Action anticipation using latent goal learning

Debaditya Roy and Basura Fernando

Social and Cognitive Computing, IHPC, A*STAR, Singapore.

Abstract

To get something done, humans perform a sequence of actions dictated by a goal. So, predicting the next action in the sequence becomes easier once we know the goal that guides the entire activity. We present an action anticipation model that uses goal information in an effective manner. Specifically, we use a latent goal representation as a proxy for the "real goal" of the sequence and use this goal information when predicting the next action. We design a model to compute the latent goal representation from the observed video and use it to predict the next action. We also exploit two properties of goals to propose new losses for training the model. First, the effect of the next action should be closer to the latent goal than the observed action, termed as "goal closeness". Second, the latent goal should remain consistent before and after the execution of the next action which we coined as "goal consistency". Using this technique, we obtain state-of-the-art action anticipation performance on scripted datasets 50Salads and Breakfast that have predefined goals in all their videos. We also evaluate the latent goal-based model on EPIC-KITCHENS55 which is an unscripted dataset with multiple goals being pursued simultaneously. Even though this is not an ideal setup for using latent goals, our model is able to predict the next noun better than existing approaches on both seen and unseen kitchens in the test set.[1]

1. Introduction

Humans perform complex activities like cooking by executing actions that follow a rational order. For example, making a salad may involve the following sequence of actions - *washing vegetables* $\succ$ *cutting vegetables* $\succ$ *seasoning* $\succ$ *mixing*. Humans can recognize the intent of another human after observing a few actions being performed. Moreover, humans have a belief over plausible goals based on the observed actions. The most likely next action is the one that helps achieve the most plausible goal. In this paper, we propose a model that uses goal information to anticipate the next action. It has been shown that recognizing the goal of a human can help robots better anticipate human actions [14]. Particularly in cooking activities, goal or overall activity recognition is beneficial in anticipating one or more future actions [23]. However, some cooking activities like *making pancake*, *making fried eggs*, and *making scrambled eggs* start with a similar sequence of actions. In such cases, it is difficult to identify the intent or goal explicitly from the first few actions. Hence, we propose an approach that uses an abstract representation of the goal termed as *latent goal* to anticipate the next action.

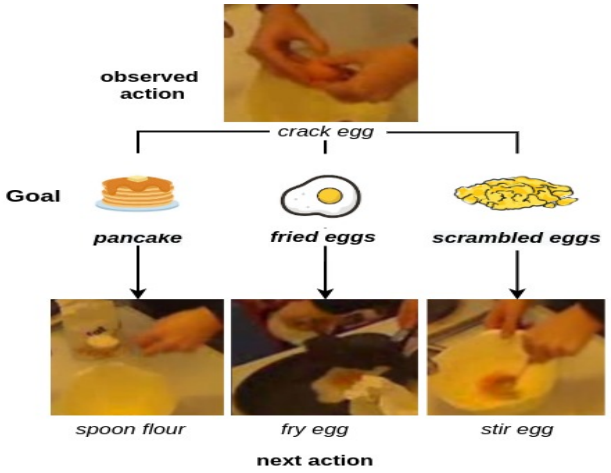

Figure 1. Next action anticipation depends on the underlying goal. Examples from the Breakfast dataset.

The concept of a latent goal has been used in pedestrian intent detection [20, 21] and human trajectory prediction [31]. For pedestrians, the latent goal is a *single intent* [31] to reach a destination. On the other hand, complex activities like cooking have *sequential intent* [31] where a person interacts with various objects in the kitchen sequentially. The sequential intent or latent goal becomes more apparent as more actions are performed in the sequence but it always guides the sequence of actions [15] as shown in Figure 1.

[1]Code:https://github.com/debadityaroy/LatentGoal

Hence, in this work, we use a representation of the latent goal derived from the observed visual representation. We then use this latent goal representation and the observed visual representation to predict the next action and the next visual representation.

A property of the latent goal is that every subsequent action in the sequence should bring us closer to the goal. In the trajectory prediction [27, 6], closeness to a goal is defined as predicted destination of a trajectory to the actual destination. In procedure planning, goal closeness is defined based on the final action, which is always known beforehand [3]. However, there are tasks where the final action may not be the goal of the task. For example, *pouring milk* is the final action for both *making coffee* and *making cereal* but does not capture the intent of the activities. As the input to "next action anticipation" model is the observed visual representation, measuring the activity progress based on "goal closeness" is not trivial. Hence, we propose *goal closeness loss* which encourages the anticipated action to produce a visual representation that is closer to the latent goal compared observed visual representation. Furthermore, the latent goal representation should remain consistent throughout for consecutive actions. The current latent goal should not differ a lot from the latent goal computed after anticipating the next action. We propose *goal consistency loss* to minimize the difference in the latent goal representation before and after action anticipation.

We evaluate our method on two scripted datasets (50Salads and Breakfast) and we show that latent goal improves action anticipation performance. We also evaluate the impact of latent goals for anticipating unscripted actions (nouns and verbs) using the EPIC-KITCHENS55 dataset. With the use of latent goal we obtain better noun anticipation accuracy than existing approaches. In summary, our contributions are as follows -

- We propose a novel latent goal-based action anticipation framework.
- We propose two new losses - a) *goal closeness loss* that ensures progress towards the latent goal, and b) *goal consistency loss* that ensures the latent goal representation remains consistent for consecutive actions.

2. Related Work

Goals have been used in literature for predicting one or more future actions. In [23], authors use the complex activity label for the entire sequence of actions along with the observed action to predict the next action. In [3], the goal of the entire sequence is the final visual representation at the end of a sequence of actions. Each action in the sequence is predicted based on its closeness to the goal. These approaches are feasible for instructional videos [3] where the final action/visual representation is the underlying goal or the underlying complex activity is known for the sequence [23]. However, in case of regular activities (EPIC-KITCHENS55 [4]), the overall complex activity guiding the sequence may not be known or the final action may not represent the overall goal. So, we devise a latent goal that serves as a representation for the underlying intent or the overall activity or the final action depending on the activity. Using a latent goal means that our approach does not require the overall activity label [23] or the final action and its visual representation [3] for action anticipation.

Action anticipation is defined as the task of generating the visual representation of future frames by leveraging the temporal structure of videos in [28]. From a single input frame, multiple possible future frames are generated using a regression-based CNN network and subsequently classified to predict the action label. In [17], the action label of a frame 1 second into the future is predicted using a low-rank linear model called the transitional model. In [1], a sequence of future actions is predicted instead of a single future action using an RNN and a CNN based on the observed action labels as input. In [12], time-conditioned skip connections are used in addition to attended temporal features for anticipating future actions. In [8], a new Jaccard vector similarity is used to correlate past features with the future features for action anticipation. Effective use of human-object feature interaction model for action anticipation is presented in [22]. In our work, we leverage the visual features directly to predict the next action.

Along with the action labels for each frame, spatial representation of a frame was added to forecast future action labels in [10]. Authors propose a neural memory network that stores information in an LSTM cell by comparing the similarity of the input with the existing memory content for both the labels and spatial streams. Another approach that considers three frame-based representations - spatial, motion, and object, to predict future actions was proposed in [9]. Using an unrolling LSTM, the authors showed that multiple time-steps in the future could be predicted. A multi-modal attention network is used to decide the best possible combination of the spatial, motion and object representations. Apart from LSTMs, other temporal networks that have also been used extensively used for action anticipation include Recurrent Neural Networks (RNN) [24, 2, 26, 18, 21] and Conditional Random Fields (CRF) [13, 30]. Instead of using RNNs or CRF for summarizing temporal information, a simpler approach for aggregating both recent and long-term temporal history using non-local blocks [29] is presented in [23]. The authors show that while long-term aggregation plays a part in anticipation, recent actions are more informative in determining the immediate future. We also show that the recent past is more effective because the previous action is more informative than the long-term past for anticipating

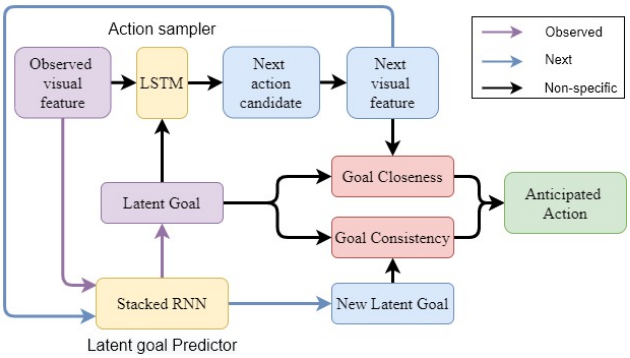

Figure 2. Observed visual features of the current action are used to generate a latent goal using a stacked RNN. A number of sample action candidates are generated for the next action using the latent goal and observed features. Each candidate produces a latent goal and a visual feature and we compare them using - *goal closeness* and *goal consistency*, respectively. The best candidate to fit these criteria is the anticipated action. Best viewed in color.

the next action.

3. Action Anticipation with Latent Goal

For next action anticipation, we observe a video for a few seconds and predict the action which occurs 1 second after the observation period. The action to be predicted is the next action in the sequence of actions which is not part of the observed video. The goal of the action sequence may be completed long after the anticipated action but it often determines which action follows the action in observed video as previously shown in Figure 1. So, in our proposed framework (Figure 2), we use the latent goal representation along with the visual features from the observed video to generate multiple candidates for the next action and next visual feature. The next action is chosen from the candidates based on two criteria - *goal closeness* and *goal consistency*. Goal closeness helps choose the action candidate whose corresponding next visual feature is closest to the latent goal. Goal consistency ensures that the new latent goal computed using the next visual feature is closest to the latent goal computed using the observed visual feature.

Algorithm 1 lists the entire procedure of action anticipation with latent goal. At first, we need a single visual feature from the observed video. The duration of actions differ across actors and datasets which means that we need to vary the observation period to cover the entire action. So, we employ temporal aggregation to obtain a single visual representation for the entire observed video. Temporal aggregation can be through mean-pooling, max-pooling or an RNN. We empirically found that max-pooling produces the best aggregate representation. The aggregated visual representation is a d-dimensional feature denoted by $\mathbf{x}^o$.

Algorithm 1 Action Anticipation with Latent Goal

Input: Features for observed video $\{\mathbf{x}_1^o, \mathbf{x}_2^o, \cdots, \mathbf{x}_n^o\}$, aggregation function $f(\cdot)$, inverse mapping $h(\cdot)$, consistency threshold δ, next feature predictor $k(\cdot)$, separation threshold ϵ

$\mathbf{x}^o = f(\{\mathbf{x}_1^o, \mathbf{x}_2^o, \cdots, \mathbf{x}_n^o\})$ // observed visual feature
$\bar{a}^o = \varnothing$ // initial action representation
$\mathbf{x}^* = \mathbf{x}^o, \bar{a}^* = \bar{a}^o$ // best visual and action candidate
$\mathbf{x}_g^o = StackedRNN(\mathbf{x}^o, \bar{a}^o)$ // initial latent goal
if $||\mathbf{x}^* - \mathbf{x}_g^o||_2^2 > \epsilon$ **then**
  $\{\bar{a}'\} \sim RNN(\mathbf{x}^o, \bar{a}^o)$ //Sample actions
  for $\bar{a}_i \in \bar{a}'$ **do**
    $\mathbf{x} = \phi_{\mathbf{x}}(\mathbf{x}^o, \bar{a}_i, \mathbf{x}_g^o)$ // next visual representation
    $\mathbf{x}_g = StackedRNN(\mathbf{x}, \bar{a}_i)$
    if $||\mathbf{x}^* - \mathbf{x}_g^o||_2^2 - ||\mathbf{x} - \mathbf{x}_g^o||_2^2 > 0$ and $||\mathbf{x}_g^o - \mathbf{x}_g||_2^2 < \delta$ **then**
      $\mathbf{x}^* = \mathbf{x}$ // best next visual representation
      $\bar{a}^* = \bar{a}_i$ // best next action representation
      $\mathbf{x}_g^* = \mathbf{x}_g$ // best next latent goal
    end if
  end for
end if
$\hat{a} = \phi_a(\bar{a}^*)$ //anticipated action
return $\hat{a}, \mathbf{x}^*, \mathbf{x}_g^*, \mathbf{x}^o, \mathbf{x}_g^o$

3.1. Latent goal computation

In our paper, we define goal of the action sequence as the visual representation after performing the final action based on the procedure planning paradigm in [3]. Unlike procedure planning, the final visual representation is not available during next action anticipation. So, we use the observed visual feature to generate the *latent goal* representation as a surrogate for the goal. There are two challenges in generating the latent goal representation. First, the observed video may contain the first or second action in the sequence which means the goal is far away in the future. Second, we do not have access to the intermediate actions during anticipation and need proxy representations for each intermediate action. Our solution is to use stacked LSTMs that have been shown to add levels of abstraction from input observations over time by operating at different timescales [19]. We use a stacked LSTM with the number of layers equal to the average number of actions per video (varies by dataset). Each layer in the stacked LSTM represents an intermediate action that leads to the latent goal. The latent goal based on the observed visual feature is given by

$$\mathbf{x}_g^o = StackedRNN(\mathbf{x}^o, \bar{a}^o), \tag{1}$$

where $\bar{a}^o$ is the initial action representation initialized with zeros instead of ground truth action labels to mimic the testing scenario where action labels are not available.

It is also important to check that the observed visual representation is not very close to the latent goal estimate. The next visual representation (effect of anticipated action) should be closer to the latent goal than the observed visual representation. So, we enforce a minimum separation threshold (ϵ) between the latent goal estimate and observed visual representation. We determine the minimum separation threshold ϵ empirically.

3.2. Next Action Candidate Selection

The next action depends on the observed visual feature and the latent goal. There can be multiple plausible next actions and we employ an LSTM to sample all the possible candidates. Each candidate's viability as the next action depends on whether it takes us closest to the latent goal. So, we derive a visual representation that shows the effect of performing each candidate action called *next visual feature*. The next visual feature is computed using the action candidate ($\bar{a}_i$), the observed visual representation, and the sequence's goal

$$\mathbf{x} = \phi_{\mathbf{x}}([\mathbf{x}^o, \bar{a}_i, \mathbf{x}_g^o]), \tag{2}$$

where $\phi_{\mathbf{x}}$ is a linear layer, $[\cdot,\cdot,\cdot]$ represents concatenation. Once the next visual feature is obtained, we compute its closeness to the latent goal to determine whether we have moved closer to the goal compared to the observed visual feature. We refer to this criteria as *candidate closeness* and use it to determine the best action candidate ($\mathbf{x}^*$) as follows

$$||\mathbf{x}^* - \mathbf{x}_g^o||_2^2 > ||\mathbf{x} - \mathbf{x}_g^o||_2^2. \tag{3}$$

Another criteria for ensuring that we have chosen the best action candidate is to match the latent goal generated and compare it to the initial latent goal. We term this criterion as *candidate goal consistency* which measures the distance between latent goal representations for observed and anticipated action given as

$$||\mathbf{x}_g^o - \mathbf{x}_g||_2^2 < \delta \tag{4}$$

where δ is the closeness threshold and $\mathbf{x}_g$ is the latent goal generated using the action candidate and its corresponding visual feature as per Equation1. The action candidate that best satisfies both candidate goal closeness and candidate goal consistency is chosen as the best action candidate.

The choice of the best action candidate is designed to be greedy as we want to predict only the next action in the sequence. Once the best action candidate is obtained, we use an inverse mapping function to predict the actual action class using a linear transformation (ϕ_a). The entire process of next action anticipation is described in Algorithm 1.

3.3. Losses

The primary objective of the proposed architecture is to predict the next action. So, we compute the cross entropy loss for the next action $\hat{a}$ with respect to the ground truth action a given by

$$\mathcal{L}_{ant} = -\sum_{c=1}^{C} a_c log(\hat{a}_c), \tag{5}$$

where C is the total number of action classes. Our network also depends on the observed visual feature and its ability to encode the information in the observed video. To ensure that the observed visual feature represents the ongoing action accurately we compute the following cross entropy loss

$$\mathcal{L}_{obs} = -\sum_{c=1}^{C} a_c^o log(\hat{a}_c^o), \tag{6}$$

where $\hat{a}^o$ is the prediction over the current action obtained with a linear transformation on the observed visual feature.

The cross entropy loss focus on next action and current action predictions. We also need to measure the quality of the latent goal representation by the model. Hence, we propose *goal consistency loss* which ensures that the latent goal representation is consistent during training. Goal consistency loss is realized as a max-margin loss given as

$$\mathcal{L}_{cons} = max(0, ||\mathbf{x}_g^* - \mathbf{x}_g^o||_2^2 - \delta + m) \tag{7}$$

which measures the deviation between the consistency threshold δ and latent goal difference $||\mathbf{x}_g^* - \mathbf{x}_g^o||$. The margin m is a very small value to ensure numerical stability.

We also enforce the property of closeness to the latent goal to improve the anticipation performance of the model. Similar to goal consistency, goal closeness is represented as a max-margin loss as follows

$$\mathcal{L}_{close} = max(0, ||\mathbf{x}^* - \mathbf{x}_g^o||_2^2 - ||\mathbf{x}^o - \mathbf{x}_g^o||_2^2 + m). \tag{8}$$

We measure the distance to the latent goal $\mathbf{x}_g^o$ from the observed visual $\mathbf{x}^o$ and the next visual (effect of anticipated action) $\mathbf{x}^*$. The max-margin criteria enforces the next visual to be closer to the latent goal than the observed visual. Both goal consistency loss (Equation 7) and goal closeness (Equation 7) may appear similar to candidate closeness (Equation 3) and candidate consistency (Equation 4), respectively. While candidate closeness and consistency are used to obtain the best action candidate, the losses are applied on the obtained action candidate to ensure that the model is trained to follow these properties.

4. Experiments and Results

4.1. Rationale for choosing datasets

We choose 3 datasets for our experiments - 50Salads [25], Breakfast [15], and EPIC-KITCHENS55 [4]. The

goal of 50Salads dataset is for every person to make two different salads following recipes. For the Breakfast dataset, the goal of every video is to prepare one of the 10 breakfast items *coffee*, *orange juice*, *chocolate milk*, *tea*, *cereals*, *fried eggs*, *pancakes*, *fruit salad*, *sandwich*, and *scrambled eggs*. Though both 50Salads and Breakfast datasets are scripted, all the videos in 50Salads have only one underlying goal with a lot more actions per video compared to Breakfast. The EPIC-KTICHENS55 dataset covers unscripted activities mostly involving cooking, food preparation, washing, and others. All the actions are unscripted and depict real-life scenarios involving multi-tasking, searching for an item, thinking what to do next, changing one's mind, or even unexpected surprises. Regular activities present the most challenge as multiple goals may be pursued simultaneously, and the actions for different goals may be intertwined.

4.2. Datasets and Features

50 Salads[25] dataset consists of 50 videos of 25 actors making salads based on recipes provided beforehand. The videos are recorded with a resolution of 640×480 at 30 frames per second. The actors perform 17 different fine-grained actions, and the gaps between these actions are annotated using a background class. The average video length is 6.4 minutes, and there are 20 action instances per video. The published dataset provides five splits, and all the results presented here are averaged over the five splits.

Breakfast[15] dataset consists of 77 hours of procedural videos or 4.1 million frames of 52 actors making breakfast that yields 48 fine-grained action classes. The videos are recorded with a resolution of 320×240 at 15 frames per second. The videos' average duration is comparably shorter at 2.3 minutes with an average of 6 action instances. All the results presented here are averaged over the four splits provided by the authors of the dataset [15].

EPIC-KITCHENS55[4] contains a total of 55 hours of unscripted videos comprising of 39,596 action annotations, 125 verbs, 351 nouns, and 2,513 actions. All the videos are recorded at 60 frames per second with a resolution of 1920×1080. The training set is divided into 232 videos for training (23,493 segments), and 40 videos for validation (4,979 segments) based on the splits provided by [9]. We perform ablation studies on the validation set to obtain the best network architecture. The results of our proposed approach are then compared on the seen and unseen kitchens of the test set using the evaluation server.

We use I3D features provided by [7] for both Breakfast and 50 Salads datasets. I3D features have been shown better for action anticipation than similar representations like R(2+1)D [23]. In addition to I3D, we also demonstrate our results on Fisher vector representation of dense trajectory features provided by [16] for the 50 Salads dataset and by [15] for the Breakfast dataset. For the EPIC-KTICHENS55 dataset, we perform our experiments using Temporal Segment Network (TSN) and bag-of-object features as most approaches for action anticipation [9, 23] on EPIC-KITCHENS55 have used the same.

Method	**Anticipation Accuracy**
Deep Regression [28]	8.1
RNN [1]	30.1
CNN [1]	29.8
Temporal Agg. [23]	40.7
Latent goal	**59.6**

Table 1. Comparison with state-of-the-art on 50 Salads

Method	**Anticipation Accuracy**
Deep Regression [28]	6.2
RNN [1]	30.1
CNN [1]	27.0
Predictive+ Transitional [17]	32.3
Temporal Agg. [23]	47.0
Latent goal (I3D)	**47.2**

Table 2. Comparison with state-of-the-art on Breakfast

Method	**Top-1 Anticipation Accuracy**			**Top-5 Anticipation Accuracy**		
	VERB	**NOUN**	**ACT.**	**VERB**	**NOUN**	**ACT.**
Seen Kitchens (S1)						
RU-LSTM [9]	33.04	22.78	14.39	79.55	50.95	33.73
Temp. Agg. [23]	**37.87**	24.10	16.64	79.74	53.98	36.06
Vid. Trans. [11]	34.36	20.16	**16.84**	**80.03**	51.57	**36.52**
Latent goal TSN-RGB (Verb) + Obj. (Noun)	27.96	**27.40**	8.10	78.09	**55.98**	26.46
Unseen Kitchens (S2)						
RU-LSTM [9]	27.01	15.19	08.16	69.55	34.38	21.10
Temp. Agg. [23]	29.50	16.52	10.04	70.13	37.83	23.42
Vid. Trans. [11]	**30.66**	15.64	**10.41**	**72.17**	40.76	**24.27**
Latent goal TSN-RGB (Verb) + Obj. (Noun)	22.40	**19.12**	4.78	72.07	**42.68**	16.97

Table 3. Comparison with state-of-the-art on EPIC-KITCHENS55 test sets

All the LSTMs used for modeling the RNNs in Algorithm 1 have 2048 hidden dimensions for Breakfast, 128 for 50Salads, and 1024 for EPIC-KITCHENS55 unless stated otherwise. The $StackedRNN$ in Algorithm 1 for predicting the latent goal representation consists of 6 layers for Breakfast, and 10 layers for 50Salads based on the average number of actions in the video. As the number of actions per video is quite large for EPIC-KITCHENS55, we choose 10 layers for the $StackedRNN$ LSTM in our experiments. The choice of the number of $StackedRNN$ layers is empirically validated (details in Supplementary).

4.3. Comparison with state-of-the-art

In Table 1 and 2, we compare the results of our latent goal-based anticipation approach to state-of-the-art results on 50Salads and Breakfast datasets, respectively. On the 50 Salads dataset, we achieve using latent goal achieves a vast improvement of 18.9% over the state-of-the-art method [23]

using an observation period of 3 seconds. In case of Breakfast, we see an improvement using the same I3D features as temporal aggregation [23]. We see a significant improvement in 50Salads compared to Breakfast because the different goals in Breakfast are harder to infer distinctly from the video itself. The Breakfast dataset is recorded from various angles where the lighting conditions and low resolution make it more difficult to identify each action distinctly. Hence, action segmentation labels are added in [23] to improve the anticipation performance to 47.0%. Furthermore, the 50Salads dataset is shot from an overhead camera where all the actions are easily identifiable.

The observation period for Breakfast that produces the best performance is 15 seconds which is close to the average duration of actions in the dataset. So, observing the previous action in Breakfast provides enough temporal context for anticipating the next action as shown in [23] (a combination of 10, 15, 20 seconds produces the best result on Breakfast dataset). It also shows that using the latent goal improves the next action anticipation performance on the same features (I3D) with similar observation periods.

In Table 3, we compare state-of-the-art techniques on EPIC-KITCHENS55. The EPIC-KITCHENS55 is not an ideal setup for our model because there are multiple goals being pursued simultaneously. The actions from different goals are intertwined. So, our assumptions about actions in a sequence sharing the same latent goal are not justified for this dataset. On verb anticipation, latent goal is comparable to existing approaches in terms of Top-5 accuracy but not for Top-1 accuracy[1]. The unscripted nature of EPIC-KITCHENS55 involves people multi-tasking and changing their minds which makes the underlying goal difficult to identify. In such cases, using a latent goal can only help in approximately predicting the next verb, which explains why latent goal performs better at Top-5 anticipation. Interestingly, latent goal performs better than all other approaches for both Top-1 and Top-5 noun anticipation. There are many more objects compared to verbs (352 vs 125), and hence, the latent goals are limited for each object. Also, we use bag-of-object features that have high weights for 2 to 3 objects per frame during the observed action. One of the objects will most likely be used in the next action. So, the latent goal obtained with respect to objects (nouns) can be more precise than verbs and contribute to better noun anticipation.

Our top-5 noun and verb anticipation performances are comparable or better than prior state-of-the-art methods in both S1 and S2 sets. However, our action anticipation performance remains poor on both S1 and S2. This can be due to the fact that when our noun model is correct, it seems the verb model is not and vice-versa. This could also lead to poor action anticipation results. Secondly, 92% of the actions in the EPIC-KITCHENS dataset has fewer than 5 examples in total across training, validation, and the two test sets [5]. So, we do not encounter many of the actions while training our model. Our model overfits on the seen actions which leads to poorer action anticipation on unseen actions in the test sets.

[1]Evaluation results under username `royd`

Obs. duration	Mean-pooling	Max-pooling	LSTM
Breakfast			
3s	37.2	39.3	31.3
15s	*46.6*	**47.2**	*37.8*
25s	44.4	43.2	36.1
50 Salads			
3s	*54.7*	**59.6**	*52.8*
15s	40.8	47.2	39.3
25s	37.4	37.4	40.6

Table 4. Comparison of different temporal aggregation approaches for various observation periods.

4.4. Performance of aggregation methods

Table 4 compares three different temporal aggregation strategies - mean-pooling, max-pooling and LSTM. We have a single-layer LSTM with a 2048 hidden dimensions for Breakfast and 128 hidden dimensions for 50 Salads that produces the best results. Both max-pooling and mean-pooling outperform LSTM for aggregation and max-pooling performs the best. We hypothesize that taking the maximum of all feature dimensions across frames preserves salient features that maybe smoothed due to mean-pooling.

4.5. Impact of observation period

As the observation period changes the temporal context available for anticipation, we study the performance of different observation periods in Table 4. For the Breakfast dataset, increasing the observation period helps in anticipation till 15 seconds. Observing more than 15 seconds, i.e., more than the preceding action causes a deterioration in anticipation accuracy. Many activities in the Breakfast dataset like *making pancake*, *making fried eggs*, and *making scrambled eggs* share common actions. So, if these common actions are observed, it can cause confusion while predicting the next action. In the 50 Salads dataset, a shorter observation period of 3 seconds yields the best anticipation accuracy. So, very recent history towards the end of the preceding action is most informative for action anticipation on 50 Salads.

The actions in EPIC-KITCHENS55 dataset [4] are much more fine-grained compared to Breakfast and 50Salads. The median action segment duration is much shorter (8-16x) than either the Breakfast and 50Salads. As Table 6 shows, we can see that 0.5 seconds of observation is also sufficient for the next action anticipation. We observe a deterioration in performance when we increase the observed duration to 3 seconds for both verb and noun anticipa-

Obs. duration	Dense Trajectory			I3D		
	Mean	Max	LSTM	Mean	Max	LSTM
Breakfast						
10s	25.2	25.7	23.4	41.4	41.6	33.1
15s	26.2	*28.4*	25.2	44.4	**47.2**	36.2
50 Salads						
3s	36.4	*38.3*	36.1	*55.1*	**59.6**	*52.9*
5s	35.9	37.3	34.9	53.1	57.9	53.4

Table 5. Comparison of dense trajectory and I3D features

tion. All the results shown here are based on the max-pooling aggregation of features. The key difference in EPIC-KITCHENS55 is that we train separate networks for verbs and nouns. So, action anticipation refers to correctly predicting both the noun and the verb.

4.6. Impact of features

We compare the effect of features on Breakfast and 50Salads datasets in Table 5. We use 64-dimensional Fisher Vector representation of Dense Trajectory features (FVDT) provided by [1] for both Breakfast and 50Salad datasets. To accommodate the low-dimensional FVDT representation, we reduce the LSTM hidden dimensions to 64 for both the for 50 Salads and Breakfast dataset. I3D easily outperforms FTDV by a handsome margin for all temporal aggregation schemes which shows that choice of visual features is vital when anticipating the next action. For EPIC-KITCHENS55, we compare Temporal Segment Network (TSN) features on RGB frames (TSN-RGB), TSN features on optical flow (TSN-Flow), and bag-of-object (Object) features for our latent goal based approach. The ablation studies are conducted on the validation set of the EPIC-KITCHENS55. We obtain superior noun anticipation for Object features than both TSN-RGB and TSN-Flow. The bag-of-object features are histograms of detected objects in a frame normalized by the total number of appearances of every object in the entire training set. Hence, the bag-of-object features can explicitly emphasize which objects are in the frame compared to all objects. There is a often a direct correlation between the objects used in consecutive actions which explains the performance of bag-of-object features.

4.7. Component Validation

In this subsection, we show the impact of various components in our anticipation framework. Table 7 shows that computing new candidate-wise latent goals (CWLG) improves anticipation instead of just the initial goal estimate from the observed features. Further, if we check the consistency of the CWLG with the initial goal estimate, it substantially improves anticipation accuracy (3.6% for Breakfast and 6.1% for 50 Salads). Both these results show that our proposed model predicts unique CWLG for next action candidates that are different from the initial goal estimate. Though we choose the best candidate for the next action

Feature	Obs. Duration	Anticipation Accuracy		
		VERB	NOUN	ACTION
TSN-RGB	0.5s	29.21	14.09	05.21
TSN-Flow		28.45	12.32	03.36
Object		25.79	20.98	13.65
TSN-RGB	1s	30.94	13.32	02.54
TSN-Flow		27.96	12.32	04.67
Object		26.21	21.61	13.98
TSN-RGB	2s	**33.21**	13.21	04.34
TSN-Flow		29.06	12.25	05.24
Object		27.81	**22.14**	**14.13**
TSN-RGB	3s	30.82	12.23	04.12
TSN-Flow		28.73	12.01	04.26
Object		28.41	21.02	13.78

Table 6. Comparing performance of features on EPIC-KITCHENS55 validation set on Top-1 verb, noun, and action accuracy.

Candidate choice	Breakfast	50 Salads
w/o Candidate-wise Latent Goal (CWLG)	41.8	54.4
w/ CWLG but w/o consistency check (CC)	42.3	55.6
w/ CWLG and CC	**47.2**	**59.6**

Table 7. Effect of computing candidate-wise latent goal and goal consistency check while choosing action candidates.

Loss	Breakfast	50Salads
Cross-entropy ($\mathcal{L}_{ant}$)	44.8	54.1
$\mathcal{L}_{ant}$ + Obs. action ($\mathcal{L}_{obs}$)	45.8	54.2
$\mathcal{L}_{ant}$ + $\mathcal{L}_{obs}$ + Goal closeness ($\mathcal{L}_{close}$)	46.2	56.2
$\mathcal{L}_{ant}$ + $\mathcal{L}_{obs}$ + Goal consistency ($\mathcal{L}_{cons}$)	46.9	55.2
$\mathcal{L}_{ant} + \mathcal{L}_{obs} + \mathcal{L}_{close} + \mathcal{L}_{cons}$	**47.2**	**59.6**

Table 8. Impact of different losses on anticipation performance

using the new latent goal and its properties, we also need to enforce these conditions while training the network. Table 8 compares the effect of each loss on anticipation accuracy on both Breakfast and 50 Salads dataset. We can observe that adding the goal closeness loss and goal consistency loss to the cross-entropy loss improves the anticipation accuracy. The effect of these losses is more prominent when they are applied together. Particularly, goal closeness produces a larger improvement as it drives the network to favor those actions that produce a closer representation to the latent goal. A stable latent goal representation before and after anticipation also allows us to compute the closeness to the latent goal more accurately.

4.8. Varying number of action candidates

We compare the effect of considering different number of action candidates representations while predicting the anticipated action. A comparison between 5, 10, and 15 candidates reveals that 10 candidates are optimal for Breakfast and 15 for 50 Salads as shown in Table 9. As every video of 50Salads contains all 17 actions in the datasets, every action can be a possible next action, and having 15 candidates helps in such a case. On the other hand, Breakfast has an average of 6 actions per video, and choosing from a pool of 10 candidates is sufficient.

# Sampled candidates	Breakfast	50 Salads
5	44.1	58.5
10	**47.2**	58.9
15	45.5	**59.6**

Table 9. Effect of varying the number of sampled action representation candidates on anticipation

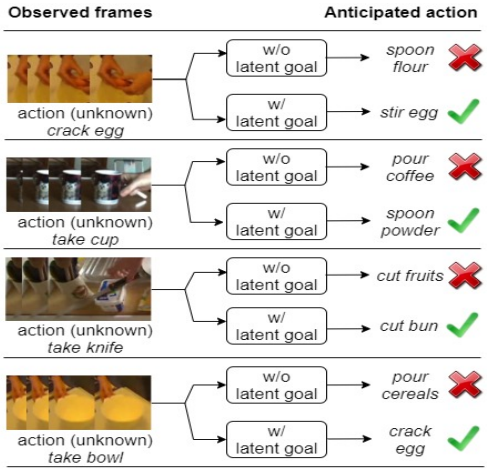

Figure 3. Examples of next action prediction with and without latent goal from the Breakfast dataset. Latent goal leads to the correct prediction of actions when there are multiple possibilities by capturing the intention. The observed action labels are not supplied to the network but are only mentioned for understanding.

4.9. Qualitative Results

We show the impact of latent goal on next action anticipation using some examples from the Breakfast in Figure 3. For each example, we replace the computation of latent goal ($\mathbf{x}_g^o$ in line 3 and $\mathbf{x}_g$ in line 8 of Algorithm 1) with a random vector during inference. We term this network as without (w/o) latent goal in Figure 3. All the examples in Figure 3 demonstrate that the latent goal helps in predicting the correct next action. For example, *crack egg* leads to an incorrect prediction of *spoon flour* without latent goal but leads to the correct prediction of *stir egg* when latent goal is used.

We also study goal consistency between observed and anticipated actions through qualitative examples in Figure 4. For most of the observed and anticipated action pairs having close latent goal representation leads to correct anticipation. However, in some cases like (*pour_milk*, *stir_dough*) or (*pour_dough2pan*, *fry_pancake*) having close latent goals does not lead to correct anticipation. So, goal consistency is not always enough to predict the next action accurately.

5. Conclusion

Human activities are guided by an implicit or explicit goal that determines the sequence of actions performed to achieve the goal. We propose a novel technique to characterize the goal using a latent representation for action anticipation. We exploit latent goal properties, namely, goal closeness and goal consistency, to predict the next action. Our experiments on both scripted datasets - 50Salads and Breakfast, and regular activities dataset - EPIC-KITCHENS55 show that using latent goal achieves state-of-the-art action anticipation performance. We show that the proposed goal closeness and goal consistency losses improve action anticipation and qualitative results show a latent goal helps to make correct choice when there are multiple choices for the next action.

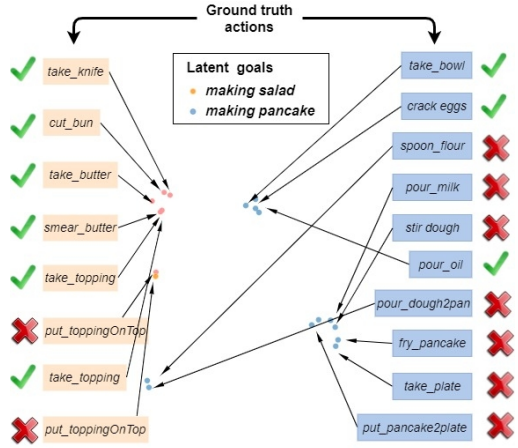

Figure 4. Goal consistency vs. action anticipation. For most of the observed and anticipated action pairs having close latent goal representation leads to correct anticipation.

Acknowledgment

This research/project is supported in part by the National Research Foundation, Singapore under its AI Singapore Program (AISG Award No: AISG2-RP-2020-016) and the National Research Foundation Singapore under its AI Singapore Program (Award Number: AISG-RP-2019-010).

References

[1] Y. Abu Farha, A. Richard, and J. Gall. When will you do what?-anticipating temporal occurrences of activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5343–5352, 2018.

[2] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun. Anticipating accidents in dashcam videos. In *Asian Conference on Computer Vision*, pages 136–153. Springer, 2016.

[3] C.-Y. Chang, D.-A. Huang, D. Xu, E. Adeli, L. Fei-Fei, and J. C. Niebles. Procedure planning in instructional videos. In *European Conference on Computer Vision*, pages 334–350. Springer, 2020.

[4] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. The epic-kitchens dataset: Collec-

tion, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[5] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020.

[6] P. Dendorfer, A. Osep, and L. Leal-Taixé. Goal-gan: Multimodal trajectory prediction based on goal position estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[7] Y. A. Farha and J. Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.

[8] B. Fernando and S. Herath. 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*, pages 13224–13233, 2021.

[9] A. Furnari and G. M. Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6252–6261, 2019.

[10] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Forecasting future action sequences with neural memory networks. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 298. BMVA Press, 2019.

[11] R. Girdhar and K. Grauman. Anticipative video transformer. *arXiv preprint arXiv:2106.02036*, 2021.

[12] Q. Ke, M. Fritz, and B. Schiele. Time-conditioned action anticipation in one shot. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9925–9934, 2019.

[13] H. Koppula and A. Saxena. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *International conference on machine learning*, pages 792–800, 2013.

[14] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2015.

[15] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014.

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

[17] A. Miech, I. Laptev, J. Sivic, H. Wang, L. Torresani, and D. Tran. Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[18] M. Oliu, J. Selva, and S. Escalera. Folded recurrent neural networks for future video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 716–731, 2018.

[19] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks: Proceedings of the second international conference on learning representations (iclr 2014). In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[20] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6262–6271, 2019.

[21] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Pedestrian action anticipation using contextual feature fusion in stacked rnns. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 171. BMVA Press, 2019.

[22] D. Roy and B. Fernando. Action anticipation using pairwise human-object interactions and transformers. *IEEE Transactions on Image Processing*, 2021.

[23] F. Sener, D. Singhania, and A. Yao. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer, 2020.

[24] Y. Shi, B. Fernando, and R. Hartley. Action anticipation with rbf kernelized feature mapping rnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 301–317, 2018.

[25] S. Stein and S. 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.

[26] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident db. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3521–3529, 2018.

[27] H. Tran, V. Le, and T. Tran. Goal-driven long-term trajectory prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 796–805, 2021.

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

[29] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[30] X. Wei, P. Lucey, S. Vidas, S. Morgan, and S. Sridharan. Forecasting events using an augmented hidden conditional random field. In *Asian Conference on Computer Vision*, pages 569–582. Springer, 2014.

[31] D. Xie, T. Shu, S. Todorovic, and S. C. Zhu. Learning and inferring "dark matter" and predicting human intents and trajectories in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1639–1652, 2018.