Future Transformer for Long-term Action Anticipation

Dayoung Gong[1] Joonseok Lee[1] Manjin Kim[1] Seong Jong Ha[2] Minsu Cho[1]
POSTECH[1] NCSOFT[2]
http://cvlab.postech.ac.kr/research/FUTR

Abstract

The task of predicting future actions from a video is crucial for a real-world agent interacting with others. When anticipating actions in the distant future, we humans typically consider long-term relations over the whole sequence of actions, i.e., not only observed actions in the past but also potential actions in the future. In a similar spirit, we propose an end-to-end attention model for action anticipation, dubbed Future Transformer (FUTR), that leverages global attention over all input frames and output tokens to predict a minutes-long sequence of future actions. Unlike the previous autoregressive models, the proposed method learns to predict the whole sequence of future actions in parallel decoding, enabling more accurate and fast inference for long-term anticipation. We evaluate our method on two standard benchmarks for long-term action anticipation, Breakfast and 50 Salads, achieving state-of-the-art results.

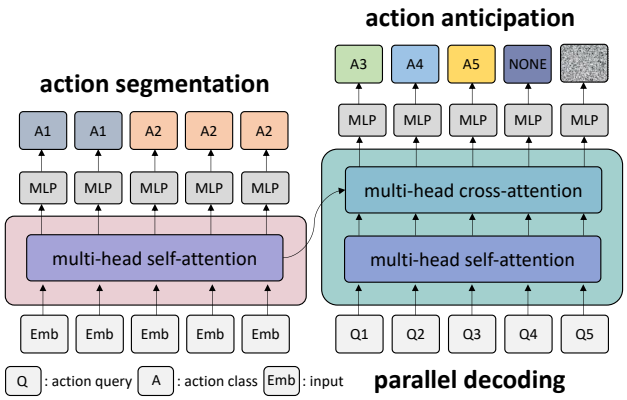


Figure 1. **Future Transformer (FUTR).** The proposed method is an end-to-end attention neural network to anticipate actions in parallel decoding, leveraging global interactions between past and future actions for long-term action anticipation.

1. Introduction

Long-term action anticipation from a video is recently emerging as an essential task for advanced intelligent systems. It aims to predict a sequence of actions in the future from a limited observation of past actions in a video. While there exists a growing body of research on action anticipation, most of the recent work focuses on predicting a single action in a few seconds [14–17, 30, 35–37]. In contrast, long-term action anticipation [2, 13, 19, 36] aims to predict a minutes-long sequence of multiple actions in the future. This task is challenging since it requires learning long-range dependencies between past and future actions.

Recent long-term anticipation methods [13, 36] encode observed video frames into condensed vectors and decode them via recurrent neural networks (RNNs) to predict a sequence of future actions in an autoregressive manner. Despite the impressive performance on the standard benchmarks [21, 38], they have several limitations. First, the encoder excessively compresses the input frame features so that fine-grained temporal relations between the observed frames are not preserved. Second, the RNN decoder is limited in modeling long-term dependencies over the input sequence and also in considering global relations between past and future actions. Third, the sequential prediction of autoregressive decoding may accumulate errors from the precedent results and also increase inference time.

To resolve the limitations, we introduce an end-to-end attention neural network, *Future Transformer (FUTR)*, for long-term action anticipation. The proposed method effectively captures long-term relations over the whole sequence of actions. *i.e.*, not only observed actions in the past but also potential actions in the future. FUTR is an encoder-decoder structure [6, 42] as illustrated in Fig. 1; the encoder learns to capture fine-grained long-range temporal relations between the observed frames from the past, while the decoder learns to capture global relations between upcoming actions in the future along with the observed features from the encoder. Different from the previous autoregressive models, FUTR anticipates a sequence of future actions in parallel decoding, enabling more accurate and faster inference without error accumulations. Furthermore, we employ an action segmentation loss for input frames to learn distinctive feature representations in the encoder. We evaluate FUTR on the

standard benchmarks for long-term action anticipation and achieve new state-of-the-art results on Breakfast and 50 Salads. The main contribution of our paper is four-fold:

- We introduce an end-to-end attention neural network, dubbed FUTR, which effectively leverages fine-grained features and global interactions for long-term action anticipation.
- We propose to predict a sequence of actions in parallel decoding, enabling accurate and fast inference.
- We develop an integrated model that learns distinctive feature representation by segmenting actions in the encoder and anticipating actions in the decoder.
- The proposed method sets a new state of the arts on standard benchmarks for long-term action anticipation, Breakfast and 50 Salads.

2. Related Work

Action anticipation. Action anticipation aims to predict future actions given a limited observation of a video. With the emergence of the large-scale dataset [8, 9], many methods have been proposed to solve next action anticipation, predicting a single future action within a few seconds [14–17, 30, 35–37]. Long-term action anticipation has been recently proposed to predict a sequence of future actions in the distant future from a long-range video [1, 2, 13, 19, 36]. Farha et al. [2] first introduce the long-term action anticipation task and propose two models, RNN and CNN, to tackle the task. Farha and Gall [1] introduce a GRU network to model the uncertainty of future activities in an autoregressive way. They predict multiple possible sequences of future actions at test time. Ke et al. [19] introduce a model that predicts an action in a specific future timestamp without anticipating intermediate actions. They show that iterative predictions of the intermediate actions cause error accumulations. Previous methods [1, 2, 19] typically take action labels of observed frames as input, extracting action labels using the action segmentation model [34]. In contrast, recent work [13, 36] uses visual features as input. Farha et al. [13] propose the end-to-end model of long-term action anticipation, employing the action segmentation model [12] for visual features in training. They also introduce a GRU model with cycle consistency between past and future actions. Sener et al. [36] suggest a multi-scale temporal aggregation model that aggregates past visual features in condensed vectors and then iteratively predicts future actions using the LSTM network. The recent work [13, 36] commonly utilizes RNNs with compressed representation of past frames. In contrast, we propose an end-to-end attention model that anticipates all future actions in parallel using fine-grained visual features of past frames.

Self-attention mechanisms. Self-attention [42] was initially introduced for neural machine translation to mitigate the problem of learning long-term dependencies in RNNs and has been widely adopted in a variety of computer vision tasks [10, 11, 20, 40]. Self-attention is effective in learning global interactions among image pixels or patches in image domains [4, 10, 26, 33, 40, 41, 44, 46, 47]. Several methods employ attention mechanisms in video domains to model temporal dynamics in short-term [3, 5, 11, 20, 32, 45, 48] and long-term videos [17, 24, 29, 31, 49]. Related to action anticipation, Girdhar and Grauman [17] recently introduce the anticipative video transformer (AVT) that uses a self-attention decoder to predict the next action. Unlike AVT, which requires autoregressive predictions for long-term anticipation, our encoder-decoder model effectively predicts a minutes-long sequence of future actions in parallel.

Parallel decoding. The transformer [42] is designed to predict outputs sequentially, *i.e.*, autoregressive decoding. Due to the inference cost, which increases with the length of the output sequence, recent methods in natural language processing [18, 39] replace autoregressive decoding with parallel decoding. The transformer models with parallel decoding have also been used for computer vision tasks such as object detection [6], camera calibration [23], and dense video captioning [43]. We adopt it for long-term action anticipation, predicting a sequence of future actions simultaneously. In long-term action anticipation, parallel decoding not only enables faster inference but also captures bi-directional relations among future actions.

3. Problem Setup

The problem of long-term action anticipation is to predict a sequence of actions for future video frames from a given observable part of a video. Figure 2 illustrates the problem setup. For a video with T frames, the first αT frames are observed and a sequence of actions for the next βT frames is anticipated; $\alpha \in [0, 1]$ is an observation ratio of the video while $\beta \in [0, 1-\alpha]$ is a prediction ratio. The anticipation thus takes the observable frames $\boldsymbol{I}^{\text{past}} = [\boldsymbol{I}_1, \dots, \boldsymbol{I}_{\alpha T}]^\top \in \mathbb{R}^{\alpha T \times H \times W \times 3}$ as input and predicts a sequence of frame-wise action class labels for the next βT frames, $\boldsymbol{S}^{\text{future}} = [\boldsymbol{s}_{\alpha T+1}, \dots, \boldsymbol{s}_{\alpha T+\beta T}]^\top \in \mathbb{R}^{\beta T \times K}$ where K is the number of target actions. Following the previous work [1, 2, 13, 19, 36], we represent $\boldsymbol{S}^{\text{future}}$ as a sequence of action segments, each of which consists of an action and its duration, and predict a sequence of action class labels $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_N]^\top \in \mathbb{R}^{N \times K}$ and their durations $\boldsymbol{d} = [d_1, \dots, d_N] \in \mathbb{R}^N$ where $\sum_{j=0}^{N} d_j = 1$.

For evaluation, the sequence of action segments is translated to that of frame-wise actions; the action label $\boldsymbol{s}_{\alpha T+t}$ at time $\alpha T + t$ and that of i^{th} segment are related by

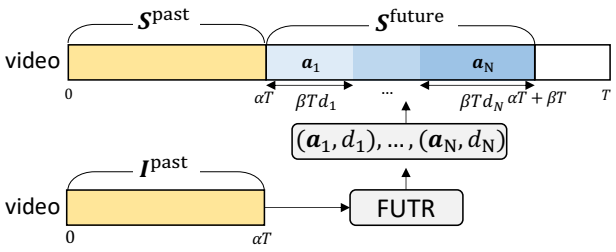


Figure 2. **Long-term action anticipation**. The problem of long-term action anticipation aims to predict action labels of βT future frames observing αT frames from a video, where α and β indicate the observation and prediction ratio of the full video frames T, respectively. FUTR anticipates action labels and durations of the N action segments, where predicted action labels and duration are decoded into frame-level action labels for evaluation.

$$\boldsymbol{s}_{\alpha T+t} = \boldsymbol{a}_i, \quad \beta T \sum_{j=0}^{i-1} d_j < t \leq \beta T \sum_{j=0}^{i} d_j, \tag{1}$$

where $d_0 = 0$.

In addition, the ground-truth action labels for the past frames are denoted by $\boldsymbol{S}^{\text{past}} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{\alpha T}]^\top \in \mathbb{R}^{\alpha T \times K}$, which are used for action segmentation loss in our work.

4. Future Transformer (FUTR)

In this section, we introduce a fully attention-based network, FUTR, for long-term action anticipation. The overall architecture consists of a transformer encoder and a decoder, as depicted in Fig. 3. Section 4.1 explains the encoder, which segments action labels from the fine-grained visual features of past frames, Section 4.2 describes the decoder, which predicts action labels and durations of future frames in parallel decoding, and then Section 4.3 presents the training objective of the proposed method.

4.1. Encoder

The encoder takes visual features as input and segments actions of past frames, learning distinctive feature representations via self-attention.

Input embedding. As input to the encoder, we use visual features extracted from the input frames $\boldsymbol{I}^{\text{past}}$, which are denoted by $\boldsymbol{F}^{\text{past}} \in \mathbb{R}^{\alpha T \times C}$ [13, 36]. We sample frames with a temporal stride of τ, establishing $\boldsymbol{E} \in \mathbb{R}^{T^{\text{O}} \times C}$ where $T^{\text{O}} = \lfloor \frac{\alpha T}{\tau} \rfloor$ is the number of sampled frames. The sampled frame features are fed to linear layer $\boldsymbol{W}^{\text{F}} \in \mathbb{R}^{C \times D}$ followed by ReLU activation function to $\boldsymbol{E}$, creating input tokens $\boldsymbol{X}_0 \in \mathbb{R}^{T^{\text{O}} \times D}$:

$$\boldsymbol{X}_0 = \text{ReLU}(\boldsymbol{E}\boldsymbol{W}^{\text{F}}). \tag{2}$$

Attention. The encoder consists of the L^{E} number of encoder layers. Each encoder layer is composed of a multi-head self-attention (MHSA), layer normalization (LN) and feed-forward networks (FFN) with residual connection. We define a multi-head attention (MHA) based on the scaled dot-product attention [42] with input variables $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$\text{MHA}(\boldsymbol{X}, \boldsymbol{Y}) = [\boldsymbol{Z}_1, .., \boldsymbol{Z}_h]\boldsymbol{W}^{\text{O}}, \tag{3}$$

$$\boldsymbol{Z}_i = \text{ATTN}_i(\boldsymbol{X}, \boldsymbol{Y}), \tag{4}$$

$$\text{ATTN}_i(\boldsymbol{X}, \boldsymbol{Y}) = \sigma(\frac{(\boldsymbol{X}\boldsymbol{W}_i^{\text{Q}})(\boldsymbol{Y}\boldsymbol{W}_i^{\text{K}})^\top}{\sqrt{D/h}})\boldsymbol{Y}\boldsymbol{W}_i^{\text{V}}, \tag{5}$$

where $\boldsymbol{W}_i^{\text{Q}}, \boldsymbol{W}_i^{\text{K}}$ and $\boldsymbol{W}_i^{\text{V}} \in \mathbb{R}^{D \times D/h}$ are query, key, and value projection layer at i^{th} head, respectively, $\boldsymbol{W}^{\text{O}} \in \mathbb{R}^{D \times D}$ is an output projection layer, h is the number of heads, and σ indicates a softmax. MHSA is MHA with the two same inputs:

$$\text{MHSA}(\boldsymbol{X}) = \text{MHA}(\boldsymbol{X}, \boldsymbol{X}). \tag{6}$$

The output token X_{l+1} is obtained from the l^{th} encoder layer:

$$\boldsymbol{X}_{l+1} = \text{LN}(\text{FFN}(\boldsymbol{X}'_l) + \boldsymbol{X}'_l), \tag{7}$$

$$\boldsymbol{X}'_l = \text{LN}(\text{MHSA}(\boldsymbol{X}_l + \boldsymbol{P}) + \boldsymbol{X}_l), \tag{8}$$

where an absolute 1-D positional embeddings $\boldsymbol{P} \in \mathbb{R}^{T^{\text{O}} \times D}$ is added to the input of the l^{th} layer $\boldsymbol{X}_l \in \mathbb{R}^{T^{\text{O}} \times D}$ for each MHSA layer.

Action segmentation. The final output of the last encoder layer $X_{L^{\text{E}}}$ is utilized to generate action segmentation logits $\hat{\boldsymbol{S}}^{\text{past}} \in \mathbb{R}^{T^{\text{O}} \times K}$ by applying a fully-connected (FC) layer $\boldsymbol{W}^{\text{S}} \in \mathbb{R}^{D \times K}$ followed by a softmax:

$$\hat{\boldsymbol{S}}^{\text{past}} = \sigma(\boldsymbol{X}_{L^{\text{E}}}\boldsymbol{W}^{\text{S}}). \tag{9}$$

4.2. Decoder

The decoder takes learnable tokens as input, referred to as *action queries*, and anticipates future action labels and corresponding durations in parallel, learning long-term relations between past and future actions via self-attention and cross-attention.

Action query. Action queries are embedded with M learnable tokens $\boldsymbol{Q} \in \mathbb{R}^{M \times D}$. The temporal orders of the queries is fixed to be equivalent to that of the future actions, *i.e.*, the i^{th} query corresponds to the i^{th} future action. We demonstrate that fixing temporal orders of the queries is effective for long-term action anticipation (Sec. 5.4).

Attention. The decoder consists of L^{D} number of decoder layers. Each decoder layer is composed of an MHSA, a multi-head cross-attention (MHCA), LN, and FFN. The output query $\boldsymbol{Q}_{l+1}$ is obtained from the l^{th} decoder layer:

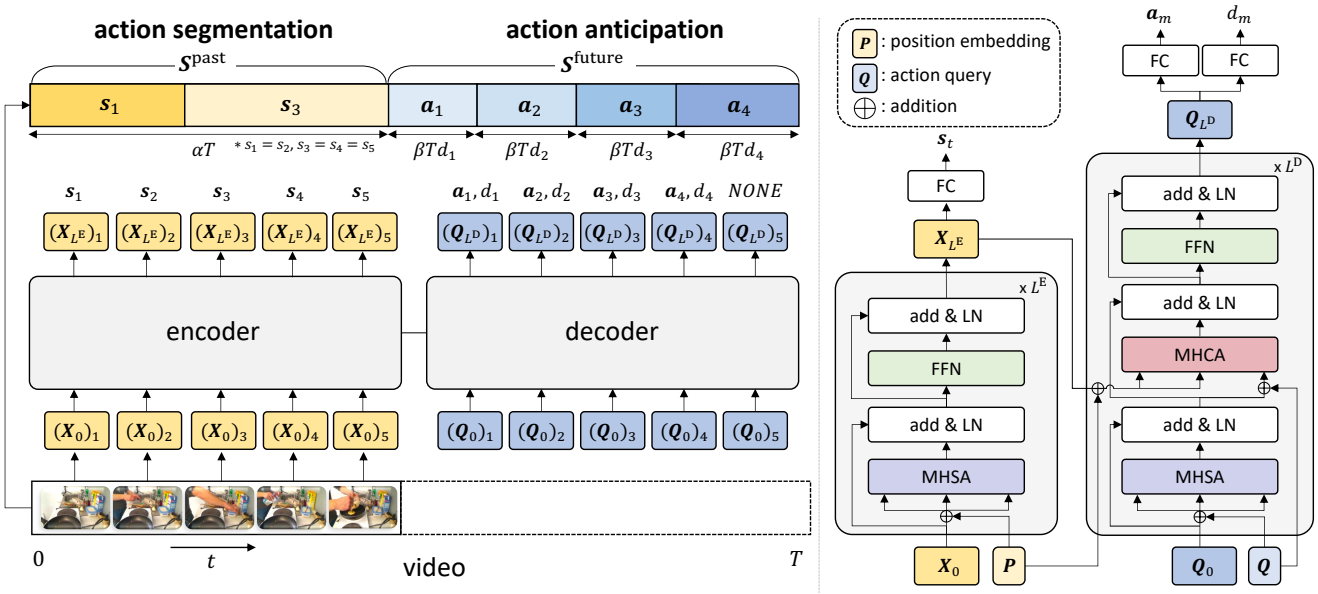


Figure 3. **Overall architecture of FUTR.** The proposed method is composed of an encoder and a decoder; each classifies action labels of past frames (action segmentation) and anticipates future action labels and corresponding durations (action anticipation), respectively. The encoder learns distinctive feature representation from past actions via self-attention, and the decoder learns long-term relations between past and future actions via self-attention and cross-attention. For simplicity, we set the number of past frames αT as 5 and the number of object queries M as 5 in this figure. Note that $(X_l)_i$ and $(Q_l)_i$ indicate i^{th} index of X_l and Q_l, respectively.

$$\boldsymbol{Q}_{l+1} = \text{LN}(\text{FFN}(\boldsymbol{Q}''_l) + \boldsymbol{Q}''_l), \tag{10}$$

$$\boldsymbol{Q}''_l = \text{LN}(\text{MHA}(\boldsymbol{Q}'_l + \boldsymbol{Q}, \boldsymbol{X}_{L^{\text{E}}} + \boldsymbol{P}) + \boldsymbol{Q}'_l), \tag{11}$$

$$\boldsymbol{Q}'_l = \text{LN}(\text{MHSA}(\boldsymbol{Q}_l + \boldsymbol{Q}) + \boldsymbol{Q}_l), \tag{12}$$

$$\boldsymbol{Q}_0 = [0, \dots, 0]^\top \in \mathbb{R}^D, \tag{13}$$

where $\boldsymbol{X}_{L^{\text{E}}}$ is the final output of the encoder. Note that action query Q is added to the input of the l^{th} layer $Q_l \in \mathbb{R}^{M \times D}$ for each MHSA layer.

Action anticipation. The final output of the last decoder layer $Q_{L^{\text{D}}}$ is utilized to generate future actions logits $\hat{\boldsymbol{A}} \in \mathbb{R}^{M \times (K+1)}$ by appling a FC layer $\boldsymbol{W}^{\text{A}} \in \mathbb{R}^{D \times K+1}$ followed by a softmax and duration vectors $\hat{\boldsymbol{d}} \in \mathbb{R}^M$ by applying a FC layer $\boldsymbol{W}^{\text{D}} \in \mathbb{R}^D$:

$$\hat{\boldsymbol{A}} = \sigma(\boldsymbol{Q}_{L^{\text{D}}} \boldsymbol{W}^{\text{A}}), \tag{14}$$

$$\hat{\boldsymbol{d}} = \boldsymbol{Q}_{L^{\text{D}}} \boldsymbol{W}^{\text{D}}. \tag{15}$$

4.3. Training objective

Action segmentation loss. An auxiliary action segmentation loss is used to learn distinctive feature representations of past actions in the encoder. The action segmentation loss is defined with the cross-entropy loss between target actions $\boldsymbol{S}^{\text{past}}$ and logits $\hat{\boldsymbol{S}}^{\text{past}}$:

$$\mathcal{L}^{\text{seg}} = -\sum_{i=1}^{T^{\text{O}}} \sum_{j=1}^{K} \boldsymbol{S}^{\text{past}}_{i,j} \log \hat{\boldsymbol{S}}^{\text{past}}_{i,j}. \tag{16}$$

Action anticipation losses. M action queries are matched to the N ground-truth actions to apply action anticipation losses. If none of the future actions is predicted, we let the queries predict a dummy class, '*NONE*.' Action anticipation loss $\mathcal{L}^{\text{action}}$ is defined with the cross-entropy between target actions $\boldsymbol{A}$ and logits $\hat{\boldsymbol{A}}$, and duration regression loss $\mathcal{L}^{\text{duration}}$ is defined with L2 loss between target durations $\boldsymbol{d}$ and the predicted durations $\hat{\boldsymbol{d}}$:

$$\mathcal{L}^{\text{action}} = -\sum_{i=1}^{M} \sum_{j=1}^{K+1} \mathbb{1}_{i \leq \delta} \boldsymbol{A}_{i,j} \log(\hat{\boldsymbol{A}}_{i,j}), \tag{17}$$

$$\mathcal{L}^{\text{duration}} = \sum_{i=1}^{M} \mathbb{1}_{\text{argmax}(\hat{\boldsymbol{A}}_i) \neq \textit{NONE}} (\boldsymbol{d}_i - \hat{\boldsymbol{d}}_i)^2, \tag{18}$$

where δ is the position of the first query that predicts *NONE* and $\mathbb{1}_{i \leq \delta}$ is an indicator function that sets to one where the query position i is less than or equal to δ. $\mathbb{1}_{\text{argmax}(\hat{\boldsymbol{A}}_i) \neq \textit{NONE}}$ is also an indicator function that sets to one where the predicted action of the i^{th} query is not *NONE*. Note that we apply gaussian normalization to the predicted duration to make summation of the whole durations as 1 following the previous work [2, 13].

Final loss. The overall training objective $\mathcal{L}^{\text{total}}$ of FUTR is the sum of action segmentation loss $\mathcal{L}^{\text{seg}}$, action anticipation loss $\mathcal{L}^{\text{action}}$, and duration regression loss $\mathcal{L}^{\text{duration}}$. The final loss $\mathcal{L}^{\text{total}}$ is defined by

$$\mathcal{L}^{\text{total}} = \mathcal{L}^{\text{seg}} + \mathcal{L}^{\text{action}} + \mathcal{L}^{\text{duration}}. \tag{19}$$

dataset	input type	methods	β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
			0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
Breakfast	label	RNN [2]	18.11	17.20	15.94	15.81	21.64	20.02	19.73	19.21
		CNN [2]	17.90	16.35	15.37	14.54	22.44	20.12	19.69	18.76
		UAAA (mode) [1]	16.71	15.40	14.47	14.20	20.73	18.27	18.42	16.86
		Time-Cond. [19]	18.41	17.21	16.42	15.84	22.75	20.44	19.64	19.75
	features	CNN [2]	12.78	11.62	11.21	10.27	17.72	16.87	15.48	14.09
		Temporal Agg. [36]	24.20	21.10	20.00	18.10	<u>30.40</u>	26.30	23.80	21.20
		Cycle Cons. [13]	<u>25.88</u>	<u>23.42</u>	<u>22.42</u>	<u>21.54</u>	29.66	<u>27.37</u>	<u>25.58</u>	<u>25.20</u>
		FUTR (ours)	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**
50 Salads	label	RNN [2]	30.06	25.43	18.74	13.49	30.77	17.19	14.79	09.77
		CNN [2]	21.24	19.03	15.98	09.87	29.14	20.14	17.46	10.86
		UAAA (mode) [1]	24.86	22.37	19.88	12.82	29.10	20.50	15.28	12.31
		Time-Cond. [19]	32.51	27.61	21.26	<u>15.99</u>	<u>35.12</u>	**27.05**	<u>22.05</u>	<u>15.59</u>
	features	Temporal Agg. [36]	25.50	19.90	18.20	15.10	30.60	22.50	19.10	11.20
		Cycle Cons. [13]	<u>34.76</u>	**28.41**	<u>21.82</u>	15.25	34.39	23.70	18.95	**15.89**
		FUTR (ours)	**39.55**	<u>27.54</u>	**23.31**	**17.77**	**35.15**	<u>24.86</u>	**24.22**	15.26

Table 1. **Comparison with the state of the art.** Our models set a new state of the art on Breakfast, and 50 Salads. The numbers in bold-faced and in underline indicates the highest and the second highest accuracy, respectively.

5. Experiments

5.1. Datasets

We evaluate our method on two standard action anticipation benchmarks: the Breakfast dataset and 50 Salads.

The Breakfast [21] dataset comprises 1,712 videos of 52 different individuals cooking breakfast in 18 different kitchens. Every video is categorized into one of the 10 activities related to breakfast preparation. There exist 48 fine-grained action labels which are used to make up the activities. On average, each video is about 2.3 minutes long and includes approximately 6 actions. All videos were downsampled to a resolution of 240×320 pixels with a frame rate of 15 fps. The dataset comprises 4 splits of training and test set, and we report the average performance over all the splits following the previous work [2, 13, 36].

The 50 Salads [38] dataset comprises 50 videos of 25 people preparing a salad. The dataset contains over 4 hours of RGB-D video data, annotated with 17 fine-grained action labels and 3 high-level activities. Since 50 Salads is usually longer than Breakfast, each video contains 20 actions on average. Every video in the dataset has a resolution of 480×640 pixels with a frame rate of 30 fps. The dataset comprises 5 splits of training and test set, and we report the average results over all the splits.

5.2. Implementation details

Architecture details. Our model consists of two encoder layers and one decoder layer for Breakfast and two encoder layers and two encoder layers for 50 Salads. We set the number of object queries M to 8 for Breakfast and 20 for 50 Salads since 50 Salads includes more actions than Breakfast in a video. The size of hidden dimension D is set to 128 for Breakfast and 512 for 50 Salads.

Training & testing. We use pre-extracted I3D features [7] as input visual features for both Breakfast and 50 Salads provided by [12]. We sample the I3D features with a stride τ of 3 for Breakfast and 6 for 50 Salads. In training, we set the observation rate $\alpha \in \{0.2, 0.3, 0.5\}$ and fix the prediction rate β to 0.5. We use AdamW optimizer [28] with a learning rate of 1e-3. We train our model for 60 epochs with a batch size of 16, employing a cosine annealing warm-up scheduler [27] with warm-up stages of 10 epochs. In inference, we set the observation rate $\alpha \in \{0.2, 0.3\}$ and prediction rate $\beta \in \{0.1, 0.2, 0.3, 0.5\}$ and measure mean over classes (MoC) accuracy following the long-term action anticipation framework protocol [2, 13, 19, 36].

5.3. Comparison with the state of the art

In Table 1, we compare our methods with the state-of-the-art methods on Breakfast and 50 Salads. The table is divided into two compartments according to the dataset, and each compartment is divided into two sub-compartments according to the input types; the first and the second sub-compartment utilize action labels extracted from the action segmentation model [34] and visual features, respectively. For Breakfast, CNN [2] uses the Fisher vectors [2], and the other models use I3D features [7] as input. For 50 Salads, Sener et al. [36] use the Fisher vectors and Farha et al. [13] use I3D features. As a result, our methods achieve the

method	AR	causal mask	β ($\alpha = 0.3$)				time (ms)
			0.1	0.2	0.3	0.5	
FUTR-A	✓	✓	27.10	25.41	23.28	20.51	14.68
FUTR-M	-	✓	31.82	28.55	26.57	24.17	5.70
FUTR	-	-	**32.27**	**29.88**	**27.49**	**25.87**	**3.91**

Table 2. **Parallel decoding vs. autoregressive decoding.** Parallel decoding significantly improves both accuracy and inference speed. FUTR-A autoregressively anticipates future actions using predicted action labels with masked self-attention, and FUTR-M anticipates future actions using action queries with masked self-attention. All the reported inference speeds are measured on a single RTX 3090 GPU, ignoring data loading time. We feed a single video on GPU and take an average on the test set, except the first ten samples as a warm-up stage for stable inference time [25].

state-of-the-art performance in all experimental settings on Breakfast and 6 out of 8 settings on 50 Salads, respectively, using visual features only.

5.4. Analysis

We conduct in-depth analyses to validate the efficacy of the proposed method. In the following experiments, we evaluate our method on the Breakfast dataset setting the observation ratio α as 0.3. Unless otherwise specified, all experimental settings are the same as those in Sec. 5.2. Further experimental details are indicated in Supp. A.

Parallel decoding vs. autoregressive decoding. To validate the effectiveness of parallel decoding for long-term action anticipation, we compare our method with two FUTR variants with different decoding strategies. The first variant FUTR-A autoregressively anticipates future actions similar to transformer [42]. FUTR-A takes the output action labels from the previous predictions as input and utilizes masked self-attention. Masked self-attention employs a causal mask to MHSA, which prevents attending to future actions. The second variant FUTR-M is equivalent to FUTR except for masked self-attention applied to action queries. FUTR-M takes the action queries as input and predicts future actions in parallel, but each query only attends to the past queries.

Table 2 summarizes the results. FUTR-M outperforms FUTR-A by 3.1-4.7%p, with 2.6× faster inference time. These results demonstrate the effectiveness of parallel decoding using action queries in terms of accuracy and efficiency. As we remove the causal mask from FUTR-M, we obtain an additional accuracy improvement of 0.4-1.7%p. Compared to FUTR-A, FUTR achieves higher accuracy by 4.2-5.4%p, inferring 3.8× faster. These results show the effectiveness of parallel decoding of leveraging bi-directional dependencies between action queries leading to a more accurate and faster inference.

Global self-attention vs. local self-attention. We investigate the effect of learning long-term temporal dependencies between past and future actions by comparing global self-attention (GSA) and local self-attention (LSA) [33]. We build a FUTR variant that computes LSA in both the encoder and the decoder, and then replaces LSA with GSA one by one. We set the window sizes of LSA in the encoder and the decoder as 201 and 3, respectively, where only local area within the window size is utilized for MHA.

encoder	decoder	β ($\alpha = 0.3$)			
		0.1	0.2	0.3	0.5
LSA	LSA	27.70	24.39	23.18	21.60
GSA	LSA	30.15	27.51	25.62	23.28
LSA	GSA	28.37	25.08	24.03	22.28
GSA	GSA	**32.27**	**29.88**	**27.49**	**25.87**

Table 3. **Global self-attention vs. local self-attention.** Using GSA in both the encoder and the decoder improves the performance, indicating that learning long-term dependencies not only between the observed frames but also between the possible future actions is important for long-term action anticipation.

method	GT Assign.	regression	β ($\alpha = 0.3$)			
			0.1	0.2	0.3	0.5
FUTR-S	sequential	start-end	29.15	25.51	24.20	21.43
FUTR-H	Hungarian	start-end	25.26	23.85	22.63	21.45
FUTR	sequential	duration	**32.27**	**29.88**	**27.49**	**25.87**

Table 4. **Output structuring.** FUTR shows the superior performance over FUTR-S and FUTR-H. We find that sequential order of the ground truth assignment and duration regression is effective for long-term action anticipation. Start-end regression predicts normalized start and end timestamp for each query instead of the duration.

The results in Table 3 validate the efficacy of using GSA in long-term action anticipation. From the 1st and 2nd rows, we observe that replacing LSA with GSA in the encoder improves the accuracy by 1.7-3.1%p. The result verifies that learning the global context between past frames is crucial. As we replace LSA with GSA in the decoder from the 1st and 3rd rows, we find consistent accuracy improvement by 0.7-0.9%p. We find that learning global temporal relations between action queries is also essential for anticipating a sequence of future actions. Finally, we replace all LSA to GSA in both the encoder and the decoder from the 1st and 4th rows, which brings significant improvements by 4.3-5.5%p, achieving the best accuracy. These results demonstrate the importance of learning long-term relations between the observed actions in the past and potential actions in the future.

Output structuring. To obtain the final output, we consider the action queries as an ordered *sequence* and train FUTR to predict an action label and its duration from each in the sequence of action queries; in training, the ground truths of label and duration are directly assigned to the out-

loss			β ($\alpha = 0.3$)			
$\mathcal{L}^{\text{seg}}$	$\mathcal{L}^{\text{action}}$	$\mathcal{L}^{\text{duration}}$	0.1	0.2	0.3	0.5
-	✓	✓	28.31	25.85	24.91	22.50
✓	✓	✓	**32.27**	**29.88**	**27.49**	**25.87**

Table 5. **Loss ablations.** Action segmentation loss significantly improves the performance, indicating that recognizing past frames is crucial for anticipating future actions.

puts of the queries in the sequential order. To validate the effectiveness of this output structuring strategy, we compare it with two FUTR variants with different output structuring methods. FUTR-S is a variant of our method that is trained to predict a temporal window of starting and ending points, instead of a duration, from each in the query sequence. In inference, the predicted start-end windows are merged with priority according to classification logits; the most confident action labels are assigned to overlapping regions of windows. FUTR-H is a DETR-like variant [6] that considers the action queries as an unordered *set*, not a sequence, and predicts a start-end window from each in the query set. In training, the ground truths are assigned to the outputs of the queries by the Hungarian matching [22]. The matching cost function is defined as the sum of negative class probability and a window loss. The Hungarian matching loss is defined as the sum of the action anticipation loss and the window loss. See Supp. A for the details.

Table 4 shows the performances of the two variants and ours. The comparison between FUTR-S and FUTR-H shows that the sequential ground-truth assignment is more effective in training than the Hungarian assignment, which implies that the fixed sequence of action queries facilitates to capture temporal dependencies in a more effective manner. The comparison between FUTR and FUTR-H finds that the duration regression is more effective than the start-end regression, achieving a significant accuracy gain.

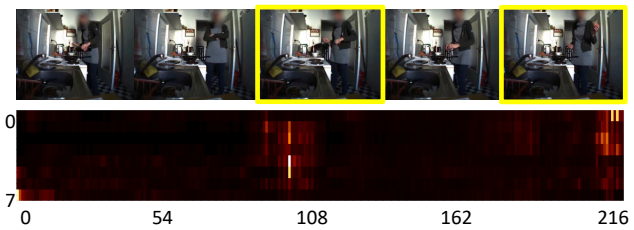
Loss ablations. In Table 5, we evaluate the effectiveness of action segmentation loss. As a result, action segmentation loss significantly improves the performance, indicating that recognizing past frames plays a crucial role in anticipating future actions.

Number of action queries. To analyze the impact of the number of action queries M in FUTR, we adjust the value of M from 6 to 10. In Table 6, the performance becomes saturated as we gradually increase the number of action queries. By this experiment, we set M to 8 for Breakfast.

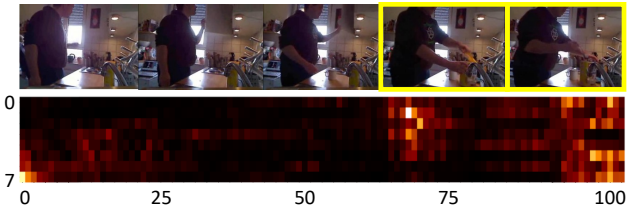
See Supp. C and D for additional analysis and results.

5.5. Attention map visualization

We visualize the cross-attention layers in the decoder in Fig. 4. The vertical and horizontal axis indicates the index of the action queries and input past frames, respectively.



(a) Activity: Fried egg



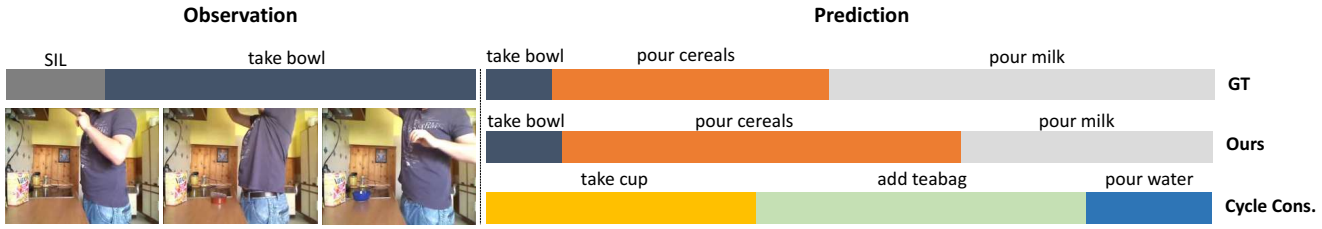
(b) Activity: Milk

Figure 4. **Cross-attention map visualization on Breakfast**. The horizontal and vertical axis indicates the index of the past frames and the action queries, respectively. The brighter color implies a higher attention score. We highlight a video frame with a yellow box where the attention score of the frame is highly activated. RGB frames are uniformly sampled from a video in this figure.

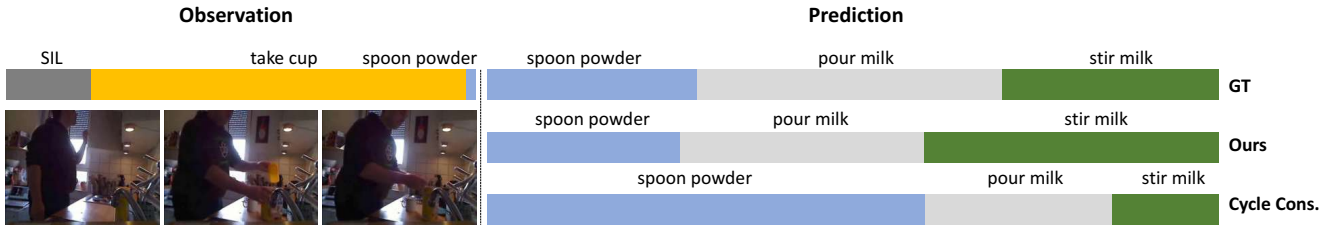
M	β ($\alpha = 0.3$)			
	0.1	0.2	0.3	0.5
6	29.95	26.47	25.46	23.27
7	30.03	27.94	27.00	24.23
8	**32.27**	**29.88**	27.49	**25.87**
9	31.24	28.65	26.87	24.95
10	31.32	28.86	**27.74**	25.01

Table 6. **Number of action queries.** We adjust the number of action queries M, showing that a sufficient number of action queries shows saturated performance.

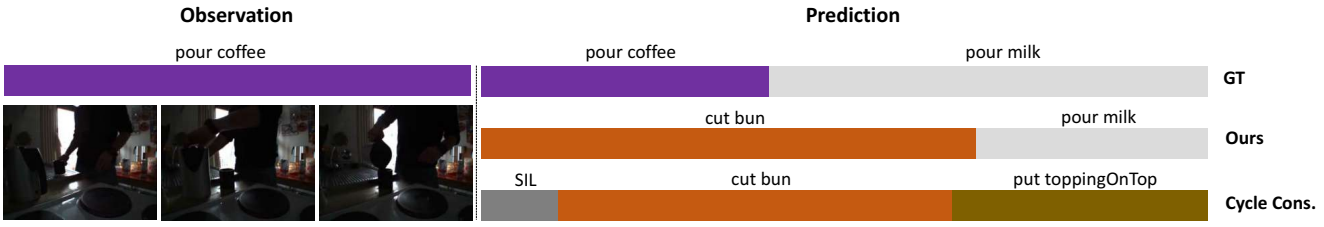
We find two interesting results from this experiment. First, our model learns to attend to visual features in the recent past, showing that the nearest frames provide crucial keys for predicting future actions. It is in alignment with the previous work [19, 36] that reflects the importance of the recent past in designing anticipation models. FUTR also attends to the recent past without any prior knowledge applied to the model. Second, we find that FUTR is trained to attend to important actions not only from the recent past, but also from the entire past frames. In Fig. 4a, essential frames with yellow boxes are detected by the queries with the high attention scores, providing contextual clues of the activity, *e.g.*'holding a pan' and 'taking an egg' actions in the 'fried egg' activity. Furthermore, action queries anticipating *NONE* class attend to the irrelevant features such as the beginning of the videos. The results show that FUTR effectively leverages long-term dependencies using the en-



(a) Activity: Cereal



(b) Activity: Milk



(c) Activity: Coffee

Figure 5. **Qualitative results on Breakfast**. Each subfigure visualizes the ground-truth labels and predicted results of the FUTR and the Cycle Cons. [13]. We set α as 0.3 and β as 0.5 in this experiment. We decode action labels and durations as frame-wise action classes. Each color in the color bar indicates an action label written above.

tire past frames regardless of the position, and also detects key frames of the given activity. More visualization results are shown in Supp. E.

5.6. Qualitative results

Figure 5 shows the qualitative results of FUTR and Cycle Cons. [13], evaluating on long-term action anticipation. In this experiment, we plot the prediction results based on the a sequence of predicted action label and corresponding duration. Each subfigure consists of observed frames, the ground-truth (GT) labels, and prediction results from the two models. Observed frames are uniformly sampled from videos. Figure 5a shows the importance of utilizing fine-grained features for action anticipation. FUTR anticipates 'take bowl' action from the observed frames, while Cycle Cons. model anticipates 'take cup' action missing fine-grained features, which leads to the error accumulation of the rest of the predictions. Figure 5c validates the robustness of parallel decoding on error accumulations from the previous predictions. Although the two models were wrong in the first anticipation, our model correctly predicts the following action label while Cycle Cons. generates false results during iterative predictions. The results also validates effectiveness of the proposed methods on various activities. See Supp. E for more qualitative results.

6. Conclusion

We have introduced an end-to-end attention neural network, FUTR, which leverages global relations of past and future actions for long-term action anticipation. The proposed method utilizes fine-grained visual features as input and anticipates future actions in parallel decoding, enabling accurate and faster inference. We have demonstrated the advantages of our method through extensive experiments on two benchmarks, achieving a new state of the art on Breakfast and 50Salads. While we have focused on long-term action anticipation in this work, we proposed an integrated model of action segmentation and anticipation in the same framework. We believe that FUTR suggested the direction that enhances comprehension of the actions in long-range videos.

Acknowledgements. This research was supported by NCSOFT, the IITP grant funded by MSIT (No.2019-0-01906, AI Graduate School Program - POSTECH), and the Center for Applied Research in Artificial Intelligence (CARAI) grant funded by DAPA and ADD (UD190031RD).

References

[1] Yazan Abu Farha and Juergen Gall. Uncertainty-aware anticipation of activities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (CVPRW)*, pages 0–0, 2019. 2, 5

[2] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5343–5352, 2018. 1, 2, 4, 5

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 6836–6846, 2021. 2

[4] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proc. International Conference on Machine Learning (ICML)*, July 2021. 2

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. European Conference on Computer Vision (ECCV)*, pages 213–229. Springer, 2020. 1, 2, 7

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 5

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. 2

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. International Conference on Learning Representations (ICLR)*, 2020. 2

[11] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 6824–6835, 2021. 2

[12] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3575–3584, 2019. 2, 5

[13] Yazan Abu Farha, Qiuhong Ke, Bernt Schiele, and Juergen Gall. Long-Term Anticipation of Activities with Cycle Consistency. In *Proc. German Conference on Pattern Recognition (GCPR)*. Springer, 2020. 1, 2, 3, 4, 5, 8

[14] Basura Fernando and Samitha Herath. Anticipating human actions by correlating past with the future with jaccard similarity measures. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13224–13233, 2021. 1, 2

[15] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 6252–6261, 2019. 1, 2

[16] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5562–5571, 2019. 1, 2

[17] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[18] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 2

[19] Qiuhong Ke, Mario Fritz, and Bernt Schiele. Time-conditioned action anticipation in one shot. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9925–9934, 2019. 1, 2, 5, 7

[20] Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. Relational self-attention: What's missing in attention for video understanding. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 2

[21] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 780–787, 2014. 1, 5

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 7

[23] Jinwoo Lee, Hyunsung Go, Hyunjoon Lee, Sunghyun Cho, Minhyuk Sung, and Junho Kim. Ctrl-c: Camera calibration transformer with line-classification. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 16228–16237, 2021. 2

[24] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065, 2020. 2

[25] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 6

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *Proc. International Conference on Learning Representations (ICLR)*, 2017. 5

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 5

[29] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[30] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 0–0, 2019. 1, 2

[31] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*, 2021. 2

[32] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Proc. Neural Information Processing Systems (NeurIPS)*, 34, 2021. 2

[33] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Proc. Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 2, 6

[34] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 754–763, 2017. 2, 5

[35] Debaditya Roy and Basura Fernando. Action anticipation using pairwise human-object interactions and transformers. *In Proc. IEEE Transactions on Image Processing*, 30:8116–8129, 2021. 1, 2

[36] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *Proc. European Conference on Computer Vision (ECCV)*, pages 154–171. Springer, 2020. 1, 2, 3, 5, 7

[37] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 862–871, 2019. 1, 2

[38] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738, 2013. 1, 5

[39] Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. Blockwise parallel decoding for deep autoregressive models. *Proc. Neural Information Processing Systems (NeurIPS)*, 31, 2018. 2

[40] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 7262–7272, 2021. 2

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proc. International Conference on Machine Learning (ICML)*, pages 10347–10357. PMLR, 2021. 2

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. Neural Information Processing Systems (NeurIPS)*, 30, 2017. 1, 2, 3, 6

[43] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. End-to-end dense video captioning with parallel decoding. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 6847–6857, 2021. 2

[44] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 568–578, 2021. 2

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018. 2

[46] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 22–31, 2021. 2

[47] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 558–567, 2021. 2

[48] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 13577–13587, 2021. 2

[49] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8746–8755, 2020. 2