J., LeCun, Y.: Predicting deeper into the future of semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2017)
35. Ma, S., Sigal, L., Sclaroff, S.: Learning activity progression in lstms for activity detection and early detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
36. Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., Manocha, D.: Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2019)
37. Mahmud, T., Hasan, M., Roy-Chowdhury, A.K.: Joint prediction of activity labels and starting times in untrimmed videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2017)
38. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
39. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: Proceedings of the International Conference on Learning Representations (ICLR) (2016)
40. Mehrasa, N., Jyothi, A.A., Durand, T., He, J., Sigal, L., Mori, G.: A variational auto-encoder model for stochastic point processes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
41. Meinhardt, T., Kirillov, A., Leal-Taixe, L., Feichtenhofer, C.: Trackformer: Multi-object tracking with transformers. arXiv preprint arXiv:2101.02702 (2021)
42. Nagarajan, T., Li, Y., Feichtenhofer, C., Grauman, K.: Ego-topo: Environment affordances from egocentric video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
43. Nawhal, M., Mori, G.: Activity graph transformer for temporal action localization. arXiv preprint arXiv:2101.08540 (2021)
44. Ng, Y.B., Fernando, B.: Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting. IEEE Transactions on Image Processing (TIP) (2020)
45. Osman, N., Camporese, G., Coscia, P., Ballan, L.: Slowfast rolling-unrolling lstms for action anticipation in egocentric videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
46. Piergiovanni, A., Angelova, A., Toshev, A., Ryoo, M.S.: Adversarial generative grammars for human activity prediction. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
47. Qi, Z., Wang, S., Su, C., Su, L., Huang, Q., Tian, Q.: Self-regulated learning for egocentric video activity anticipation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
48. Rodin, I., Furnari, A., Mavroeidis, D., Farinella, G.M.: Untrimmed action anticipation. arXiv preprint arXiv:2202.04132 (2022)
49. Rodriguez, C., Fernando, B., Li, H.: Action anticipation by predicting future dynamic images. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)

50. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2011)
51. Sener, F., Singhania, D., Yao, A.: Temporal aggregate representations for long-range video understanding. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
52. Sener, F., Yao, A.: Zero-shot anticipation for instructional activities. In: Proceedings of the IEEE International Conference on Computer Vision (CVPR) (2019)
53. Shi, Y., Fernando, B., Hartley, R.: Action anticipation with rbf kernelized feature mapping rnn. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
54. Shou, Z., Pan, J., Chan, J., Miyazawa, K., Mansour, H., Vetro, A., Giro-i Nieto, X., Chang, S.F.: Online detection of action start in untrimmed, streaming videos. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
55. Soran, B., Farhadi, A., Shapiro, L.: Generating notifications for missing actions: Don't forget to turn the lights off! In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2015)
56. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (2013)
57. Sun, C., Shrivastava, A., Vondrick, C., Sukthankar, R., Murphy, K., Schmid, C.: Relational action forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
58. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS) (2017)
59. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating visual representations from unlabeled video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
60. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-end video instance segmentation with transformers. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
61. Wu, B., Nair, S., Martin-Martin, R., Fei-Fei, L., Finn, C.: Greedy hierarchical variational autoencoders for large-scale video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
62. Yu, C., Ma, X., Ren, J., Zhao, H., Yi, S.: Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
63. Zatsarynna, O., Abu Farha, Y., Gall, J.: Multi-modal temporal convolutional network for anticipating actions in egocentric videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
64. Zeng, K.H., Shen, W.B., Huang, D.A., Sun, M., Carlos Niebles, J.: Visual forecasting by imitating dynamics in natural sequences. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2017)
65. Zhang, H., Chen, F., Yao, A.: Weakly-supervised dense action anticipation. In: Proceedings of the British Machine Vision Conference (BMVC) (2021)
66. Zhang, Y., Li, X., Liu, C., Shuai, B., Zhu, Y., Brattoli, B., Chen, H., Marsic, I., Tighe, J.: Vidtr: Video transformer without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)

67. Zou, C., Wang, B., Hu, Y., Liu, J., Wu, Q., Zhao, Y., Li, B., Zhang, C., Zhang, C., Wei, Y., et al.: End-to-end human object interaction detection with hoi transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

Rethinking Learning Approaches for Long-Term Action Anticipation

Megha Nawhal[1], Akash Abdu Jyothi[1], and Greg Mori[1,2]

[1] Simon Fraser University, Burnaby, Canada
[2] Borealis AI, Vancouver, Canada

A Appendix

In this document, we provide additional quantitative and qualitative analyses, and additional details of the implementation of our approach. Specifically, this document contains the following items.

- Sec. A.1: Technical details of the implementation and evaluation of our proposed approach
  - Sec. A.1.1: Architecture details (network architectures and loss function)
  - Sec. A.1.2: Implementation details (input representations and hyperparameters)
  - Sec. A.1.3: Evaluation details
- Sec. A.2: Additional ablation analysis
- Sec. A.3: Additional visualizations and qualitative analysis
- Sec. A.4 Additional discussion

A.1 Technical Details

In this section, we provide additional details for implementation of our proposed approach ANTICIPATR to supplement Sec. 3 in the main paper.

A.1.1 Architecture Details

We propose ANTICIPATR that uses a two-stage learning approach to train a transformer-based model for the task of long-term action anticipation. The model comprises three networks: *segment encoder*, *video encoder* and *anticipation decoder*. Fig. F1 shows the architecture of the three networks.

In the first stage, we train a *segment encoder* that receives a segment (sequence of frames from a video) as input and predicts the set of action labels that would occur at any future time instant after the occurrence of the segment in the video.

In the second stage, we train a *video encoder* and an *anticipation decoder* to be used along with the segment encoder for long-term action anticipation. The video encoder encodes the observed video to a video-level representation. The segment encoder (trained in the first stage) is fed with a sequence of segments

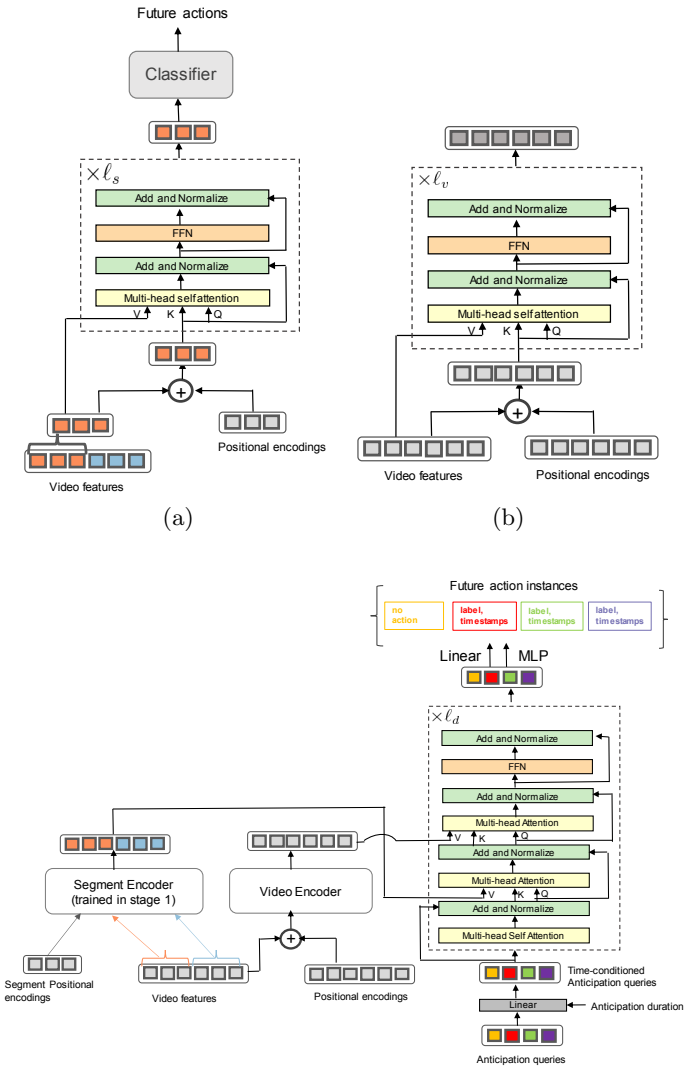


Fig. F1. Detailed Architecture. Architecture overview of (a) Segment encoder, (b) Video encoder, and (c) Anticipation Decoder. Refer to Sec. A.1.1 for details. 'Q','K','V' are query, key and value to the self-attention layer as described in [10].

from the observed video as input to obtain a segment-level representation of the video. The anticipation decoder receives the two representations along with the anticipation duration to predict a set of future action instances over the given anticipation duration in a single pass. The video encoder and anticipation decoder are trained using classification losses on the action labels and two temporal losses (L_1 loss and temporal IoU loss) on the timestamps while the segment encoder is kept unchanged.

Positional Encoding for Segment Encoder. The input to the segment encoder is a video segment. We represent the segment as a sequence of features. As the encoder is permutation-invariant, we provide temporal information in the segment using the sinusoidal positional encodings (c.f. Vaswani *et al.* [10]) based on timestamps corresponding to the features of input segment. Specifically, for each input feature of each embedding we independently use sine and cosine functions with different frequencies. We then concatenate them along the channel dimension to get the final positional encoding. In our implementation, the embedding size is same as that of the segment feature so that they can be combined by simple addition of the positional encodings and segment features.

Positional Encoding for Video Encoder. The input to the video encoder is a video. We represent the video as a sequence of features. As the transformer encoder is permutation-invariant, we provide temporal information in the input video using the sinusoidal positional encodings (c.f. Vaswani *et al.* [10]) based on timestamps corresponding to the features of input video. Specifically, for each input feature of each embedding we independently use sine and cosine functions with different frequencies. We then concatenate them along the channel dimension to to get the final positional encoding. In our implementation, the embedding size is same as that of the video feature so that they can be combined by simple addition of the positional encodings and video features.

Anticipation Queries (Anticipation Decoder). The anticipation queries are learnable positional encoding designed as a learnable embedding layer. The positional encoding layer receives integer index i as input corresponding to $i-th$ anticipation query and provides an embedding $\mathbf{q}_0^i$ where $i \in \{1, \ldots, N_a\}$. In our implementation, we use `torch.nn.Embedding` in Pytorch to implement this. The weights of the layer are learnable during training, thus, the positional encoding layer is also learnable. The initialization of this layer requires maximum possible value of the index, *i.e.*, N_a in our case.

The anticipation queries $\mathbf{q}_0$ are then combined with anticipation duration T_a using a simple neural network to create time-conditioned anticipation queries $\mathbf{q}_a$. These time-conditioned queries enable the model to predict actions over any specified anticipation duration.

Training. We provide supplemental details about computation of loss function used to train the networks in the second stage (*i.e.*, action anticipation stage) of our ANTICIPATR approach. The training involves aligning groundtruth and predicted set of action instances followed by computing the anticipation loss over all aligned pairs.

Greedy Set Correspondence. Given an observed video, the groundtruth set of future action instances varies based on input whereas our anticipation decoder predicts a set of fixed size (larger than maximum size of groundtruth sets in the dataset). Therefore, there is no prior correspondence between the groundtruth and predicted set. We derive this correspondence using a greedy algorithm based on temporal overlap among instances. Intuitively, the objective is to correctly align actions at as many future time instants as possible. We first sort the action instances groundtruth set based on the descending order of the duration of the instances. We begin the alignment process with the groundtruth instance having the maximum duration. We lookup the predicted set to find the predicted instance that has maximum temporal overlap with this groundtruth instance. Since the predicted set is designed to represent a single action instance, the alignment between groundtruth and predicted set is one-to-one. Thus, to continue the alignment process, the matched groundtruth instance and predicted instance are removed from the corresponding sets. In this way, this process is repeated until the groundtruth set is empty. As the predicted set is of size larger than groundtruth set, the remaining predicted instances are mapped to $\varnothing$ denoting no action. In Sec. A.2, we also evaluate anticipation results of models trained using another set correspondence algorithm, namely, Hungarian matcher (see Table T9 and Table T10).

Loss function. We compute loss $\mathcal{L}$ (defined in Eq. (4) in the main paper) over all the matched pairs as a weighted combination of cross-entropy loss for classification and two temporal losses ($L1$ loss and IoU loss $\mathcal{L}_{iou}$) for prediction of segment timestamps. Here, we provide our motivation behind temporal loss and provide additional description.

The L_1 temporal loss is sensitive to the absolute value of the duration of the segments. The IoU loss $\mathcal{L}_{iou}$ is invariant to the duration of the segments. Thus, these two losses together are designed to incorporate different aspects of segment prediction. For completeness, we describe $\mathcal{L}_{iou}$ as follows.

$$\mathcal{L}_{iou}(s^i, \hat{s}^{\gamma(i)}) = 1 - \frac{|s^i \cap \hat{s}^{\gamma(i)}|}{|s^i \cup \hat{s}^{\gamma(i)}|}, \tag{1}$$

where $|.|$ is the duration of the instance, *i.e.*, difference between end and start timestamp.

A.1.2 Training Details

For training of first stage, we use dropout probability of 0.1. For the segment encoder, we use base model dimension as 2048 and set the number of encoder layers as 3 with 8 attention heads. We use an effective batch size of 64 for training segment encoder on this dataset. For training in the second stage, we use base model dimension in the video encoder and anticipation decoder as 2048 and set the number of encoder and decoder layers as 3 with 8 heads.

We use four datasets – Breakfast, 50Salads, EPIC-Kitchens-55, EGTEA Gaze+ – to evaluate our model on long-term action anticipation. We provide dataset-specific hyperparameters as follows.

We train all our models using AdamW [6] optimizer on 4 Nvidia V100 32GB GPUs. We initialize all the learnable weights using Xavier initialization.

Breakfast. We represent input videos as I3D features provided by [1]. We choose N_a (anticipation queries) to be 150. We use an effective batch size of 16 for training the video encoder and anticipation decoder on this dataset on the long-term anticipation task. We train our models with a learning rate of 1e-4 and a weight decay of 0. The model is trained for 4000k steps. We use a dropout probability of 0.1. We set $\lambda_{L1} = 3$ and $\lambda_{iou} = 5$. To obtain segment-level representation of the observed video during action anticipation, we use a temporal window of length $k = 16$.

50Salads. We represent input videos as Fisher vectors computed using [2]. We choose N_a (anticipation queries) to be 80. We use an effective batch size of 16 for training the video encoder and anticipation decoder on this dataset on the long-term anticipation task. We use a learning rate of 1e-5 and a weight decay of 1e-5. We train the model for 3000k steps and reduce the learning rate by factor of 10 after 1500k steps. We don't use dropout for this dataset. We set $\lambda_{L1} = 3$ and $\lambda_{iou} = 5$. To obtain segment-level representation of the observed video during action anticipation, we use a temporal window of length $k = 48$.

EPIC-Kitchens-55. We represent input videos as I3D features provided by [3, 7]. We use an effective batch size of 16 for training the video encoder and anticipation decoder in the second stage. We choose N_a (anticipation queries) to be 900. We use a learning rate of 1e-4 and a weight decay of 1e-5. We train the model for 6000k steps and reduce the learning rate by factor of 10 after 4000k steps. We use a dropout probability of 0.1. We set $\lambda_{L1} = 5$ and $\lambda_{iou} = 8$. To obtain segment-level representation of the observed video during action anticipation, we use a temporal window of length $k = 32$.

EGTEA Gaze+. We represent input videos as I3D features provided by [3, 7]. We use an effective batch size of 16 for training the video encoder and anticipation decoder in the second stage. We choose N_a to be 600. We use a learning rate of 1e-5 and a weight decay of 1e-5. We train the model for 4000k steps and reduce the learning rate by factor of 10 after 3000k steps. We use a dropout probability of 0.1. We set $\lambda_{L1} = 3$ and $\lambda_{iou} = 5$. To obtain segment-level representation of the observed video during action anticipation, we use a temporal window of length $k = 24$.

A.1.3 Evaluation Details

Note that our model predicts a set of action instances, wherein, each action instance is of the form (label, start time, end time). To evaluate the model outputs as per the benchmarks, we do the following postprocessing.

For Breakfast and 50Salads, following the benchmark [9], we evaluate the action anticipation outputs over a dense timeline. Our proposed ANTICIPATR predicts a set of action instances. During evaluation, we process this set of action

Table T1. Ablation: Loss function (Breakfast and 50Salads). We report the mean over classes accuracy for different observation/anticipation durations. Higher values indicate better performance. ✓ and ✗ indicate whether the component of the temporal loss is used or not respectively.

	Method	$\beta_o \rightarrow$	20%				30%			
		$\beta_a \rightarrow$	10%	20%	30%	50%	10%	20%	30%	50%
Breakfast	L_1: ✗; $\mathcal{L}_{iou}$:✓		36.2	30.7	28.6	26.4	38.7	33.9	31.0	27.3
	L_1: ✓; $\mathcal{L}_{iou}$:✗		36.5	31.1	29.1	28.2	39.2	34.2	31.7	28.1
	L_1: ✓; $\mathcal{L}_{iou}$:✓		**37.4**	**32.0**	**30.3**	**28.6**	**39.9**	**35.7**	**32.1**	**29.4**
50Salads	L_1: ✗; $\mathcal{L}_{iou}$:✓		40.2	33.9	26.8	26.0	41.9	41.4	27.6	23.3
	L_1: ✓; $\mathcal{L}_{iou}$:✗		40.8	34.5	27.1	26.8	42.1	41.6	27.9	23.4
	L_1: ✓; $\mathcal{L}_{iou}$:✓		**41.1**	**35.0**	**27.6**	**27.3**	**42.8**	**42.3**	**28.5**	**23.6**

Table T2. Ablation: Loss function (EK-55 and EGTEA+). We report mAP values for ALL classes, FREQUENT classes (> 100 action instances) and RARE class (< 10 action instances). Following [7], we report the mAP values averaged over different observation durations. Higher values implies better performance. ✓ and ✗ indicate whether the component of the temporal loss is used or not respectively.

Method	**EK-55**			**EGTEA+**		
	ALL	FREQ	RARE	ALL	FREQ	RARE
L_1: ✗; $\mathcal{L}_{iou}$:✓	34.9	56.4	27.3	75.2	82.1	53.8
L_1: ✓; $\mathcal{L}_{iou}$:✗	37.7	57.8	28.4	76.0	82.7	54.6
L_1: ✓; $\mathcal{L}_{iou}$:✓	**39.1**	**58.1**	**29.1**	**76.8**	**83.3**	**55.1**

instances to construct a timeline corresponding to the anticipation duration. We refer to the timeline as a sequence of action labels for time instants in the anticipation duration, *i.e.*, between $T_o + 1, \ldots, T_o + T_a$. In the benchmarks, the timeline contains a single action class corresponding to each time instant. We iterate over the predicted set to assign class labels to this timeline. Specifically, for each action instance in the predicted set, we assign the predicted action class to the time instants that are within the predicted segment (determined by predicted start and end timestamp). When predicted action instances overlap at certain time instants, we assign the action class with highest probability score among the overlapping predictions. Once the timeline is constructed, we compute mean over classes accuracy [9] to evaluate the model performance. Note that we are constructing this timeline only during evaluation to follow the benchmark evaluation protocols.

For EPIC-Kitchens-55 and EGTEA Gaze+, we perform a union over the action classes in the predicted set of instances to obtain a set of future action classes. We remove $\varnothing$ class from this set and use this set to compute mAP as described in benchmark [7].

A.2 Additional Ablation Analysis

In this section, we report our findings from additional ablation experiments.

Ablation: Loss function. The training loss function defined in Eq. (4) in the main paper contains three components (cross-entropy loss and two temporal losses). We conduct ablation experiments by removing one of the temporal losses.

Table T3. Ablation: Anticipation Queries (Breakfast and 50Salads). We report the mean over classes accuracy for different observation/anticipation durations. Higher values indicate better performance.

	Method	$\beta_o \rightarrow$	20%				30%			
		$\beta_a \rightarrow$	10%	20%	30%	50%	10%	20%	30%	50%
Breakfast	$N_a = 50$		32.6	28.2	26.4	24.3	35.8	31.4	28.7	25.3
	$N_a = 150$		**37.4**	**32.0**	**30.3**	**28.6**	**39.9**	**35.7**	**32.1**	**29.4**
	$N_a = 500$		36.6	31.5	29.4	27.3	38.5	34.4	31.3	28.3
50Salads	$N_a = 20$		38.4	33.2	24.2	23.6	39.1	35.6	25.5	24.2
	$N_a = 80$		**41.1**	**35.0**	**27.6**	**27.3**	**42.8**	**42.3**	**28.5**	**23.6**
	$N_a = 320$		40.5	34.2	26.0	25.6	41.3	40.9	27.4	23.3

Table T4. Ablation: Anticipation Queries (EK-55 and EGTEA+). We report mAP values for ALL classes, FREQUENT classes ($>$ 100 action instances) and RARE class ($<$ 10 action instances). Following [7], we report the mAP values averaged over different observation durations. Higher values implies better performance.

Dataset		ALL	FREQ	RARE
EK-55	$N_a = 300$	34.3	55.6	24.2
	$N_a = 900$	**39.1**	**58.1**	**29.1**
	$N_a = 2700$	38.2	56.9	28.3
EGTEA+	$N_a = 200$	70.2	79.5	49.7
	$N_a = 600$	**76.8**	**83.3**	**55.1**
	$N_a = 1800$	75.3	82.4	53.3

Note that we always need cross entropy loss for the classification task. Results in Table T1 and Table T2 show that models trained with overall loss perform better than the ones trained with the ablated versions. Moreover, the models trained with only L_1 temporal loss perform better than the ones trained with only $\mathcal{L}_{iou}$.

Ablation: Anticipation queries. The number of anticipation queries discerns the maximum number of action instances the model is supposed to predict. Results in Table T3 and Table T4 shows the performance of our model with different number of anticipation queries. The results suggest minor improvement with higher number of anticipation queries, however, the models with more number of queries require longer training times. Intuitively, a very large number of anticipation queries implies the model will require more time to learn the non-maximal suppression of the irrelevant predictions. On the other hand, when the number of anticipation queries is reduced, the anticipation performance of our model degrades. A very small number of anticipation queries implies less number of action are anticipated. Thus, for very complex video with many future action instances, the model would miss several action instances resulting in poor anticipation performance. Additionally, as shown in Table T3, the anticipation error increases over time. This is because there are more actions to be anticipated and the model is limited by the number of anticipation queries.

Ablation: Segment window length. Results in Table T5 and Table T6 shows the performance of our model with different values of temporal window lengths used to extract segment-level representations during action anticipation. The results suggest that neither a very small window length nor a very large window is helpful. The segment encoder is trained to predict future actions given a video

Table T5. Ablation: Segment window length (Breakfast and 50Salads). We report the mean over classes accuracy for different observation/anticipation durations. Higher values indicate better performance.

	Method $\beta_o \rightarrow$		20%				30%			
		$\beta_a \rightarrow$	10%	20%	30%	50%	10%	20%	30%	50%
Breakfast	$k = 4$		35.9	30.6	26.3	26.1	38.4	33.6	30.8	28.2
	$k = 16$		**37.4**	**32.0**	**30.3**	**28.6**	**39.9**	**35.7**	**32.1**	**29.4**
	$k = 64$		37.4	31.7	29.9	28.1	39.1	35.0	31.7	28.7
50Salads	$k = 12$		39.0	33.5	25.8	25.4	39.6	38.4	26.4	21.5
	$k = 48$		**41.1**	**35.0**	**27.6**	**27.3**	**42.8**	**42.3**	**28.5**	**23.6**
	$k = 192$		41.0	34.8	27.2	26.8	42.6	42.1	27.5	22.8

Table T6. Ablation: Segment window length (EK-55 and EGTEA+). We report mAP values for ALL classes, FREQUENT classes (> 100 action instances) and RARE class (< 10 action instances). Following [7], we report the mAP values averaged over different observation durations. Higher values implies better performance.

Dataset		ALL	FREQ	RARE
EK-55	$k = 8$	37.9	57.2	27.4
	$k = 32$	**39.1**	**58.1**	**29.1**
	$k = 128$	38.8	58.0	28.7
EGTEA+	$k = 6$	75.4	81.7	53.9
	$k = 24$	**76.8**	**83.3**	**55.1**
	$k = 96$	76.3	82.9	54.8

segment depicting a single action. During the action anticipation stage, when the segment encoder is used to extract segment-level representations, the observed video is divided into a series of non-overlapping segment using temporal sliding windows as the action boundaries are not known. Intuitively, when the temporal sliding window is very small, the individual segments do not have enough information to obtain effective representations. On the other hand, when the window is very large, the segments contain more than one action and potentially results in segment-level representations with overlapping semantic content. We observe that the drop in performance with models that use smaller window lengths is larger as compared to the ones with larger window lengths.

Ablation: Sliding Windows for Segment Encoder Training. Instead of using action boundaries we used sliding temporal windows of length=k (same as used during stage 2) to obtain segments for segment-level training. Results in Table T7 and Table T8 show that this approach results in a slightly lower performance than our proposed training approach. This is possibly due to increased noise in the segment-level representations from this training approach.

Table T7. Ablation: Sliding windows for Segment Encoder Training. Mean over classes accuracy for different observation/anticipation durations. Higher is better. [BF: Breakfast; 50SL: 50Salads]

	Observation (β_o) $\rightarrow$	20%				30%			
	Anticipation (β_a) $\rightarrow$	10%	20%	30%	50%	10%	20%	30%	50%
BF	Sliding windows	35.9	30.7	28.0	26.4	37.8	33.5	29.9	25.2
	Anticipatr(Full)	**37.4**	**32.0**	**30.3**	**28.6**	**39.9**	**35.7**	**32.1**	**29.4**
50SL	Sliding windows	37.2	33.5	26.3	25.8	37.9	37.0	26.1	24.5
	Anticipatr(Full)	**41.1**	**35.0**	**27.6**	**27.3**	**42.8**	**42.3**	**28.5**	**23.6**

Table T8. Ablation: Sliding windows for Segment Encoder Training. mAP values for ALL classes, FREQUENT classes (> 100 action instances) and RARE class (< 10 action instances). Higher is better.

Method	EK-55			EGTEA+		
	ALL	FREQ	RARE	ALL	FREQ	RARE
Sliding Windows	37.6	56.5	27.4	74.8	81.2	53.0
No Video Encoder	30.9	51.8	21.2	70.2	79.9	50.1
Anticipatr(Full)	**39.1**	**58.1**	**29.1**	**76.8**	**83.3**	**55.1**

Ablation: Set correspondence. To compute the anticipation loss, we use a greedy algorithm to align groundtruth and predicted set of action instances. Another commonly employed set correspondence algorithm is Hungarian matcher algorithm used in prior works [4, 5]. For completeness, we also conducted experiments with Hungarian matcher optimized over the cost function with all three terms (classification loss and two temporal losses) following [8]. We didn't observe any significant difference in performance of the models trained using either of the two matchers as shown in Table T9 and Table T10.

A.3 Additional Qualitative Analysis

Visualizations in Fig. F2 and Fig. F3 show that our model is generally able to anticipate correct actions at future time instants long anticipation durations for Breakfast and 50Salads benchmarks respectively.

Visualizations in Fig. F4 and Fig. F5 show that our model is able to effectively predict future action classes for EK-55 and EGTEA benchmarks respectively.

Failure Cases. We observe that the action boundaries in some cases are not exactly aligned with the groundtruth even though the class labels are predicted accurately (See Fig. F2 and Fig. F3). We believe this could be because the visual information pertaining to the information is limited or negligible towards the beginning and end of the action instance.

Most classification errors result from the model getting confused among semantically similar classes. Some such cases from our examples are *'take ladle'* and *'pick-up ladle'* in Fig. F4(b)); *'close sandwich'* and *'close hamburger'* in Fig. F4(d)); *'put seasoning'* and *'pour seasoning'* in Fig. F5(a)).

Moreover, our model sometimes misses rare actions during predictions such as *'pour oil'* in Fig F4(a) and *'close fridge'* in Fig. F5(b).

Additionally, we also observe that having seen certain objects in the observed video, the model predicts objects that are likely to co-occur with the seen objects. See the scenario in Fig. F3(d). The model doesn't predict *'cut_cheese'* and *'place_cheese_into_bowl'* after the action *'place_cucumber_into_bowl'* and instead predicts *cut_tomato* and *'place_tomato_into_bowl'*. While the prediction is not correct for this specific activity, it is still a reasonable sequence of actions as there are several other salad recipe videos in the dataset that only use *cucumber* and *tomato*. In another scenario in Fig. F5(b), having seen *'pasta'* in the observed video, the model anticipates action classes with *'cheese'* noun. While

Table T9. Ablation: Set correspondence (Breakfast & 50Salads). We report the mean over classes accuracy for different observation/anticipation durations. Higher values indicate better performance.

	Method	$\beta_o \rightarrow$	20%				30%			
		$\beta_a \rightarrow$	10%	20%	30%	50%	10%	20%	30%	50%
Breakfast	Hungarian		36.8	**32.0**	**30.5**	28.4	39.2	35.4	31.9	**29.6**
	Greedy		**37.4**	32.0	**30.3**	**28.6**	**39.9**	**35.7**	**32.1**	29.4
50Salads	Hungarian		**41.3**	**35.1**	27.4	26.8	**42.9**	42.0	28.4	**23.8**
	Greedy		41.1	35.0	**27.6**	**27.3**	**42.8**	**42.3**	**28.5**	**23.6**

Table T10. Ablation: Set correspondence (EK-55 & EGTEA+). We report mAP values for ALL classes, FREQUENT classes ($>$ 100 action instances) and RARE class ($<$ 10 action instances). Following [7], we report the mAP values averaged over different observation durations. Higher values implies better performance.

Method	**EK-55**			**EGTEA+**		
	ALL	FREQ	RARE	ALL	FREQ	RARE
Hungarian	39.0	**58.4**	28.4	76.7	**83.5**	55.0
Greedy	**39.1**	58.1	**29.1**	**76.8**	83.3	**55.1**

'cheese' does not appear in this particular video, it is a reasonable prediction since the nouns *'pasta'* and *'cheese'* often appear together in activity videos in this dataset.

A.4 Additional Discussion

In this work, we demonstrate the effectiveness of our model on minutes-long activity videos. Handling longer videos with durations in hours or days (common in surveillance or monitoring scenarios) would be interesting future work. Furthermore, our approach assumes that the videos have an overall context provided by the ongoing long-term activity. We show that modeling interactions among segments (and, in turn, segment-level representation) is an effective technique for such activity videos as the video segments are indeed related. However, such approaches cannot tackle videos that are just a montage of several unrelated content like videos containing clips from different movies. Our approach focuses on activity videos where contextual information is present and relevant for action anticipation.

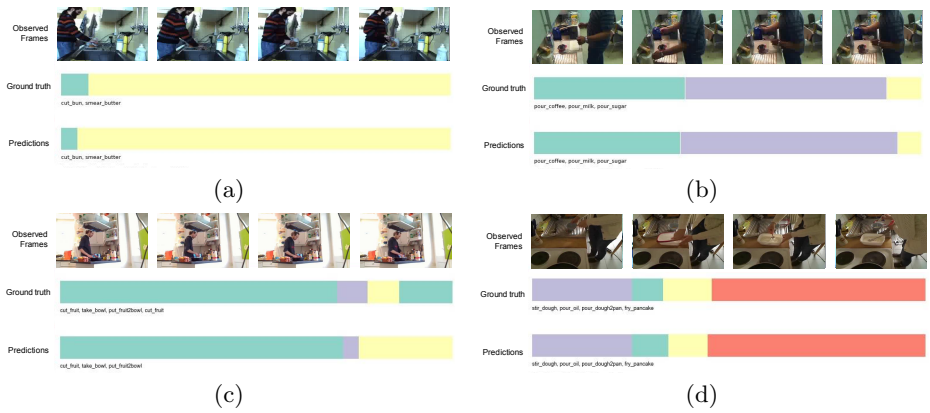


Fig. F2. Visualizations (Breakfast). Examples from Breakfast dataset for the case where observation duration is 20% of the video duration and anticipation duration involves predicting actions for 50% of the remaining video.

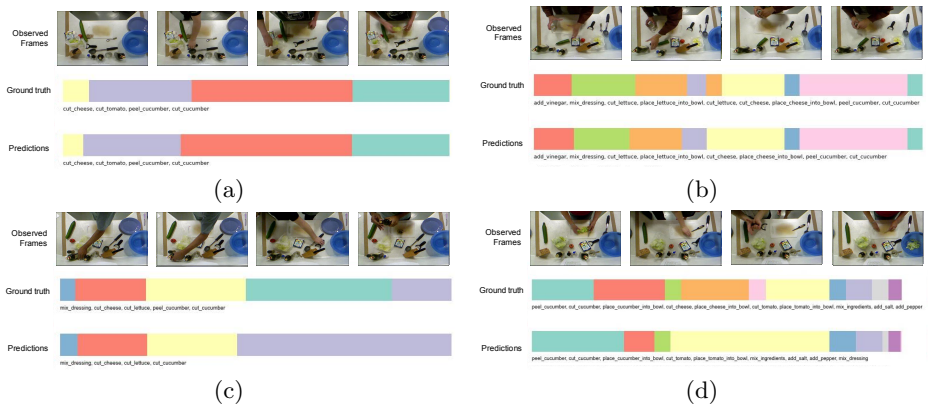


Fig. F3. Visualizations (50Salads). Examples from 50Salads dataset for the case where observation duration is 20% of the video duration and anticipation duration involves predicting actions for 50% of the remaining video.

pour oil
take pan
put-down spoon
put-down fork
take pan

turn-off stove
put-down pan
put egg
lift pan

(a)

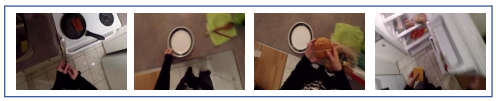
mix sauce
lift pot
put pasta
take ladle
pick-up ladle
stir pasta

mix sauce
lift pot
put spoon
put fork
put cheese

(b)

serve jambalaya
serve food
take fork
pick-up fork
put-down fork
move pot

(c)



put sausage
put cheese
put tomato
close sandwich
close hamburger
put-down tongs

(d)

Fig. F4. Visualizations (EK-55). Examples from Epic-Kitchens-55 dataset for the case where observation duration is 50% of the video duration. We show the predicted action classes in the visualization – classes in green color are correct predictions, classes in red color are wrong predictions, and classes in gray color are missed classes.

mix pasta
put condiment
take condiment-container
put seasoning
pour seasoning
take bowl
move-around pot
put pasta
pour pasta

take bell-pepper
mix bell-pepper
take oil-container
put pot
cut tomato
put cheese
mix cheese
mix seasoning

(a)

put cheese
close fridge
put patty
put lettuce
take lettuce

take condiment-container
take cheese
put tomato
squeeze sandwich
move-around patty

(b)

Fig. F5. Visualizations (EGTEA+). Examples from EGTEA Gaze+ dataset for the case where 50% of the video is observed. We show the predicted action classes in the visualization – classes in green color are correct predictions, classes in red color are wrong predictions, and classes in gray color are missed classes.

References

1. https://github.com/yabufarha/ms-tcn
2. https://bitbucket.org/doneata/fv4a/src/master/
3. https://github.com/facebookresearch/ego-topo
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
5. Kim, B., Lee, J., Kang, J., Kim, E.S., Kim, H.J.: Hotr: End-to-end human-object interaction detection with transformers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
6. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2017)
7. Nagarajan, T., Li, Y., Feichtenhofer, C., Grauman, K.: Ego-topo: Environment affordances from egocentric video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
8. Nawhal, M., Mori, G.: Activity graph transformer for temporal action localization. arXiv preprint arXiv:2101.08540 (2021)
9. Sener, F., Singhania, D., Yao, A.: Temporal aggregate representations for long-range video understanding. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS) (2017)