Forecasting Future Sequence of Actions to Complete an Activity

Yan Bin Ng  
A*AI, A*STAR  
ng_yan_bin@scei.a-star.edu.sg  

Basura Fernando  
A*AI, A*STAR  
fernando_basura@scei.a-star.edu.sg  

Abstract

Future human action forecasting from partial observations of activities is an important problem in many practical applications such as assistive robotics, video surveillance and security. We present a method to forecast actions for the unseen future of the video using a neural machine translation technique that uses encoder-decoder architecture. The input to this model is the observed RGB video, and the target is to generate the future symbolic action sequence. Unlike most methods that predict frame or clip level predictions for some unseen percentage of video, we predict the complete action sequence that is required to accomplish the activity. To cater for two types of uncertainty in the future predictions, we propose a novel loss function. We show a combination of optimal transport and future uncertainty losses help to boost results. We evaluate our model in three challenging video datasets (Charades, MPII cooking and Breakfast). We outperform other state-of-the-art techniques for frame based action forecasting task by 5.06% on average across several action forecasting setups.

1. Introduction.

We humans forecast others’ actions by anticipating their behavior. For example by looking at the video sequence in figure 1, we can say “the person is going towards the fridge, then probably he will open the refrigerator and take something from it”. Our ability to forecast comes naturally to us. We hypothesize, humans analyze the visual information to predict the plausible future actions, also known as the mental time travel [27]. One theory suggests humans’ success in evolution is due to the ability to anticipate the future [27]. Perhaps, we correlate prior experiences and examples with the current scenario to perform mental time travel.

Recently, the human action prediction/forecasting problem has been extensively studied in Computer Vision and AI community. The literature on this prediction topic can be categorized as early action [8], activity [1, 13], and event prediction [33]. In early human action prediction, methods observe an ongoing human action and aim to predict the action in progress as soon as possible [9] before it finishes. This problem is also known as action anticipation in the literature [23]. As these methods predict an on going action before it finishes, they are useful for applications when future planning is not a major requirement. In contrast, activity prediction aims at forecasting future action as soon as possible (not necessarily in the temporal order) and are useful in many robotic applications, e.g., human robotic interaction. These methods can facilitate information for some level of future planning [11]. In activity prediction, some methods observe p% of the activity and then predict actions for q% of the future frames in the video. Most interestingly, these methods predict actions per-frame which limits their practical application in many cases [1]. The most of these methods make the assumption about length of the video implicitly or explicitly [1]. Alternatively, some methods observe k number of actions in an activity and then predict only the next future action [15]. However, we humans are able to forecast the future series of actions which allows us to plan for the future, (e.g. if some one is going to cook a simple potato dish, probably we will see a sequence of actions such as peel \( \triangleright \) cut \( \triangleright \) wash \( \triangleright \) boil). We humans are able to predict the future irrespective of video length or the number of frames. We aim to solve this challenging problem of forecasting future sequence of actions to complete an activity from the partial observations of the activity.

In this paper we observe only a handful of actions within a long activity. Then we forecast the sequence of actions for the future without making any assumptions on the length of video. This type of problems arise in practice, specially in robotics, e.g., robot assisted industrial maintenance, and assistive robotics in health care. In contrast to majority of action anticipation and activity prediction methods, ours is trained to predict the future action sequence. To solve this
problem, there are several challenges that we need to tackle. First, our method needs to implicitly infer the goal of person performing the activity. Second, it should learn to what extent the person has completed the activity. Finally, it has to infer what other actions are needed to accomplish the activity.

We formulate our solution such that all of this is learned in a data driven manner. Specifically, we make use of complex relationship between observed video features and the future actions to learn a complex mapping between them. To facilitate that, we formulate it as a neural machine translation problem where the input is an observed RGB video and the target is a symbolic sequence of future actions. Specifically, we use a recurrent encoder-decoder architecture. Each future action depends on the past observed feature sequence and interestingly, some of the observed features are important in determining the future actions more than others. For example, if our model predicts ”adding sugar” as the future action, then it is more likely that our model give a higher attention weight to frames having a cup or a mug. Therefore, we make use of mechanism that allows us to align-and-attend past when generating future actions. Furthermore, the uncertainty of predictions increases with two factors; first the amount of observed data model observe, and the second, how far into the future our model predicts. If model observe more data, perhaps the predictions are likely to be reliable. Moreover, if model predicts far into the future, then predictions are likely to unreliable. We develop a novel loss function that allows us to consider these two factors and extend the traditional cross-entropy loss to cater for these uncertainties.

Finally, we also make use of optimal transport loss which allows us to tackle exposure bias issue of this challenging sequence-to-sequence machine translation problem. Exposure bias arises when we use cross-entropy loss to train neural machine translation models where it provides an individual action-level training loss (ignoring the sequential nature) which may not be suitable for our task. The optimal transport loss is a more structured loss that aims to find a better matching of similar actions between two sequences, providing a way to promote semantic and contextual similarity between action sequences. In particular, this is important when forecasting future action sequences from observed temporal features.

In a summary, our contributions are as follows:

- We demonstrate the usefulness of optimal transport and the uncertainty losses.
- We extensively evaluate our method on three challenging action recognition benchmarks and obtain state of the art results for action forecasting.

2. Related work.

We categorize the related work into three, 1. early action prediction and anticipation, 2. activity prediction and 3. machine translation.

**Early action prediction and anticipation:** Early action prediction aims at classifying the action as early as possible from partially observed action video. Typically, experiments are conducted on well segmented videos containing a single human action. In most prior work, methods observe about 50% of the video and then predict the action label [22, 10, 23]. In particular these methods can be categorized into four types. Firstly, there are methods that generate features for the future and then use classifiers to predict actions using generated features [24, 30]. Feature generation for future action sequences containing a large number of actions is a challenging task and therefore, not feasible in our case. Secondly, the methods presented in [23, 7, 14] develop novel loss functions to cater for uncertainty in the future predictions. Our work also borrows some concepts from these methods to develop loss functions but ours is applied over the future action sequence in contrast to applying over an action as in [23, 7, 14]. Thirdly, some anticipation methods generate future RGB images [34, 31] and then classify them into human actions using convolution neural networks. However, generation of RGB images for the future is a very challenging task specially for longer action sequences. Similarly, some methods aim to generate future motion images [20] and then try to predict action for the future. However, we aim to forecast action sequences for unseen part of the human activity and is more challenging than action anticipation. Therefore, action anticipation methods can not be used to solve our problem.

**Activity prediction:** The most of the activity prediction methods aim at predicting the next action in the sequence [15, 18] or focus on first person human actions [19, 3]. Some methods assume that the number of future frames is given and they try to predict the action label for each future frame [1, 6]. This method is similar to our work but we aim to predict the future sequence of action (e.g. wash ⊃ clean ⊃ peel ⊃ cut) instead of assigning a label for each future frame. Specifically, [15] used to predict the next action using previous three actions using motion, appearance, and object features with a two layered stacked LSTM. Authors in [18] use stochastic grammar to predict the next action in the video sequence. Even though these methods can be extended to predict the sequence of actions by recursively
applying their methods, we face two challenges. Firstly, errors may propagate into the future making future actions more wrong, and secondly it may not know when to stop producing action symbols, which is important when the actions are part of some larger activity. Sequence-to-sequence machine translations are naturally able to address both these two issues [28].

**Machine translation.** Our method is also related to machine translation methods [28, 2, 29, 32]. However, none of these works use machine translation for action sequence forecasting from videos. Typically, machine translation is used for language tasks [28, 2]. To the best of our knowledge, we are the first to use neural machine translation for translating a sequence of RGB frames (a video) to a sequence of future action labels. Indeed, machine translation has been used for unsupervised learning of visual features [26] in prior work which is related to us. But they did not use it for predicting future action sequences.

### 3. Future action sequence prediction.

#### 3.1. Problem

We are given a video in which a human is performing an activity. Our model only observe the initial part of the video containing initial sequence of actions. The objective of this work is to train a model to predict the future unseen sequence of actions. A visual illustration of this model is shown in figure 2. Let us denote the observed RGB video by $X^o = (x_1^o, x_2^o, x_3^o, \cdots, x_p^o)$ where $x_p^o$ is the $p^{th}$ frame. The observed action sequence is denoted by $Y^o = (y_1^o, y_2^o, \cdots, y_p^o)$ (note that $p \neq P$) and the future unseen ground truth action sequence by $Y^u = (y_1^u, y_2^u, \cdots, y_N^u)$ where each action $y \in \mathcal{Y}$ and $\mathcal{Y}$ is the set of action classes and the start time of each action $y_i$ is before or equal to the start time of $y_{i+1}$.

In contrast to other action forecasting methods that operate at frame level (or clip level), we do not know the label of each observed RGB frame $x_p^o$. Our model has access to frame sequence $X^o$ only. We train a model $\phi(\Theta)$ that predicts unseen action sequence $Y^u$ from seen RGB sequence $X^o$ where $\Theta$ are the parameters of the model, i.e. $Y^u = \phi(X^o, \Theta)$. We do not make use of ground-truth action sequence $Y^o$ during training or inference. Our objective is to predict the future action sequence $Y^u$. Therefore, our method does not need any frame level action annotations as in prior action forecasting methods [1, 6].

#### 3.2. High level solution

We formulate this problem as a sequence-to-sequence machine translation problem [28, 2, 29, 32] where we use observed rgb sequence $X^o$ as the input sequence. Then the symbolic unseen action sequence $Y^u$ is the target sequence. Specifically, we use an GRU-based encoder-decoder architecture. Our hypothesis is that the encoder-decoder machine translation would be able to learn the complex relationship between seen feature sequence and future actions. To further improve the model predictive capacity, we also use attention over encoder hidden state when generating action symbols for the future and use novel loss functions to tackle uncertainty. Next we describe our model in detail.

#### 3.3. GRU-encoder-decoder

We use GRU-based encoder-decoder architecture for translating video sequence into future action sequence. Our encoder consists of a bi-directional GRU cell. Let us first define the encoder GRU cell which takes the seen feature sequence $X^o = (x_1^o, x_2^o, \cdots, x_p^o)$ which consists of $p$ elements as input. We define the encoder GRU by $f_e(.)$ for time step $t$ as follows:

$$\overrightarrow{h}_t, \overleftarrow{h}_t = f_e(x^o_t, \overrightarrow{h}_{t-1}, \overleftarrow{h}_{t-1}) \tag{1}$$

where $\overrightarrow{h}_t, \overleftarrow{h}_t \in R^D$ are the forward and backward hidden states at time $t$. The initial hidden state of the encoder GRU is set to zero. Then we make use of a linear mapping $W_e \in R^{2D \times D}$ to generate a unified representation of both forward and backward hidden states for each time step $\overrightarrow{h}_t, \overleftarrow{h}_t$ as follows:

$$h_t = [\overrightarrow{h}_{t-1}, \overleftarrow{h}_{t-1}] \times W_e \tag{2}$$
where $[\cdot, \cdot]$ indicates the concatenation of forward and backward hidden states. Therefore, the outcome of the encoder GRU is a sequence of hidden state vectors denoted by $H = \langle h^o_1, h^o_2, \ldots, h^o_p \rangle$. The bi-directional GRU encodes more contextual information which might inherently enable the model to infer the intention of person doing the activity.

The decoder is a forward directional GRU $f_d(\cdot)$, that generates the decoder hidden state $g_q \in \mathbb{R}^D$ at decoding time step $q$ define as follows:

$$g_q = f_d([c_{q-1}, \hat{y}_{q-1}], g_{q-1})$$  \hfill (3)

where $\hat{y}_{q-1}$ is the predicted target action class score vector at step $q-1$. The input to decoder GRU $f_d(\cdot)$ at time step is a concatenation of the context vector $c_{q-1}$ and the previously predicted action score vector $\hat{y}_{q-1}$ denoted by $[c_{q-1}, \hat{y}_{q-1}]$. We obtain the action score vector at step $q$ of the decoder using following linear mapping:

$$\hat{y}_q = g_q \times U$$  \hfill (4)

where $U \in \mathbb{R}^{D \times |Y|}$ is a learnable parameter. Note that the output symbol at step $q$ of the decoder is obtained by argmax operator, i.e., $\hat{y}_q = \text{argmax} \ \hat{y}_q$. The decoder is initialized by the final hidden state of the encoder (i.e. $g_0 = h^o_p$ where $h^o_p$ is the final hidden state of the encoder). The initial symbol of the decoder is set to SOS (start of sequence symbol) during training and testing. The decision to include the previous predicted action $\hat{y}_{q-1}$ as an input in the decoder is significant as now the decoder model has more semantic information during the decoding process. Once chosen would be to simply ignore the previously prediction action symbol. However, that would hinder the predictive capacity of the decoder as decoder is not explicitly aware of what it produced in the previous time step. Conceptually, now the decoder is trying to find the most likely next symbol $P(\hat{y}_q | \hat{y}_{q-1}, g_{q-1})$ using both previous symbol and the contextual information.

Next we describe how to generate the context vector $c_{q-1}$ which summarizes the encoder-decoder hidden states using attention mechanism.

### 3.4. Attention over encoder hidden state

It is intuitive to think that not all input features contribute equally to generate the output action symbol $\hat{y}_q$ at decoder step $q$. Therefore, we propose to make use of attention over encoder hidden states $H$ to generate the context vector $c_{q-1}$ which serves as an input to the decoder GRU. Specifically, to generate $c_{q-1}$, we linearly weight the encoder hidden vectors $H = \langle h^o_1, h^o_2, \ldots, h^o_p \rangle$, i.e.,

$$c_q = \sum_i \frac{\exp(a^q_i)}{\sum_j \exp(a^q_j)} h^o_i$$  \hfill (5)

where $a^q_i$ is the weight associated with the encoder hidden state $h^o_i$ to obtain $q$-th context vector define by following equation.

$$a^q_i = \text{tanh}(h^o_i ; g_q) \times W_{\text{att}} \times V$$  \hfill (6)

Here $W_{\text{att}} \in \mathbb{R}^{2D \times D}$ and $V \in \mathbb{D}$ are learnable parameters and $a^q_i$ depends on how well the encoder-decoder hidden states $h^o_i, g_q$ are related. This strategy allows us to attend all encoder hidden states $H = \langle h^o_1, h^o_2, \ldots, h^o_p \rangle$ when generating the next action symbol using decoder GRU. During training we make use of teacher forcing strategy to learn the model parameters of the encoder-decoder GRUs where we use $y_q$ instead of $\hat{y}_q$ in equation 3 half of the time. This is to make sure that, the inference strategy is not too far away from the training strategy. During inference, given the input features sequence, we pass it thorough the encoder-decoder to generate future action sequence until we hit the end-of-sequence symbol (EOS). The model is also trained with start-of-sequence (SOS) symbol and EOS.

### 3.5. Tackling the uncertainty

Correctly predicting the future action sequence from a partial video is challenging as there are more than one plausible futures action sequences. This uncertainty increases with respect to two factors: 1. to what extent we have observed the activity, (more we observe, more information we have to make future predictions) 2. how far into the future we are going to predict using observed data (if we predict too far into the future, there are more possibilities and more uncertainty). To tackle these two factors, we propose to modify cross-entropy loss which is typically used in sequence-to-sequence machine translation\(^1\). Let us assume that we have observed $P$ number of actions, and we are predicting a total of $N$ number of action symbols. Let us denote the cross-entropy loss between the prediction ($\hat{y}_q$) and the ground truth ($y_q$) by $L(\hat{y}_q, y_q)$. Then our novel loss function that handles the uncertainty ($L_{\text{un}}(\hat{Y}^u, Y^u)$) for a given video $X^u, Y^u$ is define by

$$L_{\text{un}} = (1 - \exp(-P/N)) \sum_{q=1}^{N} \exp(-q)L(\hat{y}_q, y_q)$$  \hfill (7)

where the term $(1 - \exp(-P/N))$ takes care of shorter observations and makes sure that longer action observations contributes more to the loss function. If the observed video contains less actions (information), then predictions made by those are not reliable and therefore does not contribute much to the overall loss. Similarly, the second inner term $\exp(-q)L(\hat{y}_q, y_q)$ makes sure that the predictions that are too far into the future make only a small contribution to the loss. If our model makes a near future prediction, then possibly model should do a better job and if it is

---

\(^1\)This strategy may be applicable to other loss functions as well.
makes an error, we should penalize more. During training, we make use of sequential data augmentation to better exploit the above loss function. In-fact, for given a training video consist of $M$ actions (i.e. $Y = \{y_1, y_2, \cdots, y_M\}$), we augment the video to generate $M-1$ observed sequences where $Y^o = \{y_1^o, y_2^o, \cdots, y_n^o\}$ and $Y^n = \{y_{t+1}^n, \cdots, y_M^n\}$ for $t \in \{1, \cdots, M-1\}$. Then we train our networks with these augmented video sequences with the uncertainty loss.

### 3.6. Optimal Transport Loss (OT)

Optimal transport defines a distance measure between probability distributions over a metric space. We want to exploit the metric structure in the action sequence space, but using cross entropy loss alone does not account for the metric structure between the actions of a sequence. The issue with using the cross-entropy loss is that the loss at step-q in the decoder only relies on the ground-truth action at step-q. However, overall encoder-decoder model should generate the target future action sequence and we should consider it as a structured task. Unfortunately, the element-wise cross-entropy loss does not take this sequence-to-sequence structural nature of the task.

We propose to make use of optimal transport loss of [17] defined by

$$D_c(\mu, \nu) = \inf_{\gamma \in \Pi(\mu, \nu)} \mathbb{E}_{x \sim \mu, y \sim \nu} c(x, y)$$

where $\Pi(\mu, \nu)$ is the set of all joint distributions $\gamma(x, y)$ with marginals $\mu(x)$ and $\nu(y)$ and $c(x, y)$ is the cost function for moving $x$ to $y$ in the sequence space.

Specifically, we consider the optimal transport distance between two discrete action distributions $\mu, \nu \in \mathbb{P}(\mathcal{A})$ of the action sequences where $\mathcal{A}$ is the action space. The discrete distributions $\mu, \nu$ can be written as weighted sums of Dirac delta functions i.e. $\mu = \sum_{i=1}^{m} u_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{n} v_j \delta_{y_j}$ with $u_i = \sum_{j=1}^{n} v_j$. Given a cost matrix $C \in \mathbb{R}_{+}^{m \times n}$ where $C_{ij}$ is the cost from $x_i$ to $y_j$, the optimal transport loss is equivalent to

$$L_{ot}(\mu, \nu) = \min_{P \in \Pi(\mu, \nu)} \sum_{i,j} P_{ij} C_{ij}$$

where $\Pi(\mu, \nu) = \{ P \in \mathbb{R}_{+}^{m \times n} | P 1_m = \mu, P^T 1_n = \nu \}$ and $1_n$ is an n-dimensional vector of all ones.

We use the Sinkhorn algorithm implementation proposed in [5] to compute the optimal transport loss between the predicted and ground-truth action sequences. Let us denote the optimal transport loss by $L_{ot}(\hat{Y}^u, Y^u)$. The combination of both losses is given by the following

$$L(\hat{Y}^u, Y^u) = L_{un}(\hat{Y}^u, Y^u) + \beta \times L_{ot}(\hat{Y}^u, Y^u)$$

where $\beta$ is the trade-off parameter.

### 4. Experiments

In this section we extensively evaluate our model using three challenging action recognition datasets, namely the Charades [25], MPII Cooking [21] and Breakfast [12] datasets. Next we give a brief introduction to these datasets. MPII-Cooking Dataset has 65 fine-grained actions, 44 long videos with a total length of more than 8 hours. Twelve participants interact with different tools, ingredients and containers to make a cooking recipe. We use the standard evaluation splits where total of five subjects are permanently used in the training set. Rest six of seven subjects are added to the training set and all models are tested on a single subject and repeat seven times in a seven fold cross-validation manner. In this dataset, there are 46 actions per video on average.

Charades dataset has 7,985 video for training and 1,863 videos for testing. The dataset is collected in 15 types of indoor scenes, involves interactions with 46 object classes and has a vocabulary of 30 verbs leading to 157 action classes [25]. On average there are 6.8 actions per video which is much higher than other datasets having more than 1,000 videos.

Breakfast dataset [12] consist of 1,712 video where 52 actors making breakfast dishes. There are 48 fine-grained actions classes and four splits. On average each video consists of 6.8 actions per video.

There are no overlapping actions in Breakfast and Cooking datasets. Charades has handful of videos with overlapping actions. To generate ground truth action sequences, we sort the list of actions by start times, ignoring the end time of the actions.

#### Performance evaluation measures:

We measure the quality of generated future action sequences using BLEU-1 and BLEU-2 scores [16]. These are commonly used in other sequence evaluation tasks such as image captioning. We use the standard BLEU score definition proposed in the Machine Translation and Natural Language Community [16] which is also publicly implemented in Python nltk toolbox. We also report sequence-item classification accuracy which counts how many times the predicted sequence elements match the ground truth in the exact position. Furthermore, we also report the mean average precision (mAP) which does not account for the order of actions. To calculate mAP, we accumulate the action prediction scores of the unseen video and compare it with the ground truth. BLEU-1, BLEU-2 and sequence-item classification accuracy reflects the sequence forecasting performance while the mAP only accounts for holistic future action classification performance discarding the temporal order of actions.

#### Feature extraction and implementation details:

Unless specifically mentioned, we use effective I3D features [4] as the video representation for all datasets. First,
we fine-tune I3D networks for video action classification using provided video level annotations. Afterwards, we extract 1024-dimensional features to obtain a feature sequence for each video.

### 4.1. Evaluating our model

In this section we evaluate various aspects of our model aiming to provide some insights to the reader.

#### How well it performs in action forecasting?

In this section we evaluate our main contribution using all three datasets. During training, for each given video \( X \) and the action sequence \( Y = \langle y_1, y_2, \ldots, y_N \rangle \), our model takes feature sequence \( X^o \) corresponding to observed action sequence \( Y^o = \langle y^o_1, \ldots, y^o_i \rangle \) and then predict future action sequence \( Y^u = \langle y^u_{i+1}, \ldots, y^u_N \rangle \) for all \( i \) values (i.e. for \( i = 1, \ldots, N - 1 \)). Note that the \( i \)-th action symbol is denoted by \( y^u_i \), and corresponds to a real action e.g. "opening fridge". We use the same action sequence sampling strategy to evaluate on test videos for all possible \( i \) values. Unless otherwise specified, we use this strategy for training and testing which we call as the **Action Forecasting Setup**. With this augmentation strategy, we obtain much larger dataset for training and evaluation.

We report results using our GRU-based encoder-decoder model trained with attention and cross-entropy loss for action sequence forecasting. As a baseline, we report results for random performance. In this case, for a given video, we randomly generate the next score vector to obtain the next action symbol for the unseen sequence. As the second baseline, we report results using the entire video sequence to classify the full action sequence denoted by **Classification Setup**. Our sequence classification model uses the same sequence-to-sequence machine translation model trained with attention and cross-entropy loss. Results obtained by sequence classification model serves as a soft upper bound for the action forecasting model. As there are no prior work to compare with, we report results only with our method.

From the results shown in table 1, first, our model performs significantly better than the random performance. Sequence item classification accuracy (which is a strict measure) reflects the difficulty of action sequence forecasting task. In the forecasting setup, we obtain item classification accuracy of 2.60, 4.50, and 21.29 where the random performance is 0.28, 0.47, and 0.70 on Charades, MPII cooking and Breakfast respectively. The random performance indicates the difficulty of forecasting task. Our model is 10-30 times better than random performance.

The difference in results between classification and forecasting setups is not too drastic, specially for Breakfast and Charades. Our classification model obtains seq. item accuracy of 5.35 while our forecasting model reach 2.60 on Charades. Similarly for MPII cooking dataset, the classification model obtains 14.86 and our action forecasting model’s performance is 4.50. Interestingly, seq. item classification accuracy of 26.35 and 21.29 is obtained for classification and forecasting models respectively on Breakfast.

For action forecasting task, the Charades dataset is the most challenging and the least is Breakfast dataset. Interestingly, for BLEU-2, the classification model obtains 2.78 while future action forecasting model performs better on Charades dataset. These results indicate the effectiveness of our method for future action sequence forecasting task. However, these results also suggests that there is more to do. Later in the experiments, we show how to improve these results.
Table 1. GRU Encoder-Decoder performance on action sequence forecasting

| Dataset | Setup               | BLEU-1 (%) | BLEU-2 (%) | Seq. Item. Acc. (%) | mAP  |
|---------|---------------------|------------|------------|---------------------|------|
| Charades| Random              | 1.04       | 0.55       | 0.25                | 4.40 |
| Charades| Classification      | 15.26      | 2.78       | 5.35                | 28.40|
| Charades| Forecasting with Att.| 7.95       | 2.87       | 2.60                | 6.10 |
| MPII-Cooking| Random       | 1.28       | 0.98       | 0.47                | 8.35 |
| MPII-Cooking| Classification  | 25.73      | 14.34      | 14.86               | 20.86|
| MPII-Cooking| Forecasting     | 8.70       | 4.10       | 4.50                | 10.80|
| Breakfast| Random            | 1.38       | 0.99       | 0.50                | 5.53 |
| Breakfast| Classification    | 51.83      | 37.34      | 26.15               | 46.99|
| Breakfast| Forecasting       | 34.56      | 21.15      | 21.29               | 30.24|

Table 2. Performance comparison for predicting next action. MLP is the multiple layered perceptron.

| Method | Charades | MPII Cooking | Breakfast |
|--------|----------|--------------|-----------|
| MLP    | 3.9      | 1.7          | 7.1       |
| LSTM   | 2.5      | 1.3          | 2.4       |
| Our    | 6.8      | 3.0          | 11.0      |

How does it work for predicting the next action?

In this section, we evaluate the impact of our sequence-to-sequence encoder-decoder architecture for predicting the next action. For a given observed sequence $Y^o$, the objective is to predict the next action $y_{i+1}$ for all $i$ values of the video. Once again $y^o_i$ is the $i$-th action of the video. As before, we generate all train and test action sequences. For comparison, we also use two layered fully connected neural network (MLP) which applies mean pooling over the observed features and then use MLP as the classifier. Similarly, we also compare with a standard LSTM which takes the input feature sequence and then predict the next action only. For our method and two baselines (LSTM, MLP), we use the same hidden size of 512 dimensions. For all models, we use the same activation function, i.e. $\tanh()$. We report results in table 2.

First, we see that MLP obtains better results than LSTM. Second, our sequence-to-sequence method with attention performs better than both LSTM and MLP methods. MLP obtains 1.7 mAP for predicting the next action indicating features do not contain enough information about future and more complicated mechanism is needed to correlate past features with the future action. Our method obtains far better results than these two baselines indicating the effectiveness of our sequence-to-sequence architecture for next action prediction task. We conclude our model is better suited for future action prediction than MLP and LSTM.

What is the impact of loss functions?

In this section we evaluate our method using the uncertainty and optimal transport loss functions for action sequence forecasting setup. The uncertainty loss consists of two parts in equation 7, 1. the effect of the fraction of past observations $(1 - \exp(P/N))$ denoted by $L_{\text{un}}$, past-only, and 2. the extent of future predictions $(\exp(-q))$ denoted by $L_{\text{un}}$, future-only. First, we analyze the impact of these two terms separately and then evaluate them jointly. We also demonstrate the impact of optimal transport loss alone ($L_{\text{ot}}$). Finally, we evaluate combination of all losses where we set the $\beta$ of equation 10 to be 0.001. Results are reported in table 3.

From the results in table 3, we see that both uncertainty and optimal transport losses are more effective than the cross-entropy loss which justifies our hypothesis about these new loss functions. Interestingly, the loss term($L_{\text{un}}$, future-only), obtains the best results for BLEU scores while OT loss obtain best action sequence classification accuracy for an individual loss. The combination of two uncertainty losses perform better than individual ones. Combination of both $L_{\text{ot}}$ and $L_{\text{un}}$, future-only perform much better than all others obtaining a significant improvement in BLEU-1 and BLEU-2 scores from best of 7.95 to 9.59 and 2.87 to 3.92 on Charades dataset. Similar trend can be seen for MPII-Cooking dataset as well where we see consistent improvements. This interestingly shows that optimal transport loss and $L_{\text{un}}$, future-only are complimentary to each other. Though two uncertainty losses perform better than cross-entropy loss, unfortunately, the combination of all three losses do not seem to be useful. Perhaps we need a better way to combine both uncertainty losses with the OT loss which we leave for further investigation in the future.

We visualize some of the obtained results in figure 3. Interestingly, our method is able to generate quite interesting future action sequences. In the first example, our method accurately obtain four out of five actions. In the second example, it predicts two actions correctly, however the predicted action sequence seems plausible though it is not correct.

What if we only rely on three previous actions?

In this experiment we evaluate the performance of our model when we predict the next action using only the three previous actions. Here we train and test our method using all augmented action sequences. As before we use 3D features from the seen three actions and aims to predict the next action class. We also compare traditional cross-entropy with ($L_{\text{ot}}$ + Cross-entropy) loss. Results are reported on table 4. First, even for our method, we see a
drop in performance from the results reported in previous experiment in table 2. When we predict the next action using all previous action features, with the cross-entropy loss, we obtain a classification accuracy of 6.8% in table 2 whereas, in table 4, our cross-entropy method obtains 3.54% only. This suggests that it is better to make use of all available information from observed video features and just let the attention mechanism to find the best features. Analysis on the impact of attention is evaluated in the next section and also in the supplementary material (Table 2 of Supp. Mat.). Secondly, the optimal transport loss combined with cross-entropy loss improve results indicating it is complimentary even in this constrained case. For this experiment there is no need to make use of uncertainty loss as there is only one action to predict.

### 4.2. Comparison to other SOA methods.

The action forecasting problem we study in this paper is different from what has been explored in the literature. We focus on forecasting the future action sequence whereas most recent methods in the literature take a somewhat different approach [1, 6]. These methods observe p% of the video and aims to predict future actions for q% of the video assuming length of video is known and frame level action annotations (at least the start and end of each action is known) are provided. In contrast, our default setting does not make these assumptions and we just need the action sequence without precise temporal extent of each action. To compare with these existing methods [1, 6], we train our model (without using EOS symbol) to generate q% of predictions for a given video after observing p% of frames. We compare our results with [1] and report mean per class accuracy as done in [1]. Unfortunately, the method in [6] use ground truth action sequence labels (the observed actions) which is not a realistic setup (nevertheless, we also report results using ground truth sequence in the supplementary materials (Table 1) where our method outperforms [6]). Here, we compare with [1] using mean-per class accuracy and use features to forecast the future actions. This setup is the most realistic in practice. Results are reported in table 5.

Interestingly, our method outperforms all baselines presented in [1] by a large margin, including larger prediction percentages such as 0.5. On average, we obtain an improvement of 5.06 over the prior best methods [1]. Specifically, the biggest average improvement is obtained when we observe only the 20% of video. In this case, the average improvement is 5.96 across all prediction percentages. We also see a consistent improvement over all (p%) percentages. We also see that our attention mechanism helps to improve results, specially for larger prediction percentages. Visual illustration of some predictions are shown in figure 4. Interestingly, most of the time our method is able to get the action class correctly, although the temporal extent is not precise. Furthermore, there is significant smoothness in the prediction that we believe is due to the sequential learning used in our method.

### 5. Conclusion.

In this paper we present a method to predict future action sequence for a given video. To do that, we use a GRU-based encoder-decoder sequence-to-sequence machine translation technique. We show the effectiveness of regularizing the cross-entropy loss for this task by catering the uncertainty of future predictions. The optimal transport loss also allows us to further improve results. We observe that conditioning on few past video frames is not sufficient to predict or forecast future actions or action sequences accurately. It is better to make use of all available information and use attention mechanism to select the most relevant ones. This also allows the model to better understand the context of activity. By extending our method slightly, we also compare with already existing action forecasting methods that predict future actions for some potion of the video. In this case, our method is able to outperform prior methods by 5.06% on average. We also demonstrate the effect of attention mechanism for this task.

Conceptually, we are the first to investigate action sequence forecasting problem for a given partial observation of an activity. We believe our findings are insightful and useful for the development of future methods.

**Acknowledgment:** This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: A1S5-RP-2019-010).

---

**Table 4.** Action forecasting performance for using only the features from previous three actions on Charades.

| Loss            | Accuracy (%) | mAP (%) |
|-----------------|--------------|---------|
| Cross-entropy   | 3.34         | 1.7     |
| $L_{opt} + $Cross-entropy | 6.25 | 2.3     |

**Table 5.** Comparison of action forecasting methods using Breakfast dataset only using features.

| Observation (%) | 10% | 20% | 30% | 50% | 10% | 20% | 30% | 50% |
|-----------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Cross-entropy   | 16.60 | 14.95 | 13.87 | 11.82 | 21.10 | 18.18 | 17.46 | 16.30 |
| Grammar [1]     | 16.32 | 15.01 | 14.91 | 13.25 | 19.38 | 18.64 | 17.97 | 16.97 |
| Nearest Neighbor [1] | 18.41 | 17.20 | 15.93 | 15.01 | 21.04 | 20.02 | 19.73 | 19.21 |
| RNN [1]         | 17.60 | 16.35 | 15.37 | 14.54 | 22.92 | 20.12 | 19.69 | 18.76 |
| OUR - w/o Attention | 23.38 | 21.07 | 18.72 | 17.20 | 24.91 | 22.03 | 20.41 | 20.29 |
| OUR             | 23.74 | 22.98 | 22.23 | 21.05 | 25.47 | 25.02 | 23.92 | 23.71 |
| Improvement     | +5.03 | +5.18 | +5.29 | +5.14 | +5.03 | +5.10 | +5.16 | +5.16 |

Figure 4. Illustration of ground truth and forecasted actions for some random videos. Each color represents an action.
References

[1] Yazan Abu Farha, Alexander Richard, and Juergen Gall. When will you do what?-anticipating temporal occurrences of activities. In CVPR, pages 5343–5352, 2018. 1, 2, 3, 8
[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014. 3
[3] Syed Zahir Bokhari and Kris M Kitani. Long-term activity forecasting using first-person vision. In ACCV, pages 346–360. Springer, 2016. 2
[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In CVPR, pages 6299–6308, 2017. 5
[5] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In AISTAT, pages 2681–2690, 2019. 5
[6] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Forecasting future action sequences with neural memory networks. In BMVC, 2019. 2, 3, 8
[7] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In ICRA, pages 3118–3125. IEEE, 2016. 2
[8] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In ECCV, pages 201–214. Springer, 2012. 1
[9] Yu Kong, Shangqian Gao, Bin Sun, and Yun Fu. Action prediction from videos via memorizing hard-to-predict samples. In AAAI, 2018. 1
[10] Yu Kong, Dmitry Kit, and Yun Fu. A discriminative model with multiple temporal scales for action prediction. In ECCV, pages 596–611. Springer, 2014. 2
[11] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. TPAMI, 38(1):14–29, 2015. 1
[12] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. 5
[13] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In ECCV, pages 689–704. Springer, 2014. 1
[14] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in lstms for activity detection and early detection. In CVPR, pages 1942–1950, 2016. 2
[15] Tahmida Mahmud, Mahmudul Hasan, and Amit K Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In ICCV, pages 5773–5782, 2017. 1, 2
[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In ACL, pages 311–318. Association for Computational Linguistics, 2002. 5
[17] Gabriel Peyr and Marco Cuturi. Computational optimal transport. Foundations and Trends in Machine Learning, 11 (5–6):355–602, 2019. 5
[18] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. Predicting human activities using stochastic grammar. In ICCV, pages 1164–1172, 2017. 2
[19] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In ICCV, pages 3696–3705, 2017. 2
[20] Cristian Rodriguez, Basura Fernando, and Hongdong Li. Action anticipation by predicting future dynamic images. In ECCV, pages 0–0, 2018. 2
[21] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In CVPR, pages 1194–1201. IEEE, 2012. 5
[22] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In ICCV, pages 1036–1043. IEEE, 2011. 2
[23] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Encouraging lstms to anticipate actions early. In ECCV, pages 280–289, 2017. 1, 2
[24] Yuge Shi, Basura Fernando, and Richard Hartley. Action anticipation with rbf kernelized feature mapping rnn. In ECCV, pages 301–317, 2018. 2
[25] Gunnar A Sigurdsson, Güll Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In ECCV, pages 510–526. Springer, 2016. 5
[26] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In ICML, pages 843–852, 2015. 3
[27] Thomas Suddendorf and Michael C Corballis. The evolution of foresight: What is mental time travel, and is it unique to humans? Behavioral and brain sciences, 30(3):299–313, 2007. 1
[28] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In NIPS, pages 3104–3112, 2014. 3
[29] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence - video to text. In ICCV, December 2015. 3
[30] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In CVPR, pages 98–106, 2016. 2
[31] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip. Predmn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In NIPS, pages 879–888, 2017. 2
[32] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In CVPR, pages 4584–4593, 2016. 3
[33] Kuo-Hao Zeng, Shih-Han Chou, Fu-Hsiang Chan, Juan Carlos Niebles, and Min Sun. Agent-centric risk assessment: Accident anticipation and risky region localization. In CVPR, pages 2222–2230, 2017. 1
[34] Kuo-Hao Zeng, William B Shen, De-An Huang, Min Sun, and Juan Carlos Niebles. Visual forecasting by imitating dy-
namics in natural sequences. In *ICCV*, pages 2999–3008, 2017.