Long-term Action Forecasting Using Multi-headed Attention-based Variational Recurrent Neural Networks

Siyuan Brandon Loh[1], Debaditya Roy[1], and Basura Fernando[1,2]
[1]SCC, IHPC, A*STAR, Singapore.
[2]CFAR, IHPC, A*STAR, Singapore.

Abstract

Systems developed for predicting both the action and the amount of time someone might take to perform that action need to be aware of the inherent uncertainty in what humans do. Here, we present a novel hybrid generative model for action anticipation that attempts to capture the uncertainty in human actions. Our model uses a multi-headed attention-based variational generative model for action prediction (MAVAP), and Gaussian log-likelihood maximization to predict the corresponding action's duration. During training, we optimise three losses: a variational loss, a negative log-likelihood loss, and a discriminative cross-entropy loss. We evaluate our model on benchmark datasets (i.e., Breakfast and 50Salads) for action forecasting tasks and demonstrate improvements over prior methods using both ground truth observations and predicted features from an action segmentation network (i.e., MS-TCN++). We also show that factorizing the latent space across multiple Gaussian heads predicts better plausible future action sequences compared to a single Gaussian.

1. Introduction

The fundamental challenge behind reasoning about human actions is the inherent uncertainty behind what humans do. Sources of uncertainty that accompany human actions are nontrivial and manifold, even for the most straightforward tasks, such as predicting what someone might do next after observing her take a cup. There are variations in the possible action that follows an observed action sequence, variations in how people perform a particular action sequence, and variations in the time that people require to complete the same action, to name a few. Therefore, systems designed for fundamentally humanistic endeavours should adopt solutions capable of modelling the probabilistic nature of human actions.

In the current paper, we focus on the problem of modelling the temporal dynamicity of human actions, with the goal of predicting the plausible action, or action sequence, that follows. This task, commonly known as action anticipation, has gained significant traction from the research community in recent years, mainly due to the development of high-quality datasets along with their associated challenges [5, 6] and its practical implications on human-robot collaboration.

Our approach towards tackling the action anticipation problem draws inspiration from deep generative models [26], a class of neural networks that have shown promise in approximating complex and high-dimensional probability distributions. Recent work has further demonstrated their success in capturing crucial aspects of human reasoning [17], such as language understanding [10], intention and goal inference from actions [11, 12, 30], and even emotion inference from observed expressions [22]. Specifically, we build upon the variational recurrent neural network (VRNN) [4], a time-series deep generative model, and propose novel modifications catered to tackle the uncertainty in human actions.

Besides predicting the action (sequence) that accompanies an observed action sequence, another crucial aspect of the action anticipation task is to predict the duration of the corresponding action. Current approaches often treat both action and duration prediction as independent of one another [7, 19]. However, it is straightforward to suggest that action and duration are fundamentally related constructs. We capture this idea by imposing conditional dependencies between the duration and action prediction. Specifically, we condition the duration prediction model on both the predicted action and the variational distribution that generates the predicted action.

In summary, the contributions of this paper include:

- a novel multi-headed attention-based variational recurrent neural network architecture for action anticipation.
- a generative model that specifies the conditional dependencies between action, the variational distribution that generates the action, and its corresponding duration.

2. Related Works

In the current section, we draw the reader's attention to recent ideas and methods related to the action sequence forecasting task. This task is a subset of the action anticipation problem and involves predicting the future action sequence that follows an observed action sequence.

A literature review suggests two overarching approaches toward tackling the action sequence forecasting task. On the one hand, some researchers use neural networks to model the future action sequence as a complex, non-linear, and deterministic function of the observed action sequence. [1] proposed a two-step approach to tackle the action sequence forecasting task. In the first step, an RNN-HMM network is used to infer the action sequence from the frames of the observed video segment. The action sequence is subsequently fed into either a convolutional or recurrent neural network that predicts the future action sequence. In contrast, [9] argued that both frame- and annotation-level features contain unique information that would facilitate the prediction of the future action sequence; they modelled the future action sequence as a function of both the frames from the observed video segment and their corresponding action label. [21] proposed adapting sequence-to-sequence models to action sequence forecasting tasks. Unlike the previous two approaches, their model only takes RGB frames of the observed video segment as input. Despite not having access to the action labels of the observed video segment, [21] empirically demonstrated how a GRU-based encoder-decoder architecture, trained by minimising both an optimal transport loss and a modified cross-entropy loss, can learn the complex mapping between the observed RGB frames and the unseen sequence of future action labels. Finally, [27] proposed a novel method of summarising information from the observed sequence. Instead of using temporal networks, their method involves aggregating both recent and long-term temporal history using non-local blocks [29]. Experimental results demonstrate that while long-term aggregation sometimes play a part in anticipation, recent actions are more informative in determining the immediate future.

In contrast to the aforementioned deterministic approaches, some researchers choose to model the future action sequence as a probabilistic function of the observed action sequence. [7] proposed an uncertainty-aware anticipation model that involves sampling a future action from a softmax distribution with parameters learned from a deterministic recurrent neural network. The subsequent actions are recursively predicted one at a time, with the previously predicted actions providing estimates for the to-be-predicted action. Instead of simply learning a probability distribution of the future action, [19] attempted to model the generative process governing both the action sequence and their inter-arrival times by combining Variational Auto-Encoder (VAE) with temporal point process models. At each timestep during training, their Action Point Process VAE learns a latent distribution conditioned on the past and current actions. The conditional latent distribution is sampled at inference time to generate future actions. Finally, [23] proposed a differentiable context-free grammar that is trained in an adversarial manner to learn the stochastic production rules from the distribution of the training data. The ability to choose multiple production rules facilitated the generation of multiple plausible future action sequences during inference. Overall, the probabilistic methods that we have described above reflect the fundamental notion of uncertainty in humanistic endeavours that we have previously alluded to in the opening paragraph. On the one hand, action prediction is often a one-to-many prediction problem, with more than one plausible action following an observed action sequence. On the other hand, duration is a random variable that varies within and across actions. Complex actions such as *frying an egg* often require more time than simpler ones such as *taking a cup*, and there are also variations in the amount of time people require to complete a particular action. These considerations led us to adopt a stochastic approach to the action sequence forecasting task. Specifically, our approach parallels that of [19] - we assume that variations in observed action and duration arise from a latent random variable. Our goal is to learn the distribution of both actions and duration through a generative model.

3. Forecasting actions using a generative model

3.1. Problem

A video of a human performing an activity is given. Our model observes an initial segment of the video and is tasked to forecast the following action sequence. Formally, we denote the actions observed in the initial part of the video as $a_{1:n} = (a_1, \cdots, a_n)$ with duration $d_{1:n} = (d_1, \cdots, d_n)$, with n referring to the number of unique action labels in the observed video segment, and d_i refers to the number of consecutive frames annotated with a_i. Here, we normalise d_i using the mean μ_d and standard deviation σ_d of all action duration in the training set, such that

$$d_i = \frac{d_i - \mu_d}{\sigma_d} \forall i \in \{1, \cdots, n\}. \tag{1}$$

Finally, the future unseen ground truth action sequence is denoted as $a_{n+1:N}$ with duration $d_{n+1:N}$, with N referring to the total number of unique action labels in the video.

3.2. Model Overview

During training, our model learns parameters of a latent distribution generating $\mathbf{a}_{1:n}$ and $\mathbf{d}_{1:n}$ via a **multi-headed attention-based variational recurrent neural network**

During inference, the action decoder computes the conditional probability of the next action given the latent variable, while the duration decoder computes the conditional

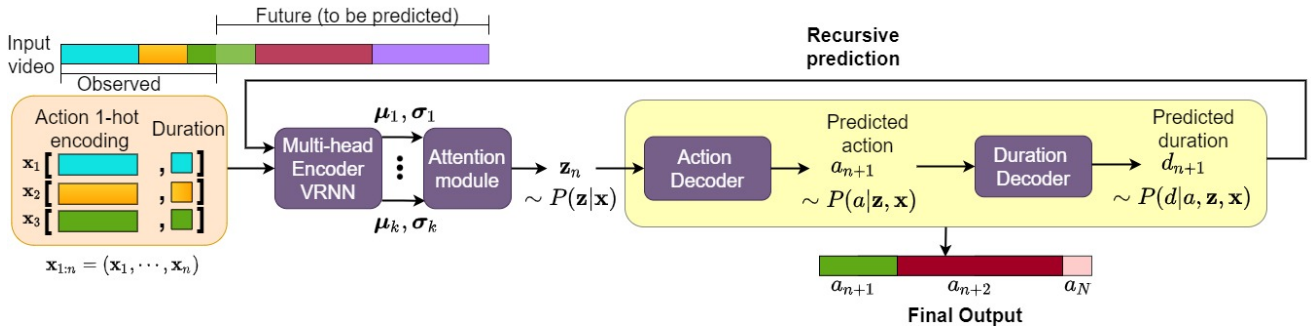


Figure 1. Model Overview. The multi-headed attention-based variational recurrent neural network encoder generates (sample from latent distribution) the latent variable $\mathbf{z}_n$. Then action decoder is conditioned on both latent variable $\mathbf{z}_n$ and the recurrent hidden state $\mathbf{h}_n$ to obtain context $\mathbf{c}_n$ to generate the next action. Duration generator predicts the parameters of the duration distribution, conditioned on the predicted action and the context $\mathbf{c}_n$. Then we sample the duration for each future action from the duration distribution.

probability of the duration given both the latent variable and the predicted action. This method is applied recursively to predict $a_{n+1:N}$ and $d_{n+1:N}$ (Figure 1).

3.3. Modelling the Observed Action Sequence

We use a multi-headed attention-based variational recurrent neural networkto model the variability in the observed action sequence. Similar to [4], we introduce a series of time-step wise latent random variables to model the observed sequence. However, the conditional prior distribution is no longer a multivariate Gaussian distribution, but a weighted sum of K multivariate Gaussians with means $\{\boldsymbol{\mu}^1_{prior}, ..., \boldsymbol{\mu}^k_{prior}\}$ and diagonal covariance $\{\boldsymbol{\sigma}^1_{prior}, \cdots, \boldsymbol{\sigma}^k_{prior}\}$, where K refers to the number of attention heads. Notably, the parameters of each Gaussian head are estimated through independent neural network ϕ^k_{prior} functions of the state variable $\mathbf{h}_{n-1}$ of an RNN:

$$\boldsymbol{\mu}^k_{prior} = \phi^{\mu_k}_{prior}(\mathbf{h}_{n-1}) \tag{2}$$

$$\boldsymbol{\sigma}^k_{prior} = \texttt{softplus}(\phi^{\sigma_k}_{prior}(\mathbf{h}_{n-1})) \tag{3}$$

From these estimated K Gaussian heads, we sample the latent variable $\mathbf{z}_{n-1}$ from the prior distribution q as follows:

$$q(\mathbf{z}_{n-1}|\mathbf{x}_{1:n-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{prior}, \boldsymbol{\sigma}_{prior}), \tag{4}$$

$$\boldsymbol{\mu}_{prior} = \sum_{k=1}^{K} \gamma^k_{prior} \times \boldsymbol{\mu}^k_{prior} \tag{5}$$

$$\boldsymbol{\sigma}_{prior} = \sum_{k=1}^{K} \gamma^k_{prior} \times \boldsymbol{\sigma}^k_{prior}, \tag{6}$$

where γ^k_{prior} is the learned attention weight on the k-th Gaussian head.

The latent variable $\mathbf{z}_n$ in equation 4 encodes the variability in the temporal structure of the action sequence prior to the incoming action. The temporal structure for every incoming action from the observed part of the video is also captured using the posterior distribution p, which is also a Gaussian distribution implemented via K dedicated Gaussian heads:

$$(\boldsymbol{\mu}^k_{pos}, \boldsymbol{\sigma}^k_{pos}) = \phi^k_{enc}([\phi_{\mathbf{a}}(\mathbf{a}_n), \phi_{\mathbf{d}}(\mathbf{d}_n), \mathbf{h}_{n-1}]). \tag{7}$$

The encoder network takes in the hidden state of the RNN and an embedding of the n^{th} action $\mathbf{a}_n$ and its corresponding duration $\mathbf{d}_n$ using feature extractor networks $\phi_{\mathbf{a}}$ and $\phi_{\mathbf{d}}$, respectively. The parameters of posterior distribution are obtained in a manner similar to that of the prior distribution(eq. 5 and 6), but with different set of learned attention weights γ^k_{pos} for $k \in \{1, \cdots, K\}$:

$$p(\mathbf{z}_n|\mathbf{a}_{1:n}, \mathbf{d}_{1:n}) \sim \mathcal{N}(\boldsymbol{\mu}_{pos}, \boldsymbol{\sigma}_{pos}). \tag{8}$$

We sample a latent variable $\mathbf{z}_n$ from the posterior distribution p using reparameterization trick [14]:

$$\mathbf{z}_n = \boldsymbol{\mu}_{pos} + \boldsymbol{\sigma}_{pos} \odot \boldsymbol{\epsilon}, \tag{9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ is a multivariate standard Gaussian distribution.

The hidden state of the Encoder RNN is subsequently updated as

$$\mathbf{h}_n = RNN_{enc}(\mathbf{h}_{n-1}, [\phi_{\mathbf{a}}(\mathbf{a}_n), \phi_{\mathbf{d}}(\mathbf{d}_n), \phi_{\mathbf{z}}(\mathbf{z}_n)]), \tag{10}$$

where $\phi_{\mathbf{z}}$ is a feature extractor over the latent variable.

3.4. Prediction

Our **Multi-headed Attention-based Variational Action Prediction (MAVAP)** follows a three-step approach. First, we compute a context vector $\mathbf{c}_{n+1}$ by concatenating the latent variable $\mathbf{z}_n$ with the state variable $\mathbf{h}_n$ of the encoder RNN:

$$\mathbf{c}_{n+1} = \phi_{con}([\phi_{\mathbf{z}}(\mathbf{z}_n), \mathbf{h}_n]), \tag{11}$$

where $\mathbf{c}_{n+1}$ is a multi-layer perceptron.

Next, we use our action decoder ϕ_{act}, a multi-layer perceptron, to compute a score vector $\mathbf{s}_{n+1}$ from the previously obtained context vector $\mathbf{c}_{n+1}$, before applying a softmax function to obtain the next action $\hat{a}_{n+1}$:

$$\mathbf{s}_{n+1} = \phi_{act}(\mathbf{c}_{n+1}) \tag{12}$$

$$\hat{a}_{n+1} = softmax(\mathbf{s}_{n+1}), \tag{13}$$

where ϕ_{act} is the action decoder network.

Finally, we use our duration decoder ϕ_{dur}, a multi-layer perceptron, compute the parameters of the duration distribution r:

$$r(d_{n+1}|\mathbf{a}_{n+1}, \mathbf{c}_{n+1}) \sim \mathcal{N}(\mu_{dur}, \sigma_{dur}), \tag{14}$$

$$\text{where } (\mu_{dur}, \sigma_{dur}) = \phi_{dur}([\mathbf{c}_{n+1}, \mathbf{s}_{n+1}]) \tag{15}$$

We predict both the action and duration sequences by recursively updating the hidden state of the encoder RNN using the predicted action and duration:

$$\mathbf{h}_{n+2} = RNN_{dec}(\mathbf{h}_{n+1}, [\phi_{\mathbf{a}}(\mathbf{a}_n), \phi_{\mathbf{d}}(\mathbf{d}_n), \phi_{\mathbf{z}}(\mathbf{z}_{n+1})]), \tag{16}$$

where $\mathbf{x}_{n+1}$ is obtained from predicted $\mathbf{a}_{n+1}$ the one-hot encoding of predicted action a_{n+1} and predicted duration d_{n+1}.

3.5. Training

Following the protocol in [7], we generate $N-1$ training examples from a single training video with N actions. For each training example, the final action serves as the target for the generator model, while the first $N-1$ actions serve as the observed sequence. This sampling strategy allows us to learn the temporal dependence between all possible observed sequences and predicted actions. Such diversity helps learn a more varied latent posterior distribution p and duration generator distribution r.

We train the network by minimising three loss functions. Firstly, we minimise is the KL-divergence between the conditional prior and posterior distributions for every observed action:

$$\mathcal{L}_{kld} = \sum_n KL(p(\mathbf{z}_n^{pos}|\mathbf{a}_{1:n}, \mathbf{d}_{1:n})||q(\mathbf{z}_n|\mathbf{a}_{1:n}, \mathbf{d}_{1:n})). \tag{17}$$

Next, we minimise the cross-entropy loss between the target and predicted action:

$$\mathcal{L}_{act} = -\sum \mathbf{a}_{n+1} \odot \log(\hat{\mathbf{a}}_{n+1}), \tag{18}$$

where $\mathbf{a}_{n+1}$ is ground truth one-hot label for the predicted action.

Finally, we minimise the Gaussian negative log-likelihood (GaussianNLL) loss on the parameters of the duration distribution:

$$\mathcal{L}_{dur} = \log(\sigma_{dur}) + \frac{(d_{n+1} - \mu_{dur})^2}{2\sigma_{dur}^2}. \tag{19}$$

It is worth noting that minimizing the GaussianNLL loss is equivalent to maximizing the likelihood of the duration being generated by the duration distribution.

These loss functions are used in a multi-task manner; the total loss is the sum of all three losses.

4. Experiments

4.1. Datasets and Implementation details

We evaluate our model on both the Breakfast [16] and 50Salads [28] datasets.

The *50 Salads* [28] 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 performed 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, with 20 action instances per video. The published dataset provided five splits, and all the results presented here are averaged over the five splits.

The *Breakfast* [16] 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. With an average duration of 2.3 minutes, videos on the Breakfast dataset are comparably shorter than that of the 50 Salads dataset. There are 48 fine-grained action classes, with an average of 6 action instances per video. All the results presented here are averaged over the four predefined splits dataset [16].

The models are implemented on PyTorch and trained for 20 epochs using the Adam optimizer with a learning rate of 0.0001 and batch size set to 1. We set the dimensions of the latent and hidden states to 64 for both datasets.

In our network, ϕ_{prior} and ϕ_{enc} are implemented as two-layered neural network with ReLU activation, $\phi_{\mathbf{x}}$, $\phi_{\mathbf{z}}$, and ϕ_{con} are implemented as a linear layer with ReLU activation. Finally, ϕ_{act} and ϕ_{dur} are linear functions over their respective inputs.

4.2. Model Evaluation

We evaluate our proposed framework with three conventional approaches to the action anticipation task. We evaluate our model's ability to forecast future action sequences in the first approach. Here, we follow the protocol in [1] - our model observes 10% or 20% of the video, with the goal of predicting the subsequent 10%, 20%, 30%, and 50 %.

Observation	20%				30%			
Prediction	10%	20%	30%	50%	10%	20%	30%	50%
Breakfast								
RNN [1]	60.4	50.4	45.3	40.4	61.5	50.3	45.0	41.8
CNN [1]	58.0	49.1	44.0	39.3	60.3	50.1	45.2	40.5
Time Cond. [13]	64.5	**56.3**	**50.2**	**44.0**	66.0	55.9	**49.1**	**44.2**
Temp. Agg. [27]	65.5	55.5	46.8	40.1	67.4	56.1	47.4	41.5
Unc. Awa. [7]	53	44.1	39.7	34.9	53.9	44.5	40.2	35.5
MAVAP (Ours)	**69.1**	54.1	45.4	35.1	**70.9**	**56.2**	47.6	38.1
50 Salads								
RNN [1]	42.3	31.2	25.2	16.8	44.2	29.5	20.0	10.4
CNN [1]	36.1	27.6	21.4	15.5	37.4	24.8	20.8	14.1
Time Cond. [13]	45.1	33.2	27.6	17.3	**46.4**	**34.8**	25.2	13.8
Temp. Agg. [27]	**47.2**	34.6	**30.5**	**19.1**	44.8	32.7	23.5	15.3
Unc. Awa. [7]	38.1	30.1	26.3	16.5	40.0	29.2	23.7	**15.5**
MAVAP (Ours)	43.3	**35.4**	28.4	17.4	43.8	31.8	**27.2**	14.2

Table 1. Comparison of action sequence forecasting on Breakfast and 50Salads with state-of-the-art. All methods take as input the ground-truth observed action sequences.

Mean over classes (MoC) accuracy is computed for each configuration. Next, we evaluate our model on an action sequence prediction task. Here, the model observes 1 or 2 actions, with the goal of predicting the remaining actions. Finally, we evaluate our model on the next action prediction task. We report top-k accuracy for action sequence prediction, and next-action prediction.

4.3. Action Sequence Forecasting with Ground-Truth Annotations

We first evaluate our proposed method's ability to forecast future action sequences by providing our model with ground-truth annotations of the observed video segment. Ground-truth annotations are consistent and without errors, ensuring fair comparison across different algorithms. For each example in the test set, we generate 50 samples and use the mode of the predicted distribution to compute the Mean over Classes (MoC) accuracy.

As shown in Table 1, the performance of our proposed method is comparable with, but does not outperform the current state-of-the-art (i.e., [27]). One possible reason for this might pertain to the unsuitability of the Gaussian distribution for action forecasting tasks.

Nonetheless, we managed to observe consistent improvements over the stochastic model recently proposed in [7] on both datasets and across all configurations. Comparing both modelling approaches illuminates the potential reasons for this performance improvement. Notably, both approaches differ in the specification of the relationship between action and duration. In our approach, we impose conditional dependencies between action and duration to capture the idea that duration is a random variable that fundamentally depends on the action variable. In contrast, [7] models both duration and action as independent of one another. Both approaches also differ in the parameterization of the action and duration probability distributions. In [7], the action and duration distributions are parameterized simply as non-linear functions of the observed video segments.

Observation	20%				30%			
Prediction	10%	20%	30%	50%	10%	20%	30%	50%
Breakfast								
RNN [1]	18.1	17.2	15.9	15.8	21.6	20.0	19.7	19.2
CNN [1]	17.9	16.4	15.4	14.5	22.4	20.1	19.7	18.8
Time Cond. [13]	18.4	17.2	16.4	15.8	22.8	20.4	19.6	19.8
Unc. Awa. [7]	28.9	28.4	27.6	28.0	32.4	31.6	32.8	30.8
Cycle Cons. [8]	25.9	23.4	22.4	21.5	29.7	27.4	25.6	25.2
Temp. Agg. [27]	37.1	31.8	30.1	27.1	39.8	34.2	31.9	27.8
Attention [21]	23.0	22.3	22.0	20.9	26.5	25.0	24.1	23.6
MAVAP w/ MS-TCN++	**69.1**	**52.6**	**43.5**	**32.9**	**68.0**	**53.7**	**44.6**	**35.5**
50 Salads								
RNN [1]	30.6	25.4	18.7	13.5	30.8	17.2	14.8	9.8
CNN [1]	21.2	19.0	16.0	9.9	29.1	20.1	17.5	10.9
Time Cond. [13]	32.5	27.6	21.3	16.0	35.1	27.0	22.0	15.6
Unc. Awa. [7]	24.9	22.4	19.9	12.8	29.1	20.5	15.3	12.3
Cycle Cons. [8]	34.7	28.4	21.8	15.2	34.4	23.7	18.9	15.9
Temp. Agg. [27]	34.7	25.9	23.7	15.7	34.5	26.1	19.0	15.5
Adv. Gram. [23]	39.5	33.2	25.9	21.2	39.5	31.5	26.4	19.8
Attention [21]	39.3	31.4	27.0	**23.9**	41.7	**32.7**	**31.4**	**26.4**
MAVAP w/ MS-TCN++	**43.5**	**32.4**	**27.5**	16.1	**42.5**	30.8	25.4	14.8

Table 2. Comparison of action sequence forecasting on Breakfast and 50Salads with state-of-the-art. All methods use features generated by action segmentation networks to predict future actions.

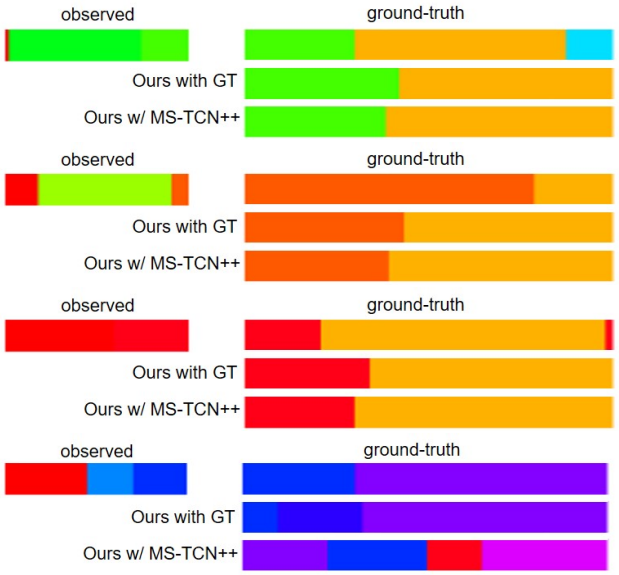
In contrast, we adopt a generative approach that is similar to that in [19]. Specifically, we use a latent distribution, parameterized as a non-linear function of the observed video segments, to generate the parameters of action and duration distributions. Our strategy to impose conditional dependencies amongst actions and duration, and the expressivity conferred by the inclusion of the latent variable allows us to better model the uncertainty in human action sequences than previous stochastic approaches.

4.4. Action Sequence Forecasting with Predicted Features

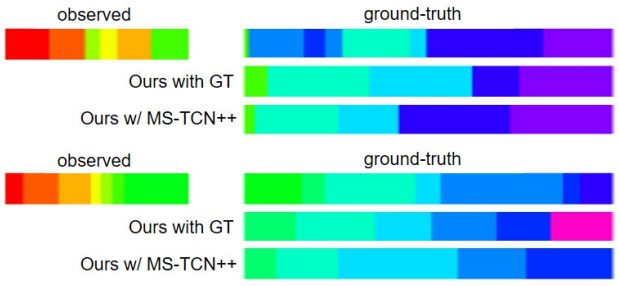
Here, we test the robustness of our current model by feeding it features predicted from the observed video segment by an action segmentation network, MS-TCN++ [18]. As shown in Table 2, we managed to achieve superior performance over existing approaches across both datasets and overall configurations when frame-level MS-TCN++ feature scores of the observed video segment were fed as input to our model. We acknowledge that this vast improvement over existing approaches might result from the action segmentation network that we deploy in the current work. Nonetheless, these results demonstrate our model's capabilities in adapting to noising input signals. They also attest to our model's flexibility in leveraging recent developments in action recognition and action segmentation methods for action forecasting tasks.

Observation	20%				30%				average
Prediction	10%	20%	30%	50%	10%	20%	30%	50%	
# Gauss	**Breakfast**								
1	69.1	53.7	44.9	34.8	70.9	55.9	47.2	**38.1**	51.8
3	**69.2**	**54.2**	45.1	35.0	**71.0**	**56.2**	47.5	**38.1**	**52.1**
5	69.1	53.9	44.8	34.3	**71.0**	56.1	47.2	37.5	51.7
8	69.1	53.9	44.8	34.5	70.9	55.9	47.0	37.5	51.7
10	69.1	53.9	**45.6**	35.0	70.9	56.1	**47.7**	38.0	52.0
12	69.1	54.1	45.4	**35.1**	70.9	**56.2**	47.6	**38.1**	**52.1**
15	68.9	52.8	43.8	33.6	70.8	55.3	46.3	37.0	51.1
# Gauss	**50 Salads**								
1	42.1	30.9	23.7	16.5	42.2	28.4	23.1	13.8	27.6
3	42.0	33.0	25.7	**18.0**	42.7	29.7	24.5	**14.7**	28.8
5	**45.2**	34.2	26.9	16.3	42.7	30.4	25.4	13.1	29.3
8	44.6	33.3	27.7	16.8	**44.0**	31.4	26.1	14.5	29.8
10	43.9	34.0	27.6	17.8	43.6	31.4	27.2	14.5	30.0
12	43.4	**35.4**	**28.4**	17.4	43.8	**31.8**	**27.3**	14.2	**30.2**
15	43.1	32.1	27.0	17.6	42.6	31.0	25.8	14.4	29.2

Table 3. Ablation results on Breakfast and 50Salads with different number of Gaussian heads.

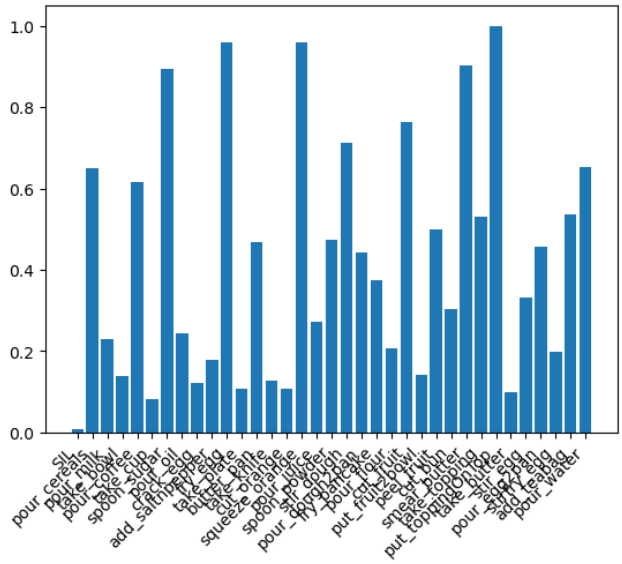


(a) Breakfast

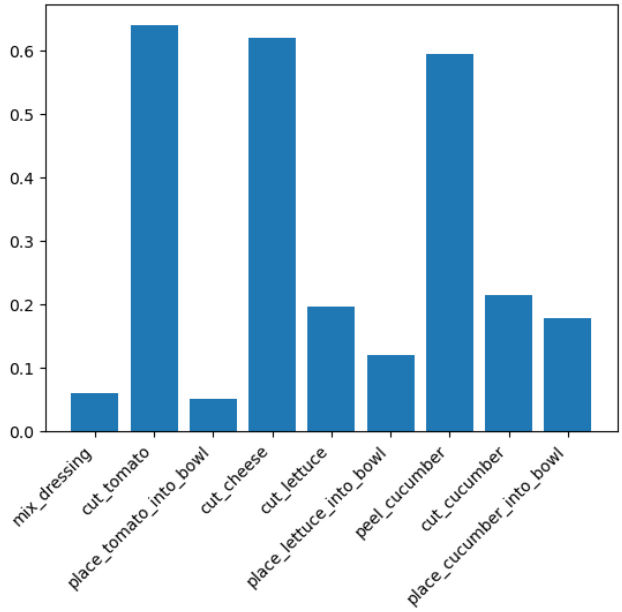


(b) 50 Salads

Figure 2. Qualitative results for action sequence forecasting (observe: 30%; predict: 50%) using ground-truth and MS-TCN++ scores on both Breakfast and 50Salads



(a) Breakfast



(b) 50 Salads

Figure 3. Per-class accuracy for action sequence forecasting on both Breakfast and 50Salads datasets.

4.5. Action Sequence Forecasting Qualitative Results

Both datasets have different modelling requirements. With an average of approximately six actions per sequence, the prediction accuracy on the Breakfast dataset is mainly dependent on the model's ability to predict the corresponding action's duration. In contrast, an action sequence in the 50Salads dataset contains approximately 20 actions. The

prediction accuracy on the 50Salads dataset would require a model sensitive to such variabilities.

Qualitative results presented in figure 2 demonstrate our model's ability to adapt to the requirements of each dataset. On the Breakfast dataset, our model can correctly estimate the remaining duration of the last observed action when it spills into the observed period. On the 50Salads dataset, our model predicts, with reasonable accuracy, the variability in a sequence of actions.

Per-class accuracy results on both Breakfast and 50Salads revealed surprising variations in performance across classes. On the Breakfast dataset, some action classes, like "spoon sugar", "fry egg", "stir milk", have almost near-perfect accuracy. In contrast, our model often fails to predict others classes, like "take cup", "take plate", "take knife". On the 50Salads dataset "cut cucumber", "peel cucumber", "cut cheese", were the best performing action. One possible explanation for these findings pertains to the qualitative nature of each action. For instance, actions such as "fry egg" have more utility than others like "add salt and pepper". This might lead our model to underestimate the predictions of action classes with lesser utility.

4.6. Effect of the number of Gaussian heads on Action Sequence Forecasting Performance

Multi-headed attention networks often permit a more nuanced state representation of the input sequence than single-head networks. Adapting multi-headed attention networks to latent space models designed for action anticipation tasks can potentially lead to a more pronounced factorization of the latent space, which is of greater effectiveness in modelling the variability in the input action and duration sequences.

However, our experimental findings revealed that the positive effect of multi-headed attention networks on forecasting performance depended on the dataset. As shown in Table 3, increasing the number of Gaussian heads did not improve the average forecasting performance on the Breakfast dataset. In contrast, we observe that increasing the number of Gaussian heads from 1 - to 12 improved average forecasting accuracy by 2.6%. This dataset-dependent effect is an interesting one that warrants further analysis. One potential reason for this might pertain to the length of the action sequence.

4.7. Action Sequence Prediction

In the current section, we evaluate the action sequence generation capability of various deterministic and stochastic action prediction architectures. Given the observed action(s), the model is tasked to predict the sequence of actions that might follow. Each architecture observes either 1 or 2 actions and predicts the rest of the actions.

A total of six encoder-decoder architectures were evaluated, namely 1) an RNN encoder with a linear layer decoder that is inspired by [1], 2) a GRU for both encoder and decoder, 3) an LSTM for both encoder and decoder, 4) a Deep Markov model (DMM) encoder [15] and a linear layer decoder, 5) a VRNN encoder with a linear layer decoder, and 6) our proposed model without the duration prediction module. We report both the Top-1 and Top-2 accuracy scores of each architecture in Table 4.

Model (Encoder-Decoder)	Observed actions	Accuracy	
		Top-1	Top-2
RNN-Linear	1	29.68	32.64
	2	22.63	29.41
GRU-GRU	1	22.75	25.32
	2	32.14	33.94
LSTM-LSTM	1	25.38	26.69
	2	33.08	34.91
DMM-Linear	1	19.34	22.45
	2	27.61	31.56
VRNN-Linear	1	23.45	26.87
	2	44.62	52.91
MAVAP (Ours)	1	27.65	37.24
	2	**51.03**	**63.87**

Table 4. Comparing different encoder-decoder models while predicting remaining actions in the sequence. Results are on the Breakfast dataset.

Results in Table 4 neatly highlight the positive relationship between network complexity and model performance. Notably, networks with rich internal state representation, such as those with gated architectures (i.e., GRU and LSTM), performed significantly better than those without (i.e., RNN-Linear, DMM). Furthermore, they highlight the utility of modelling the observed variability of action sequences via latent random variables - the best performing models are the VRNN-Linear network and our proposed model. Overall, the results of this experiment highlight the value of modelling temporal variability via a latent random variable and learning the temporal dependencies across actions.

4.8. Next Action Prediction

In this section, we evaluate the performance of our variational action prediction model on its ability to predict the class of the next action segment after observing one or more past actions (i.e., next action anticipation). Here, our method outperforms prior approaches regardless of whether predicted features or ground-truth annotations were used (Table 5). Once again, we observe the benefits of modelling the input sequence with attention-weighted Gaussian heads over those with a single Gaussian [7, 19].

Method (Features)	Accuracy
With Features	
Predictive+ Transitional (Resnet50) [20]	32.3
Temp. Agg.(I3D) [27]	47.0
MM-Transformer (I3D, Flow, Human-Obj.) [25]	48.4
MAVAP (Ours w/ MS-TCN++)	**65.9**
With Ground Truth	
Predictive+ Transitional [20]	43.0
Unc. Awa. [7]	57.8
APP-VAE [19]	62.2
Temp. Agg. [27]	64.7
MAVAP (Ours)	**70.1**

Table 5. Comparison of next action prediction on Breakfast dataset

5. Discussion & Conclusion

Models developed for reasoning about what we do have to contend with the probabilistic nature of human actions. Here, we build upon existing works on stochastic recurrent networks to present a hybrid generative model for action forecasting: actions are predicted using a multi-headed attention-based variational recurrent module, while the time taken to complete the action (i.e., duration) is predicted using a generative Gaussian likelihood maximisation. The model is trained end-to-end by optimising three loss functions, two generative losses and one discriminative loss.

Overall results of our model are promising and demonstrate the utility of our novel architecture. Our full model performed comparably with the state-of-the-art on two benchmark datasets on the action sequence forecasting task. Our variational action prediction model outperformed deterministic encoder-decoder models and stochastic deep Markov models on the action sequence prediction task. It also outperformed the state-of-the-art on the next action prediction task.

We intend to build on the current work by investigating the effects of parameterising the latent space with different probability distributions and the interaction between attention heads and action sequence length on forecasting performance. We also plan to build on this current work by further examining the semantics of latent space in variational frameworks. Specifically, we would like to investigate if we can use variational frameworks to model human Theory of Mind [24]. Indeed, a line of work in the domain of computational cognitive science has demonstrated the viability of latent space models in capturing how humans draw mental state inferences from observed actions [2, 3, 12]. However, these works have only been explored in constrained game-like environments. Thus, it would be interesting to explore whether such models can generalise to real-life situations.

Acknowledgment

This research/project is supported in part by the National Research Foundation, Singapore under its AI Singapore Program (Award Number: AISG-RP-2019-010).

References

[1] Yazan Abu Farha, Alexander Richard, and Juergen 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, 4, 5, 7

[2] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017. 8

[3] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009. 8

[4] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28:2980–2988, 2015. 1, 3

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1

[6] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2021. 1

[7] Yazan Abu Farha and Juergen Gall. Uncertainty-aware anticipation of activities. *arXiv preprint arXiv:1908.09540*, 2019. 1, 2, 4, 5, 7, 8

[8] Yazan Abu Farha, Qiuhong Ke, Bernt Schiele, and Juergen Gall. Long-term anticipation of activities with cycle consistency. *Pattern Recognition*, 12544:159, 2020. 5

[9] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Forecasting future action sequences with neural memory networks. In *Proceedings of the 30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 298. BMVA Press, 2019. 2

[10] Noah D Goodman and Michael C Frank. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829, 2016. 1

[11] Julian Jara-Ettinger. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019. 1

[12] Julian Jara-Ettinger, Laura E Schulz, and Joshua B Tenenbaum. The naive utility calculus as a unified, quantitative

framework for action understanding. *Cognitive Psychology*, 123:101334, 2020. 1, 8

[13] Qiuhong Ke, Mario Fritz, and Bernt 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. 5

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[15] Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 7

[16] Hilde Kuehne, Ali Arslan, and Thomas 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. 4

[17] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 1

[18] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 5

[19] Nazanin Mehrasa, Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. A variational auto-encoder model for stochastic point processes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3165–3174, 2019. 1, 2, 5, 7, 8

[20] Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 8

[21] Yan Bin Ng and Basura Fernando. Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting. *IEEE Transactions on Image Processing*, 29:8880–8891, 2020. 2, 5

[22] Desmond C Ong, Zhengxuan Wu, Zhi-Xuan Tan, Marianne Reddan, Isabella Kahhale, Alison Mattek, and Jamil Zaki. Modeling emotion in complex stories: the stanford emotional narratives dataset. *IEEE Transactions on Affective Computing*, 12(3):579–594, 2019. 1

[23] AJ Piergiovanni, Anelia Angelova, Alexander Toshev, and Michael S Ryoo. Adversarial generative grammars for human activity prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020. 2, 5

[24] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978. 8

[25] Debaditya Roy and Basura Fernando. Action anticipation using pairwise human-object interactions and transformers. *IEEE Transactions on Image Processing*, 30:8116–8129, 2021. 8

[26] Lars Ruthotto and Eldad Haber. An introduction to deep generative modeling. *GAMM-Mitteilungen*, 44(2):e202100008, 2021. 1

[27] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *Proceedings of European Conference on Computer Vision*, pages 154–171. Springer, 2020. 2, 5, 8

[28] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM International joint Conference on Pervasive and ubiquitous computing*, pages 729–738, 2013. 4

[29] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 2

[30] Tan Zhi-Xuan, Jordyn Mann, Tom Silver, Josh Tenenbaum, and Vikash Mansinghka. Online bayesian goal inference for boundedly rational planning agents. *Advances in Neural Information Processing Systems*, 33:19238–19250, 2020. 1