Predicting the Next Action by Modeling the Abstract Goal

Debaditya Roy[1][0000−0002−8779−1241] and Basura Fernando[1,2][0000−0002−6920−9916]

[1] Institute of High-Performance Computing, Agency for Science, Technology and Research, Singapore
roy_debaditya@ihpc.a-star.edu.sg
[2] Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore
fernando_basura@cfar.a-star.edu.sg

Abstract. The problem of predicting human actions from observed videos is an inherently uncertain one. We present an action anticipation model that leverages latent goal information to reduce the uncertainty in future predictions. We develop a latent variable representing goal information called abstract goal which is conditioned on observed sequences of visual features for action anticipation. We design the abstract goal as a distribution whose parameters are estimated using a variational recurrent model. We sample multiple candidates for the next action and use goal consistency criterion to determine the best candidate that follows from the abstract goal. Our method obtains impressive results on the very challenging Epic-Kitchens55 (EK55) and good results in Epic-Kitchens100 (EK100) datasets. Code is available at https://github.com/LUNAProject22/Abstract_Goal

Keywords: Action Anticipation · Stochastic Modeling· Variational Inference

1 Introduction

Anticipating human actions from videos has significant relevance across various domains, including but not limited to human-robot collaboration, intelligent domiciles, assistive robotics, and wearable virtual assistants. Specifically, egocentric videos, which capture the actions of the individual wearing the camera, represent a valuable resource for the development of intelligent assistants capable of forecasting the wearer's future actions and providing tailored assistance accordingly. A fundamental challenge in action anticipation lies in the inherent uncertainty surrounding future predictions. Human behavior is predominantly steered by individual goals or intentions, thus guiding the sequence of actions performed. Consequently, incorporating goal information holds promise for mitigating such uncertainty in forecasting future actions. For example, with information about the goal *wash pan*, a model can predict that *take pan* will be followed by *rinse pan* and not *put pan on stove.*

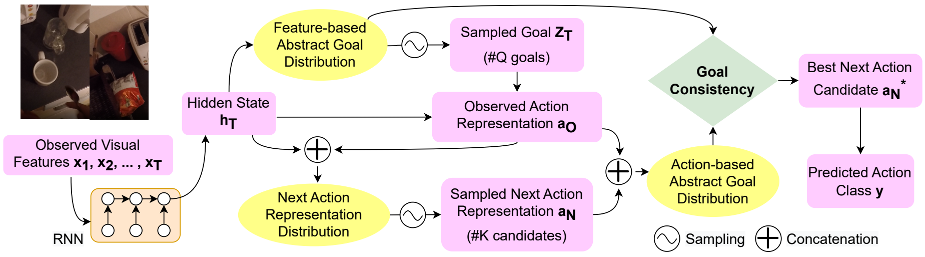
Feature-based Abstract Goal Distribution
Sampled Goal z_T (#Q goals)
Goal Consistency
Best Next Action Candidate a_N^*
Hidden State h_T
Observed Action Representation a_O
Observed Visual Features $x_1, x_2, \dots, x_T$
RNN
Next Action Representation Distribution
Sampled Next Action Representation a_N (#K candidates)
Action-based Abstract Goal Distribution
Predicted Action Class y
Sampling
Concatenation

Fig. 1. Model design for abstract goal-based action anticipation. Yellow ellipses represent distributions and pink boxes represent various variables of the model.

Goal and intentions have been adopted in some recent works for effective action anticipation [27,21,30]. In this paper, we make use of a stochastic method [4,9] for latent goal modeling to improve action anticipation that goes beyond the deterministic latent goal representation in [27]. We propose to learn a new latent variable called abstract goal as a latent distribution as shown in Figure 1. We use two types of abstract goal distributions when predicting the next action in the sequence. The first abstract goal distribution is learned using the observed visual features and a stochastic recurrent neural network [4] which we call "feature-based abstract goal" distribution. Furthermore, we design an "action-based abstract goal" distribution using the next action representation distribution and the observed action representation. We sample multiple next-action-representation candidates and use the goal consistency criterion to find the most likely next action–see Figure 1. The action that is most likely to happen in the future ("next best action") is the one that maximizes consistency between the two latent abstract goal distributions. During learning, we use goal consistency as a loss function to obtain a model informed of human behavior, i.e. the sequences of actions. Such a mechanism is not present in previous stochastic approaches [1,22,21] which only minimizes KL divergence between prior and posterior latent distribution to obtain the best future actions. Also, we introduce a goal consistency measure to choose the best next action candidate rather than mean or median sampling used in [1,22]. We show that goal consistency has the biggest impact on action anticipation. Our approach yields improvements when predicting the next action in unscripted activities on the Epic-Kitchens55 (EK55). Our contributions are:

- A new latent variable called abstract goal using a stochastic recurrent model that uses two latent distributions for the observed and the next action and enforces consistency among them to effectively predict the next action.
- A novel goal consistency term that measures how well a plausible future action (next action) aligns with abstract goal distributions.

2 Related work

Research in action anticipation has gained popularity in recent years thanks to progress in datasets [6] and challenges [5]. The activity label of the entire action sequence is used to anticipate the next action in [29]. In [27], observed features are used to obtain a fixed latent goal from visual features. [3] conceptualizes goals as the visual outputs of a sequence of actions. They predict each action in the sequence based on its relative closeness to the goal as compared to the previous action. [19] propose to use an external memory bank to store prototypes of the overall activity and contrastive learning augmented with the memory bank for forecasting the next action.

Predicting Future features for Anticipation. In [10] authors show that LSTM can be unrolled for multiple time steps to predict future features can be used to accurately predict the next action. In [26], Human-object interactions are encoded as features and fed to a transformer encoder-decoder to predict the features of future frames and the corresponding future actions. Authors in [18] estimate spatial attention maps of future human-object interactions to predict the next action. In [32], authors propose to summarize long-range sequences by processing smaller temporal sequences and caching them in memory as context and using the context for action anticipation. In [12], a real-time action anticipation framework is presented using a two-stage transformer with reduced parameters that is trained for future feature prediction and action anticipation. Due to the lightweight nature of their model, the action inference is performed in real-time. In [16], temporal features are computed using time-conditioned skip connections to anticipate the next action. In [33], an RNN is used to generate the intermediate frames between the observed frames and the anticipated action. In [13], every frame is represented using a Visual Transformer (ViT) [7] and combined using a temporal transformer to predict future features and action labels. Authors in [34], train a transformer model to predict the next action by reducing the amount of observed future available during learning from fully available to completely absent. Authors in [28] model interactions using cross-attention between humans and object visual features using a spatio-temporal visual transformer and use the modeled interaction to predict the next action.

Long-term forecasting. In [2], future actions and their duration are predicted autoregressively using an RNN with observed action labels as input. In [2,20], RNNs are used to predict future actions conditioned on observed action labels. Latent distributions are used in literature to encode the observed action and duration in [1,22]. In [1], a sample from the latent distribution of observed action is combined with previously predicted action in a decoder to predict the multiple next actions and their duration. In [22], two decoders are used to predict the action labels and duration separately. The action decoder uses the action labels in the observed video as input while the duration prediction decoder uses the duration of actions. Similarly, in [14], a transformer is used to encode past actions and duration while another transformer decoder is used to predict both future actions and their duration. In [24], authors use two transformer encoders for segment-level and long-term encoding and a decoder that fuses both encoder inputs to predict future actions. In [21], goal labels and observed features are used as input to a conditional variational encoder to predict future actions. In [37], a large language model is prompted with observed actions and narrations to predict future actions.

Correlating past and future. In [23], authors model the transition between the visual features of the observed and the next action to generate the next action features. A similar action anticipation model that correlates past observed features with the future using Jaccard vector similarity is presented in [8]. In [16], time-conditioned skip connections are used to generate features for predicting future actions at different anticipation time in the future. In [11], authors propose a neural memory network to compare an input (spatial representation or labels) with the existing memory content to predict future action labels. Similarly, in [25], authors propose an action anticipation framework with a self-regulated learning process. A counterfactual reasoning is used to improve action anticipation in [36]. Our approach correlates the past and future by enforcing goal consistency between the two abstract goal distributions computed using observed features and the next action.

3 Action anticipation with abstract goals

In this section, we explain our model design outlined in Figure 1. At first, we explain how to compute the feature-based abstract goal distribution in Section 3.1. Then, we describe how to obtain next action candidates and action-based abstract goal with respect to these candidates in Section 3.2 and 3.3, respectively. We then explain the goal consistency criterion used to obtain the best next action candidate in Section 3.4. Finally, we describe the various loss functions to train our model in Section 3.5.

3.1 Observed Feature-based abstract goal representation

In this section, we describe how to generate *feature-based abstract goal* representation using variational recurrent neural network (VRNN) framework [4,9]. Let us denote the observed feature sequence by $\mathbf{x}_1, \mathbf{x_2}, \cdots, \mathbf{x}_T$ where $\mathbf{x}_t \in \mathbb{R}^{d_f}$. Following standard VRNN, a Gaussian distribution $q_t(\mathbf{z}_t|\mathbf{x}_{1:t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t,prior}, \boldsymbol{\sigma}_{t,prior})$ is used to model the prior distribution of the abstract goal ($\mathbf{z}_t$) given the observed feature sequence $\mathbf{x}_{1:t-1}$. The parameters $\boldsymbol{\mu}_{t,prior}, \boldsymbol{\sigma}_{t,prior} \in \mathbb{R}^{d_z}$ are estimated using the hidden state of the RNN ($\mathbf{h}_{t-1} \in \mathbb{R}^{d_h}$) learned from the previous $t-1$ features, i.e. $(\boldsymbol{\mu}_{t,prior}, \boldsymbol{\sigma}_{t,prior}) = \phi_{prior}(\mathbf{h}_{t-1})$. Note that $\phi_{prior} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$ refers to two separate MLPs, one to obtain $\boldsymbol{\mu}_{t,prior}$ and another with $softplus$ activation to estimate the standard deviation ($\boldsymbol{\sigma}_{t,prior}$). Unless otherwise specified, all MLPs are two layered neural networks with ReLU activation.

The posterior distribution of the abstract goal $r(\mathbf{z}_t|\mathbf{x}_{1:t}) \sim \mathcal{N}(\boldsymbol{\mu}_{t,pos}, \boldsymbol{\sigma}_{t,pos})$ computes the effect of observing the incoming new feature $\mathbf{x}_t$. The parameters of posterior distribution r are computed as follows:

$$(\boldsymbol{\mu}_{t,pos}, \boldsymbol{\sigma}_{t,pos}) = \phi_{pos}([\phi_x(\mathbf{x}_t), \phi_h(\mathbf{h}_{t-1})]), \tag{1}$$

where $\phi_{pos} : \mathbb{R}^{2\times d_z} \to \mathbb{R}^{d_z}$, $\phi_x : \mathbb{R}^{d_f} \to \mathbb{R}^{d_z}$, $\phi_h : \mathbb{R}^{d_h} \to \mathbb{R}^{d_z}$ are linear layers and $[\cdot,\cdot]$ represents vector concatenation. We use the reparameterization trick [17] to sample an abstract goal ($\mathbf{z}_t \in \mathbb{R}^{d_z}$) from the prior distribution $q(\mathbf{z}_t|\mathbf{x}_{1:t-1})$ as follows:

$$\mathbf{z}_t = \boldsymbol{\mu}_{t,prior} + \boldsymbol{\sigma}_{t,prior} \odot \boldsymbol{\epsilon}, \tag{2}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\mathbf{1}) \in \mathbb{R}^{d_z}$ is a standard Gaussian distribution. Then sampled $\mathbf{z}_t$ is used to obtain the next hidden state of the RNN[3] as follows:

$$\mathbf{h}_t = RNN(\mathbf{h}_{t-1}, [\phi_x(\mathbf{x}_t), \phi_z(\mathbf{z}_t)]), \forall t \in 1, \cdots, T \tag{3}$$

where $\phi_z : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$ acts as a feature extractor over $\mathbf{z}_t$. The sampled abstract goal ($\mathbf{z}_t$) can be used to reconstruct (or generate) the feature sequence as done in VRNN framework [4,9]. However, we use it to represent feature-based abstract goal. Our intuition comes from the fact that humans derive action plans from goals, and videos are a realization of this action plan. Therefore, by construction, goal determines the video (feature evolution in our case). Interestingly, as the abstract goal latent variable encapsulates the video feature generation process, by analogical similarity, we make the proposition that latent variable ($\mathbf{z}_t$) represents the notion of feature-based abstract goal.

Therefore, we denote the "feature-based abstract goal distribution" as follows:

$$p(\mathbf{z}_T) = q(\mathbf{z}_T|\mathbf{x}_{1:T-1}). \tag{4}$$

The abstract goal distribution *represents all abstract goals with respect to a particular observed feature sequence*. Any observed action may lead to more than one goal. Our abstract goal representation captures these variations.

3.2 Action representations

Human actions are causal in nature and the next action in a sequence depends on the earlier actions. For example, *washing vegetables* is succeeded by *cutting vegetables* when the goal is "making a salad". We capture the causality between observed and next actions using "the observed action representation" and the "next action representation". We obtain the **observed action representation** ($\mathbf{a}_O$) using feature-based abstract goal and the hidden state of RNN as follows:

$$\mathbf{a}_O = \phi_O([\phi_z(\mathbf{z}_T), \phi_h(\mathbf{h}_T)]). \tag{5}$$

Here $\phi_O : \mathbb{R}^{2\times d_z} \to \mathbb{R}^{d_h}$ and $\mathbf{z}_T$ is sampled from the abstract goal distribution $p(\mathbf{z}_T)$ using Equation 2.

Then we obtain the distribution of **next action representation** ($\mathbf{a}_N$) conditioned on the hidden state of the RNN and the observed action representation denoted by $p(\mathbf{a}_N|\mathbf{h}_T, \mathbf{a}_O)$. The reason for modeling next action representation as a distribution conditioned on hidden state and the observed action representation is two-fold. First, a particular observed action may lead to different

[3] Our RNN is a standard GRU cell.

next actions depending on the context and goal. Note that in our model, both observed action representation $\mathbf{a}_O$ and the RNN hidden state $\mathbf{h}_T$ depend on the feature-based abstract goal representation. Second, there can be variations in human behavior when executing the same task. The next action representations are generated using a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}^2_{\mathbf{a}_N})$ where $\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}_{\mathbf{a}_N} \in \mathbb{R}^{d_z}$ The parameters of next action distribution are estimated as

$$p(\mathbf{a}_N|\mathbf{h}_T, \mathbf{a}_O) \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}^2_{\mathbf{a}_N}), \tag{6}$$

where $(\boldsymbol{\mu}_{\mathbf{a}_N}, \boldsymbol{\sigma}_{\mathbf{a}_N}) = \phi_N([\phi_h(\mathbf{h}_T), \phi_a(\mathbf{a}_O)])$. The mapping network $\phi_a : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$ and $\phi_N : \mathbb{R}^{2\times d_z} \rightarrow \mathbb{R}^{d_z}$ are two separate MLPs. Now we sample multiple next action representations from the next action representation distribution using the reparameterization trick as in Equation 7,

$$\mathbf{a}_N = \boldsymbol{\mu}_{\mathbf{a}_N} + \boldsymbol{\sigma}_{\mathbf{a}_N} \odot \boldsymbol{\epsilon}, \tag{7}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \in \mathbb{R}^{d_z}$ is a standard Gaussian distribution.

3.3 Action-based abstract goal representation

Now, we obtain *action-based abstract goal* from observed and next action representations using generative variational framework [17]. The distribution for action-based abstract goal is modeled with a Gaussian distribution conditioned on the next action representation denoted by $q(\mathbf{z}_N|\mathbf{a}_N)$ whose parameters are computed as $q(\mathbf{z}_N|\mathbf{a}_N) \sim \mathcal{N}(\boldsymbol{\mu}_{Nq}, \boldsymbol{\sigma}_{Nq})$ where $(\boldsymbol{\mu}_{Nq}, \boldsymbol{\sigma}_{Nq}) = \phi_{Nq}(\phi_a(\mathbf{a}_N))$ and $\boldsymbol{\mu}_{Nq}, \boldsymbol{\sigma}_{Nq} \in \mathbb{R}^{d_z}$ and $\phi_{Nq} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$ is implemented with two MLPs. On the other hand, parameters of the action-based abstract goal distribution (r) conditioned on both observed and next action representation are given as $r(\mathbf{z}_N|\mathbf{a}_N, \mathbf{a}_O) \sim \mathcal{N}(\boldsymbol{\mu}_{Nr}, \boldsymbol{\sigma}_{Nr})$ whose parameters are estimated as:

$$(\boldsymbol{\mu}_{Nr}, \boldsymbol{\sigma}_{Nr}) = \phi_{Nr}([\phi_a(\mathbf{a}_N), \phi_a(\mathbf{a}_O)]) \tag{8}$$

where $\boldsymbol{\mu}_{Nr}, \boldsymbol{\sigma}_{Nr} \in \mathbb{R}^{d_z}$ and $\phi_{Nr} : \mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_z}$ is a dual headed MLP. Finally, the **action-based abstract goal** distribution for the next action $p(\mathbf{z}_N)$ is given by the distribution

$$p(\mathbf{z}_N) = q(\mathbf{z}_N|\mathbf{a}_N). \tag{9}$$

We use both feature-based and action-based abstract goal representation to find the *best candidate for next action* as explained in next section. It should be noted that while the $q(\mathbf{z}_N|\mathbf{a}_N)$ only depends on next action representation and $r(\mathbf{z}_N|\mathbf{a}_N, \mathbf{a}_O)$ depends on both observed and next action representation. As $r()$ has more evidence compared to $q()$, $r()$ acts as the posterior distribution in our modeling.

3.4 Next action anticipation with goal consistency

Given a sampled feature-based abstract goal $\mathbf{z}_T$, we select the best next action representation $\mathbf{a}^*_N$ using the divergence between $p(\mathbf{z}_T)$ distribution (eq. 4) and

$p(\mathbf{z}_N)$ distribution (eq. 9). We call this divergence as the **goal consistency criterion**. For a given $\mathbf{z}_T$, observed action $\mathbf{a}_O$ and the next sampled action $\mathbf{a}_N$, the goal consistency criterion is derived from the average of KL-divergence $D_{KL}(p(\mathbf{z}_T)||p(\mathbf{z}_N)$ and $D_{KL}(p(\mathbf{z}_N)||p(\mathbf{z}_T))$ as follows:

$$D(\mathbf{a}_N) = \frac{D_{KL}(p(\mathbf{z}_T)||p(\mathbf{z}_N)) + D_{KL}(p(\mathbf{z}_N)||p(\mathbf{z}_T))}{2}. \tag{10}$$

We choose the best next action candidate (i.e. the anticipated action candidate representation) $\mathbf{a}_N^*$ that minimizes the goal consistency criterion. The rationale is that the best anticipated action should have an action-based abstract goal representation $p(\mathbf{z}_N)$ that aligns with the feature-based abstract goal distribution $p(\mathbf{z}_T)$. We use the following algorithm to find the best next action candidate $\mathbf{a}_N^*$.

Algorithm 1 Best next action selection

```
1: Sample feat-based abstract goal $\mathbf{z}_t$ from eq. 4 $\rightarrow \mathbf{z}_t \sim q_t(\mathbf{z}_t|\mathbf{x}_{1:t-1})$
2: Get observed action representation $\mathbf{a}_O$ (eq. 5)
3: Get next action representation distribution $p(\mathbf{a}_N|\mathbf{h}_t, \mathbf{a}_O)$ (eq. 6)
4: Sample $K$ next action representations $\mathcal{N} = \{\mathbf{a}_N^1, \cdots \mathbf{a}_N^K\} \sim p(\mathbf{a}_N|\mathbf{h}_t, \mathbf{a}_O)$
5: Best next action $\mathbf{a}_N^* = \arg\min_{\mathbf{a}_N^k \in \mathcal{N}} D(\mathbf{a}_N^k); k \in \{1, \cdots, K\}$
```

Finally, we predict the anticipated action from the selected next action representation as $\hat{\mathbf{y}} = \phi_c(\mathbf{a}_N^*)$. where $\phi_c : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_c}$ is the MLP classifier and $\hat{\mathbf{y}}$ is the class score vector. It should be noted that in Algorithm 1, we sample only one feature-based abstraction goal in line 1 of the algorithm. However, during training we sample Q number of feature-based abstraction goals and for each of them we sample K number of next action representations. In this case, we select the best candidate from all $K \times Q$ next action representation candidates using Equation 10. Therefore, the next best action is consistent and does not rely too much on sampling as long as we sample sufficient candidate next actions.

Even if the feature-based abstract goal $P(\mathbf{z}_T)$ is obtained from VRNN framework [4,9], the formulation of action representations $\mathbf{a}_O$ and $\mathbf{a}_N$, action-based abstract goal $P(\mathbf{z}_N)$ and goal consistency criterion is drastically different from [1,22]. In [27], goal consistency is defined between latent goals before and after the action using a hard threshold. Instead, our goal consistency is a symmetric KL divergence between $p(\mathbf{z}_T)$ and $p(\mathbf{z}_N)$ distributions which aims to align the two abstract goal distributions. This also results in a massive improvement in next action anticipation performance as shown in the experiments.

3.5 Loss functions and training of our model

Our anticipation network is trained using a number of losses. In contrast to prior stochastic methods [22,1,21], we introduce three KL divergence losses, based on

a) feature-based abstract goal ($\mathcal{L}_{OG}$), b) action-based abstract goal ($\mathcal{L}_{NG}$), and c) goal-consistency (L_{GC}). The first loss function is used to learn the parameters of the feature-based abstract goal distribution. We compute the KL-divergence between the conditional prior $q(\mathbf{z}_t|\mathbf{x}_{1:t-1})$ and posterior $r(\mathbf{z}_t|\mathbf{x}_{1:t})$ distributions for every feature in the observed feature sequence and minimize the sum given as follows $\mathcal{L}_{OG} = \sum_{t=1}^{T} D_{KL}(r(\mathbf{z}_t|\mathbf{x}_{1:t})||q(\mathbf{z}_t|\mathbf{x}_{1:t-1}))$ and we call this **observed goal loss**. This loss is based on the intuition that the abstract goal should not change due to a new observed feature.

Our second loss arises when we learn the action-based abstract goal distribution. We compute the KL-divergence between $r(\mathbf{z}_N|\mathbf{a}_N^*, \mathbf{a}_O)$ and $q(\mathbf{z}_N|\mathbf{a}_N^*)$ distributions of action-based abstract goal distributions as $\mathcal{L}_{NG} = D_{KL}(r(\mathbf{z}_N|\mathbf{a}_N^*, \mathbf{a}_O)$ $||q(\mathbf{z}_N|\mathbf{a}_N^*))$. We denote the corresponding best action-based abstract goal distribution by $p(\mathbf{z}_N^*) = q(\mathbf{z}_N|\mathbf{a}_N^*)$. The intuition is same as before, the goal should not change because of the next best action $\mathbf{a}_N^*$.

Furthermore, the feature-based and action-based abstract goal distributions should be aligned with respect to the selected next best action $\mathbf{a}_N^*$. Therefore, we minimize the symmetric KL-Divergence between the feature-based and best-action-based abstract goal distribution as follows:

$$\mathcal{L}_{GC} = \frac{D_{KL}(p(\mathbf{z}_T)||p(\mathbf{z}_N^*) + D_{KL}(p(\mathbf{z}_N^*)||p(\mathbf{z}_T)}{2}. \tag{11}$$

We coin this loss as **goal consistency loss**. This loss is based on $D(\mathbf{a}_N)$ in Equation 10 with the only difference being that $p(\mathbf{z}_N^*) = q(\mathbf{z}_N|\mathbf{a}_N^*)$ is computed with respect to the selected best next action representation $\mathbf{a}_N^*$. Finally, we have the cross-entropy loss for comparing the model's prediction $\hat{\mathbf{y}}$ with the ground truth one-hot label $\mathbf{y}$ as $\mathcal{L}_{NA} = -\sum \mathbf{y} \odot \log(\hat{\mathbf{y}})$. The loss function to train the model is a combination of all losses given as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{OG} + \mathcal{L}_{NG} + \mathcal{L}_{GC} + \mathcal{L}_{NA}. \tag{12}$$

We experimented with adding different weights to each loss but there is no significant difference in performance. Therefore, we weigh them equally.

4 Experiments and results

4.1 Datasets, features, and training details

We use well known action anticipation datasets, *Epic-Kitchens55* [5] (EK55) and *Epic-Kitchens100* [6] (EK100) to evaluate our approach.

We validate our models using the TSN features obtained from RGB and optical flow videos, and bag of object features provided by [10] for a fair comparison with existing approaches. Our base model has the following parameters: observed duration - 2 seconds, frame rate - 3 fps, RNN (GRU) hidden dimension $d_h = 256$, abstract goal dimension $d_z = 128$, number of sampled feature-based abstract goals ($Q = 3$), number of next-action-representation candidates ($K = 10$),

Table 1. Comparison of anticipation accuracy with state-of-the-art on EK55 evaluation server with anticipation time of 1 sec. ACT: for action.

Method	**Top-1 accuracy(%)**			**Top-5 accuracy(%)**			**Precision(%)**			**Recall(%)**		
	VERB	NOUN	ACT.	VERB	NOUN	ACT.	VERB	NOUN	ACT	VERB	NOUN	ACT.
					Seen Kitchens (S1)							
RU-LSTM [10]	33.04	22.78	14.39	79.55	50.95	33.73	25.50	24.12	07.37	15.73	19.81	07.66
Lat. Goal [27]	27.96	27.40	08.10	78.09	55.98	26.46	-	-	-	-	-	-
SRL [25]	34.89	22.84	14.24	79.59	52.03	34.61	28.29	25.69	06.45	12.19	19.16	06.34
ImagineRNN [33]	35.44	22.79	14.66	79.72	52.09	34.98	28.04	24.18	06.66	16.03	19.61	07.08
Temp. Agg. [29]	37.87	24.10	16.64	79.74	53.98	36.06	**36.41**	25.20	09.64	15.67	22.01	10.05
MM-Trans [26]	28.59	27.18	10.85	78.64	57.66	30.83	17.50	26.20	03.81	10.81	24.89	04.49
MM-TCN [35]	37.16	23.75	15.45	79.48	51.86	34.37	28.18	23.82	06.94	16.05	22.31	08.40
AVT [13]	34.36	20.16	16.84	80.03	51.57	36.52	23.25	17.77	09.71	14.02	18.81	10.11
DCR [34]	-	-	17.70	-	-	**38.50**	-	-	-	-	-	-
Abstract Goal (VRNN)	**51.56**	**35.34**	**22.03**	**82.56**	**58.01**	38.29	34.83	**31.33**	**13.08**	**26.67**	**31.42**	**12.20**
					Unseen Kitchens (S2)							
RU-LSTM [10]	27.01	15.19	08.16	69.55	34.38	21.10	13.69	09.87	03.64	09.21	11.97	04.83
Lat. Goal [27]	22.40	19.12	04.78	72.07	42.68	16.97	-	-	-	-	-	-
SRL [25]	27.42	15.47	08.88	71.90	36.80	22.06	20.23	12.48	02.84	07.83	12.25	04.33
ImagineRNN [33]	29.33	15.50	09.25	70.67	35.78	22.19	17.10	12.20	03.47	09.66	12.36	05.21
Temp. Agg. [29]	29.50	16.52	10.04	70.13	37.83	23.42	20.43	12.95	04.92	08.03	12.84	06.26
MM-Trans [26]	26.80	18.40	06.76	70.40	**44.18**	20.04	09.53	15.17	02.23	07.73	15.19	03.34
MM-TCN [35]	30.66	14.92	08.91	72.00	36.67	21.68	10.51	12.26	04.35	09.79	12.72	04.94
AVT [13]	30.66	15.64	10.41	72.17	40.76	24.27	12.86	11.83	04.84	09.89	13.46	06.41
DCR [34]	-	-	10.90	-	-	**24.80**	-	-	-	-	-	-
Abstract Goal (VRNN)	**41.41**	**22.36**	**13.28**	**73.10**	41.62	24.24	**23.62**	**18.29**	**08.73**	**15.70**	**18.29**	**08.29**

$\mathcal{L}_{total}$ loss, and fixed anticipation time - 1s (following EK55 and EK100 evaluation server criteria), unless specified otherwise. We use a batch size of 128 videos and train for 15 epochs with a learning rate of 0.001 using Adam optimizer with weight decay (AdamW) in Pytorch. All our MLPs have 256 hidden dimensions.

4.2 Comparison with state-of-the-art

We compare the performance of Abstract Goal (our method) with current state-of-the-art approaches on both the seen and unseen test sets of EK55 datasets in Table 1 using a late fusion of TSN-RGB, TSN-Flow, and Object features like most of the prior work. We train separate models for verb and noun anticipation and combine their predictions to obtain action anticipation accuracy. The model structure is the same for both the verb and noun models but the final classification output is either verb or noun. Our method outperforms all other prior state-of-the-art methods for both seen kitchens (S1) and unseen kitchens (S2). Notably, we outperform Transformer-based AVT [13] and Temporal-Aggregation [29] in all measures in both seen and unseen kitchens except for Top-5 accuracy on unseen kitchens. We believe this improvement is due to two factors, (i) stochastic modeling is massively important for action anticipation, and (ii) the effective use of goal information is paramount for better action anticipation.

Despite, these excellent results on EK55, our overall results on EK100 are not state-of-the-art–see Table 2. Our method performs not as well as recent methods that are extensively pre-trained vision transformer (ViT) models with image and action recognition datasets before being trained for action anticipation [13,12,28]. On the other hand, our model is trained directly on the target

Table 2. Comparison on EK100 dataset on evaluation server using test set. Accuracy measured by mean recall@5 (%) following the standard protocol.

Method	**Input**	**Overall**			**Unseen Kitchens**			**Tail Classes**		
		VERB	NOUN	ACT.	VERB	NOUN	ACT.	VERB	NOUN	ACT.
AVT [13]	Frames	26.69	32.33	16.74	21.03	27.64	12.89	19.28	24.03	13.81
RAFTformer [12]	Frames	30.10	34.10	15.40	-	-	-	-	-	-
InAViT [28]	Frames	**49.14**	**49.97**	**23.75**	**44.36**	**49.28**	**23.49**	**43.17**	**39.91**	**18.11**
RU-LSTM [6]	TSN	25.25	26.69	11.19	19.36	26.87	09.65	17.56	15.97	07.92
Temp. Agg. [29]	TSN	21.76	30.59	12.55	17.86	27.04	10.46	13.59	20.62	08.85
TransAction [15]	TSN	36.15	32.20	13.39	27.60	24.24	10.05	**32.06**	**29.87**	11.88
DCR[34]	TSN	-	-	17.30	-	-	14.10	-	-	**14.30**
Abstract Goal (VRNN)	TSN	31.40	30.10	14.29	31.36	35.56	**17.34**	22.90	16.42	07.70
Abstract Goal (TF)	TSN	**37.63**	**38.70**	14.21	**34.92**	**38.88**	14.25	30.67	29.10	09.11

dataset using temporal segment network (TSN) [31] features. Compared to the best Transformer model [15,34] trained on TSN features, Abstract Goal - VRNN performs better on both overall and unseen kitchens of the EK100 dataset but not as well on tail classes. EK100 dataset is dominated by long-tailed distribution where 228 noun classes out of 300 are in the tail classes. Similarly, 86 verbs out of 97 are in the tail classes. In our model, the next-action-representation is modeled with a Gaussian distribution (Equation 6), and therefore, it is not able to cater to exceptionally long tail class distributions as in EK100. This is a limitation of our method. We do not witness the tail-class issue in EK55 as the performance measure used is accuracy compared to mean-recall in EK100. Accuracy is influenced heavily by frequent classes but mean-recall treats all classes equally.

For completeness, we test whether the tail class issue on EK100 can be resolved using a transformer network (6 layers with 8 attention heads) instead of a GRU for observed feature summarization. While abstract goal with transformer (TF) improves tail class performance it is not able to outperform [15,34] on tail classes. This confirms our hypothesis that using Gaussian distribution for next-action-representation (action-based abstract goal) can limit tail class performance but improves overall and unseen kitchens anticipation accuracy.

4.3 Impact of goal consistency criterion and loss

In this section, we evaluate the impact of Goal Consistency (GC) criterion and the loss derived from it $\mathcal{L}_{gc}$ using the validation set of EK55 and EK100 datasets. We train separate models for verb and noun anticipation using TSN-RGB (RGB) and Object (OBJ) features, respectively. As Mean and Median sampling are used in prior variational prediction models [1], here we use mean and median sampling as two baselines to show the effect of GC. We sample $Q \times K$ number of next-action representations ($\mathbf{a_N}$) instead of selecting the best next-action candidate using GC (Algorithm 1). Then we obtain the mean/median vector of all sampled candidates and then make the prediction using the classifier (e.g. mean vector= $\frac{\sum \mathbf{a_N}}{Q \times K}$). We also experimented with a majority/median class prediction baseline.

In this case, we take all $Q \times K$ predictions from the classifier (from the next action-representation candidates) and pick the majority/median class as the final prediction. Everything else stays the same for all these mean/majority/median baseline models, except we do not use the GC criterion (Equation 10) and the goal consistency loss $\mathcal{L}_{gc}$. Results are reported in Table 3.

Table 3. The impact of goal consistency criterion and loss. @1 and @5 denotes Top-1 and Top-5 accuracy and V stands for verb and N stands for noun.

Goal candidate (Q) &		EK55				EK100			
Action candidate (K)		V@1	V@5	N@1	N@5	V@1	V@5	N@1	N@5
Mean	Q=1, K=10	41.79	72.23	25.79	49.50	44.51	76.89	22.72	50.78
Median		41.16	71.32	24.30	48.31	45.44	77.91	22.15	51.23
Majority class		41.98	72.89	25.98	50.01	42.98	74.56	24.13	53.45
Median class		41.02	72.11	22.88	49.87	44.19	77.00	22.97	51.98
Our model		**45.18**	**77.30**	**28.16**	**51.08**	**48.84**	**80.52**	**27.50**	**55.83**
Mean	Q=3, K=10	39.40	72.23	24.22	48.96	45.90	77.88	22.41	50.87
Median		41.32	71.32	26.60	51.70	45.63	77.02	24.33	52.87
Majority class		38.39	69.42	24.70	48.22	45.72	78.61	22.61	50.89
Median class		40.43	71.43	26.52	52.33	45.84	78.09	23.78	52.33
Our model		**44.68**	**77.14**	**28.29**	**53.78**	**49.02**	**80.86**	**28.52**	**54.91**
Without $\mathcal{L}_{GC}$	Q=1, K=1	38.31	70.77	19.74	43.11	43.82	77.45	21.25	51.99
With $\mathcal{L}_{GC}$		**40.88**	**71.43**	**22.09**	**46.29**	**46.80**	**78.41**	**26.80**	**53.32**

As can be seen from the results, there is a significant impact of GC. Especially, there is an improvement of 3.39% and 2.37% for top-1 verb and noun accuracy respectively using our GC model in the EK55 dataset for $Q = 1, K = 10$ over Mean sampling baseline. A similar trend can be seen for EK100 and $Q = 3, K = 10$ as well. Our model also outperforms majority and median class sampling baselines for both $[Q = 1, K = 10]$ and $[Q = 3, K = 10]$ configurations indicating the effectiveness of goal consistency both as GC criterion and GC loss $\mathcal{L}_{GC}$. Overall, our method with GC loss and criterion performs better than all other variants. Perhaps this is because the GC criterion allows the model to regularize the candidate selection while GC loss allows the model to enforce this during the training. This clearly shows the impact of *goal consistency formulation* of our model for action anticipation.

We perform a more controlled experiment to further evaluate the impact of GC loss where we set $Q = 1$ and $K = 1$ and train our model with and without GC loss ($\mathcal{L}_{GC}$). It should be noted that when $Q = 1$ and $K = 1$, GC criterion has no impact because we do not have multiple candidates to evaluate. The only meaningful way to see the effect of GC is to compare a model trained with and without the GC loss. To obtain a statistically meaningful result, we repeat this experiment 10 times and report the mean performance. As it can be seen from the results in Table 3 (last two rows), clearly GC loss has a positive impact even when we just sample a single action candidate from our stochastic model. We see that compared to our model variant $[Q = 1, K = 1$ with $\mathcal{L}_{GC}]$, the $[Q = 1, K = 10$ with $\mathcal{L}_{GC}]$ model performs significantly better (last row vs row 5 of Table 3). This indicates the impact of next-action-representation sampling (Equation 6) even for a single sampled feature-based abstract goal ($Q = 1$).

Table 4. Ablation on the sensitivity of number of sampled feature-based-abstract-goals (Q) and next-action representation candidate K on EK55 and EK100 validation set.

parameter	value	EK55				EK100			
		V@1	V@5	N@1	N@5	V@1	V@5	N@1	N@5
num. feature-based abstract goals (Q) (K = 10)	1	45.18	77.30	28.16	51.08	48.84	80.52	27.50	**55.83**
	2	44.44	76.19	**28.47**	52.38	49.25	80.44	28.41	55.65
	3	44.68	77.14	28.29	**53.78**	49.02	**80.86**	**28.52**	54.91
	4	45.31	**77.91**	26.28	50.33	48.86	80.46	28.16	55.11
	5	**45.80**	77.40	26.95	51.93	**49.71**	80.40	28.04	55.16
num. next action candidates (K) (Q=3)	1	39.81	72.31	21.48	44.96	44.24	75.67	20.06	42.56
	3	40.49	74.20	22.60	46.22	44.37	76.11	21.07	44.51
	5	41.32	74.26	23.17	48.23	45.61	78.91	22.91	45.12
	10	**44.68**	77.14	**28.29**	**53.78**	49.02	80.86	**28.52**	**54.91**
	20	43.79	**79.00**	27.07	51.10	49.01	80.36	28.13	55.40
	30	44.56	77.81	27.80	51.00	**49.18**	**81.20**	27.44	53.42

We conclude that the goal consistency loss, the goal consistency criterion, and next-action-representation distribution modeling (all novel concepts introduced in this paper) are effective for action anticipation.

Table 5. Loss ablation on EK55 and EK100 validation set. *i.e.*$\mathcal{L}_{NA}$-Next action cross-entropy loss, $\mathcal{L}_{OG}$-Feature-based abstract goal loss, $\mathcal{L}_{NG}$-Action-based abstract goal loss, $\mathcal{L}_{GC}$-Goal consistency loss.

Losses	EK55				EK100			
	V@1	V@5	N@1	N@5	V@1	V@5	N@1	N@5
$\mathcal{L}_{NA}$	21.36	69.69	27.76	51.89	24.46	72.31	27.12	54.55
$\mathcal{L}_{NA} + \mathcal{L}_{OG}$	44.42	77.79	28.41	51.31	43.23	75.63	23.45	52.89
$\mathcal{L}_{NA} + \mathcal{L}_{NG}$	46.01	77.94	29.05	52.32	46.94	78.44	22.96	49.66
$\mathcal{L}_{NA} + \mathcal{L}_{GC}$	43.83	77.43	28.06	51.87	44.45	76.72	20.31	47.87
$\mathcal{L}_{NA} + \mathcal{L}_{OG} + \mathcal{L}_{NG}$	44.47	77.12	28.51	51.34	46.73	78.62	24.56	51.33
$\mathcal{L}_{NA} + \mathcal{L}_{OG} + \mathcal{L}_{GC}$	45.47	77.42	28.61	52.34	47.25	78.11	26.91	53.34
$\mathcal{L}_{NA} + \mathcal{L}_{OG} + \mathcal{L}_{NG} + \mathcal{L}_{GC}$	**46.37**	**77.97**	**29.86**	**52.74**	**49.02**	**80.86**	**28.52**	**54.91**

Apart from GC loss, we also study the impact of other loss functions described in Section 3.5 and report the results in Table 5. If we use only the supervised cross-entropy loss (i.e., $\mathcal{L}_{NA}$), then the performance is the worst, especially for verbs. Both $\mathcal{L}_{OG}$ and $\mathcal{L}_{NG}$ help in regularizing the abstract goal representations ($\mathbf{z_t}$ and $\mathbf{a_N}$), and therefore results improve significantly. Especially, the $\mathcal{L}_{NA} + \mathcal{L}_{NG}$ is the best loss combination for a pair of losses. When we combine all four losses, we get the best results. While $\mathcal{L}_{NA} + \mathcal{L}_{NG}$ regularizes the learning of abstract goal representations, $\mathcal{L}_{GC}$ which minimizes the divergence between feature-based and action-based goal distributions improves the choice of next verb or noun among the plausible candidates. We conclude that all four losses are important for our model.

4.4 Effect of action-based abstract goal distributions

We demonstrate the efficacy of action-based abstract goal in our model by comparing it to a variant of our model having only the feature-based abstract goal (equivalent to a VRNN) in Table 6. For the feature-based abstract goal (Feat. abs. goal), we obtain a latent variable $\mathbf{z_T}$ and the observed action representation $\mathbf{a_O}$ from Equation 5. We classify $\mathbf{a_O}$ using a classifier to obtain the future action and train using cross-entropy loss and KL-divergence ($\mathcal{L}_{OG}$). We do not have GC criterion when using only the feature abstract goal distribution and hence we sample 30 candidates for $\mathbf{a_O}$ and consider their mean or median. The number of sampled candidates is chosen to match our feature + action abstract goal model with 30 next action candidates ($Q = 3, K = 10$). As shown in Table 6, using action-based abstract goal in conjunction with feature-based abstract goal performs much better than only feature abstract goal distribution (under both mean or median prediction).

Table 6. Effect of action-based abstract goal

Model	V@1	V@5	N@1	N@5
Abs. Goal (Feat)–mean	27.76	61.23	22.34	46.78
Abs. Goal (Feat)–median	38.13	68.94	23.85	47.56
Abs. Goal (Feat+Act)–mean	39.40	72.23	24.22	48.96
Abs. goal (Feat+Act)–median	**44.68**	**77.14**	**28.29**	**53.78**

5 Conclusion

We present a novel approach for action anticipation where abstract goals are learned with a stochastic recurrent model. We outperform existing approaches on EK55 and our model generalizes to unseen kitchen environments in both EK55 and EK100 datasets. We also show the importance of goal consistency criterion, goal consistency loss, next-action representation modeling, and architecture. One limitation of the current work is the inability to directly interpret the latent goal representation learned by our model. Second, our method is not able to tackle long-tail-class distribution issues. In the future, we aim to address these limitations of our model.

Acknowledgment This research/project is supported by the National Research Foundation, Singapore, under its NRF Fellowship (Award# NRF-NRFF14-2022-0001) and by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-016). This research is also supported by funding allocation to B.F. by the Agency for Science, Technology and Research (A*STAR) under its SERC Central Research Fund (CRF), as well as its Centre for Frontier AI Research (CFAR).

References

1. Abu Farha, Y., Gall, J.: Uncertainty-aware anticipation of activities. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)

2. Abu Farha, Y., Richard, A., Gall, J.: When will you do what?-anticipating temporal occurrences of activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5343–5352 (2018)
3. Chang, C.Y., Huang, D.A., Xu, D., Adeli, E., Fei-Fei, L., Niebles, J.C.: Procedure planning in instructional videos. In: European Conference on Computer Vision. pp. 334–350. Springer (2020)
4. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. Advances in neural information processing systems **28**, 2980–2988 (2015)
5. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: The epic-kitchens dataset: Collection, challenges and baselines. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
6. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. International Journal of Computer Vision pp. 1–23 (2021)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
8. Fernando, B., Herath, S.: Anticipating human actions by correlating past with the future with jaccard similarity measures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13224–13233 (2021)
9. Fraccaro, M., Sønderby, S.K., Paquet, U., Winther, O.: Sequential neural models with stochastic layers. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 2207–2215 (2016)
10. Furnari, A., Farinella, G.: Rolling-unrolling lstms for action anticipation from first-person video. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
11. Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Forecasting future action sequences with neural memory networks. In: 30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019. p. 298. BMVA Press (2019), `https://bmvc2019.org/wp-content/uploads/papers/0585-paper.pdf`
12. Girase, H., Agarwal, N., Choi, C., Mangalam, K.: Latency matters: Real-time action forecasting transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18759–18769 (2023)
13. Girdhar, R., Grauman, K.: Anticipative Video Transformer. In: ICCV (2021)
14. Gong, D., Lee, J., Kim, M., Ha, S.J., Cho, M.: Future transformer for long-term action anticipation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3052–3061 (2022)
15. Gu, X., Qiu, J., Guo, Y., Lo, B., Yang, G.: Transaction: ICL-SJTU submission to epic-kitchens action anticipation challenge 2021. CoRR **abs/2107.13259** (2021)
16. 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. pp. 9925–9934 (2019)
17. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
18. Liu, M., Tang, S., Li, Y., Rehg, J.M.: Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In: European Conference on Computer Vision. pp. 704–721. Springer (2020)

19. Liu, T., Lam, K.M.: A hybrid egocentric activity anticipation framework via memory-augmented recurrent and one-shot representation forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13904–13913 (2022)
20. Loh, S.B., Roy, D., Fernando, B.: Long-term action forecasting using multi-headed attention-based variational recurrent neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2419–2427 (2022)
21. Mascaró, E.V., Ahn, H., Lee, D.: Intention-conditioned long-term human egocentric action anticipation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6048–6057 (2023)
22. 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. pp. 3165–3174 (2019)
23. Miech, A., Laptev, I., Sivic, J., Wang, H., Torresani, L., Tran, D.: Leveraging the present to anticipate the future in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
24. Nawhal, M., Jyothi, A.A., Mori, G.: Rethinking learning approaches for long-term action anticipation. In: European Conference on Computer Vision. pp. 558–576. Springer (2022)
25. 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)
26. Roy, D., Fernando, B.: Action anticipation using pairwise human-object interactions and transformers. IEEE Transactions on Image Processing (2021)
27. Roy, D., Fernando, B.: Action anticipation using latent goal learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 2745–2753 (January 2022)
28. Roy, D., Rajendiran, R., Fernando, B.: Interaction region visual transformer for egocentric action anticipation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6740–6750 (2024)
29. Sener, F., Singhania, D., Yao, A.: Temporal aggregate representations for long-range video understanding. In: European Conference on Computer Vision. pp. 154–171. Springer (2020)
30. Song, Y., Byrne, E., Nagarajan, T., Wang, H., Martin, M., Torresani, L.: Ego4d goal-step: Toward hierarchical understanding of procedural activities. Advances in Neural Information Processing Systems **36** (2024)
31. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: European conference on computer vision. pp. 20–36. Springer (2016)
32. Wu, C.Y., Li, Y., Mangalam, K., Fan, H., Xiong, B., Malik, J., Feichtenhofer, C.: Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. arXiv preprint arXiv:2201.08383 (2022)
33. Wu, Y., Zhu, L., Wang, X., Yang, Y., Wu, F.: Learning to anticipate egocentric actions by imagination. IEEE Transactions on Image Processing **30**, 1143–1152 (2021). https://doi.org/10.1109/TIP.2020.3040521
34. Xu, X., Li, Y.L., Lu, C.: Learning to anticipate future with dynamic context removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12734–12744 (2022)

35. 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. pp. 2249–2258 (2021)
36. Zhang, T., Min, W., Yang, J., Liu, T., Jiang, S., Rui, Y.: What if we could not see? counterfactual analysis for egocentric action anticipation. In: IJCAI (2021)
37. Zhao, Q., Wang, S., Zhang, C., Fu, C., Do, M.Q., Agarwal, N., Lee, K., Sun, C.: Antgpt: Can large language models help long-term action anticipation from videos? In: The Twelfth International Conference on Learning Representations (2023)