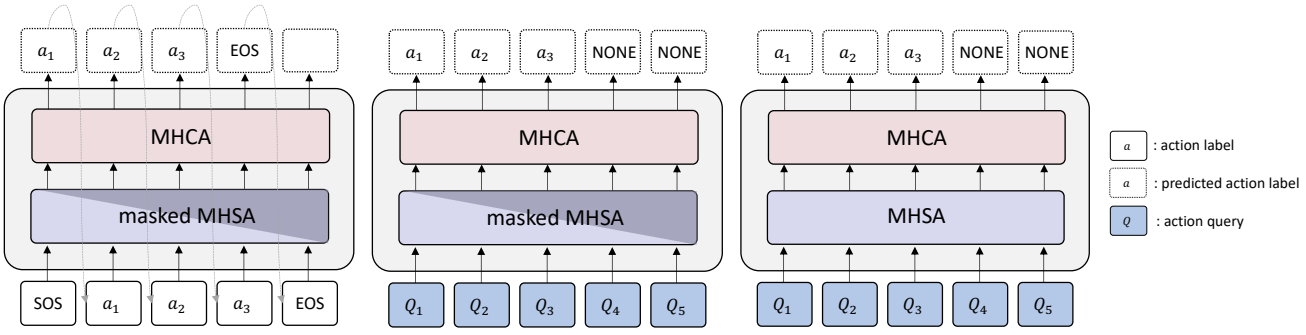
Future Transformer for Long-term Action Anticipation

–Supplementary Materials–

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



(a) FUTR-A (b) FUTR-M (c) FUTR

Figure S1. **FUTR variants with different decoding strategies.** (a) FUTR-A autoregressively anticipates future actions using the output action labels from the previous predictions as input and utilizes masked self-attention. (b) FUTR-M is equivalent to FUTR except for masked self-attention applied to action queries. (c) FUTR remove a causal mask in MHSA, where query attends to past and future queries in the sequence. FUTR-M and FUTR anticipates action and duration for each query simultaneously.

A. Experimental Details

In this section, we provide experimental details of the two experiments in Sec. 5.4.

Parallel decoding vs. autoregressive decoding. In Table 2, we compare our model with two FUTR variants, FUTR-A and FUTR-M. Two models have the same encoder but different decoders compared to FUTR, as illustrated in Fig. S1. FUTR-A anticipates the next action recurrently using a sequence of the predicted action labels as input in an autoregressive way. There exist two unique tokens: *SOS* and *EOS* in autoregressive decoding, each of which indicates the start and the end of the sequence, respectively. The decoder of FUTR-A takes *SOS* as the first input and predicts the next action label recursively until the model predicts *EOS*. FUTR-M takes a sequence of action queries as input and predicts action labels and durations in parallel with masked self-attention. Masked self-attention employs a causal mask to MHSA, which prevents attending to future actions. The core difference between FUTR and FUTR-M lies in the masked self-attention; action queries of FUTR-M only consider uni-directional temporal dependencies between action queries, while that of FUTR consider bi-directional temporal relations from the past and the future. We validate the effect of parallel decoding by comparing the three models.

Output structuring. In Table 4, we conduct experiments related to output structuring strategy. We introduce two variants of FUTR, FUTR-H and FUTR-S. FUTR-H is a DETR-like variant [2], where the ground truths are assigned to the outputs of the queries by the Hungarian matching [12]. Let us denote that $\boldsymbol{y}$ is the target set of future actions. The ground truth of the i^{th} index is defined by $\boldsymbol{y}_i = \{c_i, \boldsymbol{t}_i\}$, where c_i and $\boldsymbol{t}_i$ is the target action label and start-end window, respectively. Note that $\boldsymbol{y}$ is padded with *NONE* class to a size M. We also denote $\hat{\boldsymbol{y}}$ is the set of M predictions from the action queries. Since the Hungarian matching finds a pair-wise matching between the two set $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ minimizing the matching cost $\mathcal{L}^{\text{match}}$, we find the optimal permutation ζ from a set of permutation of M queries Z_M:

$$\hat{\zeta} = \text{argmin}_{\zeta \in Z_M} \sum_{i=1}^{M} \mathcal{L}^{\text{match}}(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_{\zeta(i)}). \tag{1}$$

We define matching cost as the sum of negative class probability and a window loss:

$$\mathcal{L}^{\text{match}}(\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i) = \mathbb{1}_{c_i \neq \varnothing}[-\hat{\boldsymbol{A}}_{\zeta(i),c_i} + \mathcal{L}^{\text{window}}(\boldsymbol{t}_i, \hat{\boldsymbol{t}}_{\zeta(i)})], \tag{2}$$

where $\mathbb{1}_{c_i \neq \varnothing}$ is an indicator function that sets to one where the gournd-truth action label is not *NONE*. We define a window loss $\mathcal{L}^{\text{window}}$ with L1 distance and IoU loss:

$$\mathcal{L}^{\text{window}}(\boldsymbol{t}_i, \hat{\boldsymbol{t}}_{\zeta(i)}) = \lambda^{\text{L1}} ||\boldsymbol{t}_i - \hat{\boldsymbol{t}}_{\zeta(i)}||_1 - \lambda^{\text{tiou}} \frac{|\boldsymbol{t}_i \cap \hat{\boldsymbol{t}}_{\zeta(i)}|}{|\boldsymbol{t}_i \cup \hat{\boldsymbol{t}}_{\zeta(i)}|}, \tag{3}$$

where $|.|$ and $\hat{\boldsymbol{t}}_{\zeta(i)}$ indicates temporal areas and the predicted start-end window. λ^{L1} and λ^{tiou} are weighting values of the two losses, which are 5 and 2, respectively. Note that starting and ending points of the temporal window $\boldsymbol{t}_i \in [0, 1]^2$ are bounded from 0 to 1. Finally, we define the Hungarian loss $\mathcal{L}^{\text{Hungarian}}$ by

$$\begin{aligned} &\mathcal{L}^{\text{Hungarian}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \\ &= \sum_{i=1}^{M} \sum_{j=1}^{K+1} [-\boldsymbol{A}_{i,j} log \hat{\boldsymbol{A}}_{\hat{\zeta}(i),j} + \mathbb{1}_{c_i \neq \varnothing} \mathcal{L}^{\text{window}}(\boldsymbol{t}_i, \hat{\boldsymbol{t}}_{\hat{\zeta}(i)})]. \end{aligned} \tag{4}$$

In training FUTR-H, we use the sum of the Hungarian loss and the action segmentation loss as our final loss.

B. Next Action Anticipation

We conduct an experiment of next action anticipation on EK55 (validation, RGB) following the previous experimental protocols [7, 8, 15].

Dataset. The Epic-Kitchens 55 dataset [4] is the large-scale dataset in first-person vision. The dataset comprises of 55 hours of recordings of 32 kitchens, including 39,594 action segments annotated with 125 verb, 331 noun, and 2,513 action classes.

Implementation details. FUTR can be applied to next action anticipation by simply setting the number of action query M to 1. We use two encoder layers and two decoder layers while setting the size of the hidden dimension D to 512. We do not include action segmentation loss in this experiment due to lack of segmentation annotations. Instead, we use additional a fully-connected layer applying to the output of the encoder layers X_{L_E} to predict features of the next frame. Then we apply a feature prediction loss of L2 distance between predicted features and the next frame similar to AVT [8]. We use AdamW optimizer [14] with a learning rate of 1e-5. We train our model for 40 epochs with a batch size of 32. We use the RGB feature embedded by TSN [16] in this experiment.

Results. The result is shown in Table S1. FUTR obtains 12.3%p at top-1 accuracy performing comparable with the state-of-the-art methods. We find that FUTR is also effective for next action anticipation, although the model is designed for long-term action anticipation.

method	backbone	top-1
RULSTM [7]	TSN	13.1
Temporal Agg. [15]	TSN	12.3
AVT [8]	TSN	13.1
FUTR (ours)	TSN	12.3

Table S1. **Performance comparison on EK55.** Although FUTR is designed for long-term action anticipation, the model is also effective in next action anticipation.

method	input	β ($\alpha = 0.2$)							
		0.01	0.02	0.03	0.05	0.1	0.2	0.3	0.5
AVT [8]	ViT	30.25	30.24	25.72	21.87	14.22	10.69	8.49	5.83
AVT [8]	I3D	26.13	22.03	20.24	13.52	17.84	13.20	9.01	4.61
FUTR (ours)	I3D	**51.16**	**44.34**	**40.84**	**40.56**	**39.43**	**27.54**	**23.31**	**17.77**
		β ($\alpha = 0.3$)							
AVT [8]	ViT	30.93	30.62	27.85	23.60	18.28	13.51	9.65	7.35
AVT [8]	I3D	31.56	35.17	33.12	24.17	14.92	12.79	10.38	5.81
FUTR (ours)	I3D	**42.20**	**38.67**	**38.56**	**36.44**	**35.15**	**24.86**	**24.22**	**15.26**

Table S2. **Performance comparison with AVT on 50Salads.** FUTR outperforms AVT especially when predicting long-term action sequences.

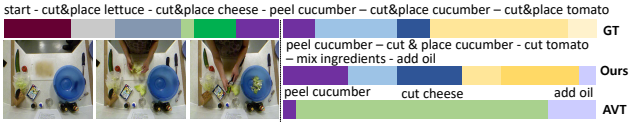


Figure S2. **Qualitative results of FUTR vs. AVT on 50Salads.** Each color in the color bar indicates an action label written above. AVT becomes inaccurate in the prolong predictions while our method is consistently accurate.

C. Additional Analysis

We conduct additional experiments for further analysis of the proposed method. In the following experiments, we evaluate our models on the Breakfast dataset with two observed ratios $\alpha \in \{0.2, 0.3\}$. Unless otherwise specified, all experimental settings are the same as those in Sec. 5.4.

Comparison with AVT. The core difference between AVT [8] and FUTR lies in the transformer architecture and the parallel decoding. While AVT uses a simple decoder that predicts the next action within a few seconds considering only the previous actions via masked self-attention, FUTR adopts a full-fledged decoder that predicts the whole sequence of actions in parallel by examining long-term relations of the actions via self-attention and cross-attention. AVT is also capable of anticipating long-term actions by unrolling the decoder iteratively, but it remains the drawbacks of error accumulation and slow inference speed. To validate our claim, we compare our method with AVT[1] on long-term action anticipation.

[1]We evaluate two 50Salads-pretrained AVT models: one is AVT with ViT, which the trained model is available on their official website (www.github.com/facebookresearch/AVT), and the other is AVT with I3D, which is trained by using their official codes.

γ	β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
0.25	24.36	21.66	20.62	20.10	30.31	27.53	25.45	23.19
0.5	25.26	22.99	22.10	21.37	31.14	28.25	25.91	23.85
1	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**

Table S3. **Effectiveness of global cross-attention.** We set observed ratio from the recent past as γ to show the effectiveness of exploiting global cross-attention at long-term action anticipation.

encoder		decoder		β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
type	Loc.	type	Loc.	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
-	-	learn	input	21.79	20.14	19.88	18.25	26.53	24.85	25.04	21.31
sine	input	learn	input	21.29	19.55	19.11	18.23	27.40	24.91	24.13	21.81
learn	input	learn	input	23.79	21.37	20.49	19.62	30.80	27.69	25.53	23.39
learn	attn.	learn	attn.	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**

Table S4. **Position embedding analysis.** Adding learnable positional embeddings before every attention layer performs the best.

model			β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
L^{E}	L^{D}	D	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
1	1	128	24.02	21.00	19.71	19.39	29.38	26.51	25.06	23.83
2	1	128	**27.70**	**24.55**	22.83	**22.04**	32.27	**29.88**	27.49	**25.87**
3	1	128	24.78	22.78	21.46	20.53	30.44	27.61	25.73	23.75
3	2	128	26.72	23.82	22.57	21.29	32.55	29.20	26.59	24.92
3	3	128	26.68	23.41	22.14	21.56	**33.06**	29.14	**28.12**	24.93
4	4	128	26.77	23.60	22.92	21.24	31.35	28.58	27.04	24.73
5	5	128	26.75	24.23	**23.55**	21.16	32.68	29.38	28.05	24.89
2	1	64	24.78	21.81	20.56	19.88	29.92	27.03	26.29	23.53
2	1	256	24.56	21.62	21.35	19.41	31.19	26.03	26.07	24.29
2	1	512	19.82	17.50	18.07	16.31	23.68	22.18	23.57	22.56

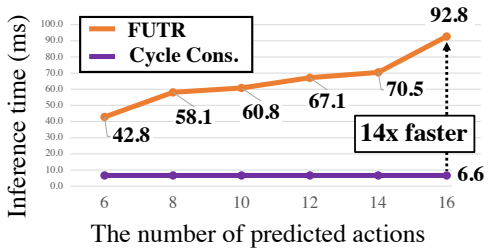
Table S5. **Model analysis.** We study the number of encoder layers L^{E}, the number of decoder layers L^{D}, and hidden dimension D of our model. We show the robustness of our methods over the number of layers, and find that the optimal hidden dimension D is 128.

loss	β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
L1	23.90	21.46	20.76	20.08	30.72	27.28	26.08	23.90
smooth L1	23.07	23.36	**22.88**	20.74	29.96	27.20	24.78	23.18
L2	**27.70**	**24.55**	22.83	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**

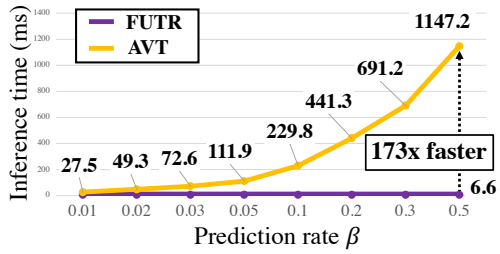
Table S6. **Duration loss analysis.** We find that utilizing L2 loss as our duration loss $\mathcal{L}^{\mathrm{duration}}$ shows better performance over L1 loss and Smooth L1 loss.

Table S2 shows the results of long-term action anticipation of both models. Since AVT is built for next action anticipation, we also adjust the prediction rate β ranging from 0.01 to 0.5. As β becomes smaller, the prediction results are closely related to next action anticipation. We find that AVT performs inferior to FUTR, especially when predicting long-term sequences. AVT is accurate for the early frames but becomes inaccurate in the prolonged predictions as shown in Fig. S2.

Inference time comparison. We compare inference time of FUTR to that of the Cycle Cons. [6] and AVT [8] in Fig. S3. The vertical axis indicates the inference time (ms) and the horizontal axis indicates the number of predicted actions for Cycle Cons. and the prediction rate β for AVT. The inference time of FUTR is consistently fast while that of Cycle Cons. and AVT linearly increases as the duration of the predicted sequence increases. From this experiment, we find that FUTR is 14$\times$ faster than Cycle Cons. when predicting 16 actions and 173$\times$ faster than AVT when β is set to 0.5.



(a) FUTR vs. Cycle Cons. [6]



(b) FUTR vs. AVT [8]

Figure S3. **Inference time comparison with other methods [6, 8]**. Inference time of other methods linearly increases as the duration of the future action sequence to be predicted increases, *i.e.*, the number of predicted actions or prediction rate β, increases, while that of FUTR is consistently fast.

The results show the efficiency of the parallel decoding for long-term action anticipation.

Effectiveness of global cross-attention. To evaluate the importance of modeling long-term dependencies between the observed frames and the action queries during the decoding stage, we measure the performance by gradually increasing the number of cross-attended frames from the most recent frame to the farthest one. For notational simplicity, we establish the ratio of the cross-attended frames γ ranging from 0.25 to 1, adjusting the number of observed frames starting from the recent past; the cross-attention layer in the decoder only attends to the most recent γT^{O} frames during the decoding stage. Note that $\gamma = 1$, our default setting, indicates that the decoder attends to the whole video frames to anticipate actions.

Table S3 summarizes the results of the effect of global attention in the cross-attention layers. As we gradually increase the γ from 0.25 to 1, the overall accuracy significantly increases by 2.0-3.3%p. This demonstrates the efficacy of modeling global interactions between the observed frames in the past and the action queries in the future for long-term action anticipation.

Position embedding analysis. In Table S4, we investigate various combinations of different types and locations of the positional embeddings. From the 1^{st} to the 3^{rd} rows, we compare three types of position embeddings in the encoder layers: none, sinusoidal, and learnable position embeddings. Here, we fix the position embedding of the decoder as learnable embedding, which is added before going into the attention layers. We find that using learnable position embeddings in the encoder is effective. Then we change the location of the position embeddings to be learned in the attention layers, obtaining additional accuracy improvements. In this experiment, we find that position embedding learned at the attention layer is effective for our model.

Model analysis. Table S5 summarizes the results of the model ablations, according to the number of encoder layers L^{E}, the number of decoder layers L^{D}, and hidden dimension D. We find that the performance is saturated when we use more than two encoder layers and one decoder layer. Thus we set $L^{\mathrm{E}} = 2$ and $L^{\mathrm{D}} = 1$ as our default number of encoder layers and decoder layers, respectively. We also evaluate our model by varying the channel dimension D and find that setting D to 128 performs the best; too small D restricts the representation power of the model while too large D causes overfitting problems.

Duration loss analysis. In Table S6, we evaluate our duration loss $\mathcal{L}^{\mathrm{duration}}$ of Eq. (14). Instead of L2 loss, we use L1 loss and Smooth L1 loss [9] in this experiment. The results show that applying L2 loss shows better performance over the L1 loss and smooth L1 loss. Since L2 loss is more robust to outliers than L1 loss and smooth L1 loss, we find that applying L2 loss is effective in the proposed method.

D. Additional Results

In Tables S7-S11, we provide the overall experimental results in Sec. 5.4 with two observation ratios $\alpha \in \{0.2, 0.3\}$. We find that overall experimental tendencies with the two observation ratios are similar although the experimental setup with $\alpha = 0.2$ is more challenging.

E. Qualitative Results

We plot additional visualization results of the cross attention map of the decoder in Fig. S4. Each subfigure contains sampled frames from videos and attention map visualizations below. We also highlight the frames with the yellow box where corresponding attention scores are highly activated. From this experiment, we find that action query in our method attends dynamically to the input visual features, which utilize fine-grained visual features from the entire past visual features.

F. Discussion

We have proposed an end-to-end attention network for long-term action anticipation, which effectively leverages global interactions in videos enabling accurate and fast inference for long-term action anticipation. We have demonstrated the effectiveness of the FUTR through extensive experiments, but there exists much room for improvement. First, the efficiency of FUTR could be further improved. For example, linear attention mechanisms [3, 11, 17] or

method	AR	causal mask	β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
			0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
FUTR-A	✓	✓	20.31	18.37	17.69	16.31	25.43	24.02	23.43	21.08
FUTR-M	-	✓	25.27	22.41	21.39	20.86	31.82	28.55	26.57	24.17
FUTR	-	-	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**

Table S7. **Parallel decoding vs. autoregressive decoding.**

encoder	decoder	β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
		0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
LSA	LSA	21.97	19.20	18.04	18.19	27.70	24.39	23.18	21.60
GSA	LSA	25.25	22.88	21.09	19.73	30.15	27.51	25.62	23.28
LSA	GSA	22.99	20.39	19.15	18.60	28.37	25.08	24.03	22.28
GSA	GSA	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**

Table S8. **Global self-attention (GSA) vs. local self-attention (LSA).**

method	GT Assign.	regression	β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
			0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
FUTR-S	sequential	start-end	23.87	19.86	18.58	18.05	29.15	25.51	24.20	21.43
FUTR-H	Hungarian	start-end	22.05	20.18	18.63	17.31	25.26	23.85	22.63	21.45
FUTR	sequential	duration	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**

Table S9. **Output structuring.**

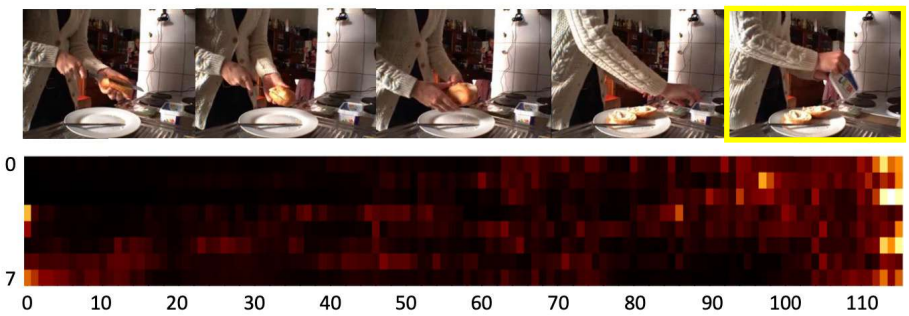
loss			β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
$\mathcal{L}^{\text{seg}}$	$\mathcal{L}^{\text{action}}$	$\mathcal{L}^{\text{duration}}$	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
-	✓	✓	25.60	22.13	21.95	20.86	28.31	25.85	24.91	22.50
✓	✓	✓	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	**27.49**	**25.87**

Table S10. **Loss ablations.**

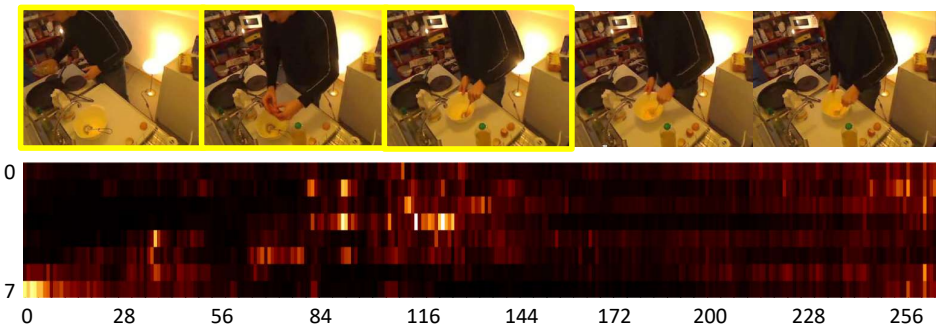
M	β ($\alpha = 0.2$)				β ($\alpha = 0.3$)			
	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
6	24.63	21.74	20.99	19.67	29.95	26.47	25.46	23.27
7	24.40	22.13	21.59	20.28	30.03	27.94	27.00	24.23
8	**27.70**	**24.55**	**22.83**	**22.04**	**32.27**	**29.88**	27.49	**25.87**
9	24.21	22.47	21.56	20.94	31.24	28.65	26.87	24.95
10	24.61	21.79	20.90	19.91	31.32	28.86	**27.74**	25.01

Table S11. **Number of action queries.**

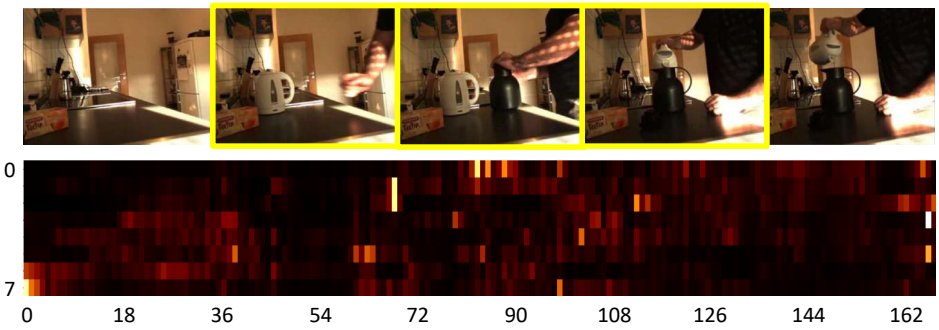
sparse attention mechanisms [1,20] could reduce both computation and memory complexity of FUTR, enabling efficient long-term video understanding. Second, considering that our encoder is a separate action segmentation network, the proposed architecture is a unified network that can handle both long-term action anticipation and action segmentation task at once. Although we focus on long-term action anticipation in this paper, we can integrate our models with other action segmentation methods [5,10,13,18,19] to solve both action segmentation and long-term action anticipation task altogether in the same framework. We leave this as our future work.



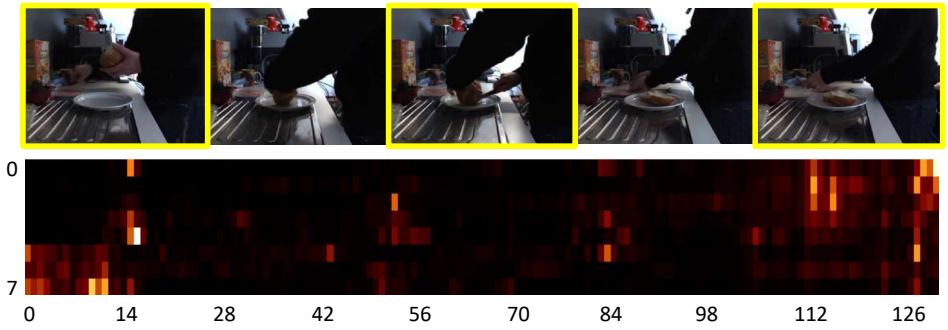
(a) Activity: Sandwich



(b) Activity: Scramble egg

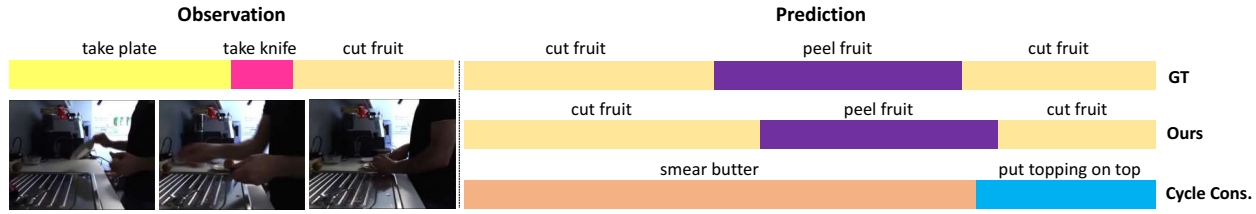


(c) Activity: Tea

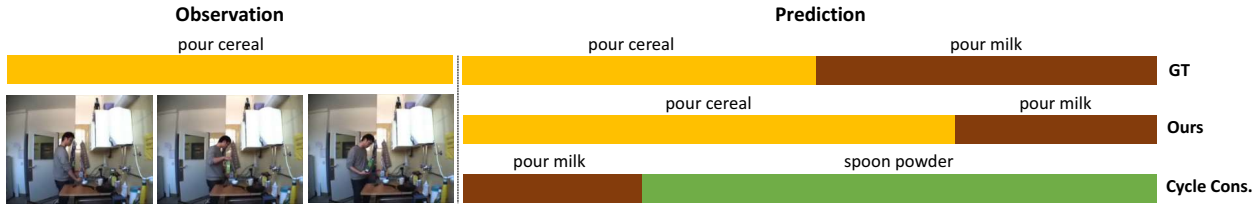


(d) Activity: Sandwich

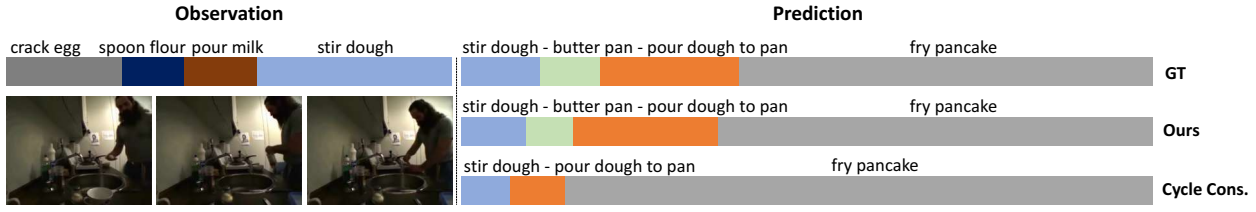
Figure S4. **Cross-attention map visualization on Breakfast**. The vertical and horizontal axis indicates the decoder queries and observed frames, respectively. The brighter color indicates a higher attention score. RGB frames above the attention map are sampled uniformly from the video. We emphasize the frames with high attention scores with yellow box and other frames are uniformly sampled. Best viewed in color.



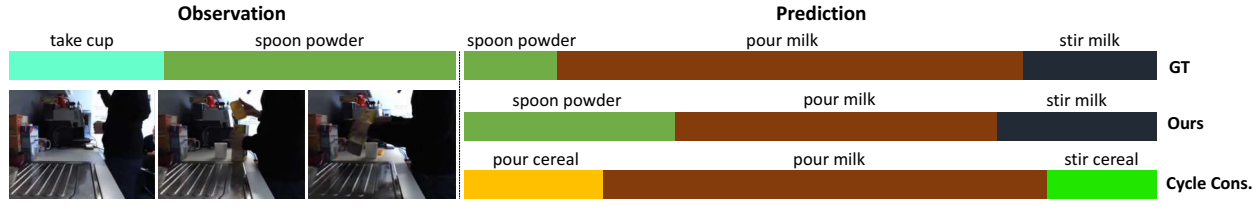
(a) Activity: salad



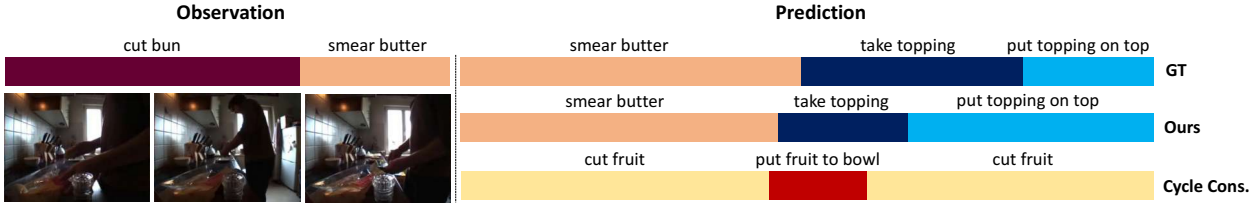
(b) Activity: cereal



(c) Activity: pancake



(d) Activity: milk



(e) Activity: pancake

Figure S5. **Qualitative results on Breakfast**. Each subfigure visualizes the ground truth label and the prediction results of the FUTR and cycle consistency model proposed from Farha et al. [6]. We set α as 0.3 and β as 0.5 in this experiment. We decode action labels and durations to the frame-wise action classes. Each color in the color bar indicates an action label written above. Best viewed in color.

References

[1] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 5

[2] 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

[3] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 4

[4] 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

[5] 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. 5

[6] 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. 3, 4, 7

[7] 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. 2

[8] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 4

[9] Ross Girshick. Fast r-cnn. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 4

[10] Yuchi Ishikawa, Seito Kasai, Yoshimitsu Aoki, and Hirokatsu Kataoka. Alleviating over-segmentation errors by detecting action boundaries. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2322–2331, 2021. 5

[11] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proc. International Conference on Machine Learning (ICML)*, pages 5156–5165. PMLR, 2020. 4

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

[13] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 156–165, 2017. 5

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 2

[15] 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. 2

[16] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. European Conference on Computer Vision (ECCV)*, 2016. 2

[17] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 4

[18] Zhenzhi Wang, Ziteng Gao, Limin Wang, Zhifeng Li, and Gangshan Wu. Boundary-aware cascade networks for temporal action segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 34–51. Springer, 2020. 5

[19] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *Proc. British Machine Vision Conference (BMVC)*, 2021. 5

[20] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 5