2022

THE PERCEPTION-ACTION LOOP IN ATTENTION-BASED PREDICTIVE AGENTS: APPLICATION TO MULTIMODAL DATA GENERATION AND RECOGNITION

Murchana Baruah

THE PERCEPTION-ACTION LOOP IN ATTENTION-BASED PREDICTIVE AGENTS: APPLICATION TO MULTIMODAL DATA GENERATION AND RECOGNITION

by

Murchana Baruah

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Major: Electrical and Computer Engineering

The University of Memphis

March 2022

Acknowledgements

At first, I wish to express my sincere gratitude towards my advisor Dr. Bonny Banerjee for his continuous guidance during my research. His patience, immense knowledge, motivation and enthusiasm helped me throughout my research as well as in writing this dissertation. It has been a great privilege to work with and learn from him. He has been an amazing mentor and advisor, not only in research, but also in other aspects of my academic life.

In addition to my advisor, I sincerely thank my other dissertation committee members, Dr. Madhusudhanan Balasubramanian, Dr. Aaron L. Robinson and Dr. Deepak Venugopal for their time and invaluable advice and constructive feedback to improve this dissertation.

I would like to express my thankfulness to my fellow members in the Computational Intelligence Laboratory (CIL) for the thought provoking discussions and suggestions throughout my research.

I gratefully acknowledge the funding support for this work from the Institute for Intelligent Systems in the form of a graduate assistantship.

Lastly but most importantly, I would like to thank my family for extending their constant emotional and spiritual support.

Abstract

Baruah, Murchana. PhD. The University of Memphis. March 2022. The perception-action loop in attention-based predictive agents: Application to multimodal data generation and recognition. Major Professor: Dr. Bonny Banerjee.

With the proliferation of soft and hard sensors, data in multiple sensor modalities has become commonplace. In this dissertation, we propose a general-purpose agent model that operates using a closed perception-action loop. The agent actively and sequentially samples its environment, driven by sensory prediction error. It learns where and what to sample by minimizing this prediction error, without any reinforcement. This end-to-end model is evaluated on three applications: (1) generation and recognition of handwritten numerals and alphabets from images and videos, (2) generation and recognition of human-human interactions from videos, and (3) recognition of emotions from speech via generation. For each application, the model yields state-of-the-art accuracy on benchmark datasets, while also maintaining sample and model size efficiency.

In order to validate our model with respect to human performance, we collect mouse-click attention tracking (mcAT) data from 382 participants trying to recognize handwritten numerals and alphabets (upper and lowercase) from images via sequential sampling. Images from benchmark datasets are presented as stimuli. The collected data consists of a sequence of sample (click) locations, predicted class label(s) at each sampling, and duration of each sampling. We show that on average, participants observe only 12.8% of an image for recognition. When exposed to the same stimuli and experimental conditions as the participants, our agent model performs handwritten numeral/alphabet recognition more efficiently than the participants as well as a highly-cited attention-based reinforcement model.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

Chapter 1

Introduction

Perception and action are inextricably tied together as, in the real world, efficiency is as important as accuracy. Nature has evolved the visual system such that, to minimize resources, it learns to selectively attend to a few locations that provide information for the task at hand. Inspired from the human visual system, in this dissertation, we propose a predictive agent model, which observes its visual environment via a sequence of glimpses. It predicts, learns and acts by minimizing sensory prediction error in a closed loop. One challenge associated with such agent models is learning the action or where to look at in the environment at subsequent glimpses. This becomes more challenging for multimodal applications. Typically, this problem is addressed by reinforcement based models. In this aspect, we explore a model where action/attention is modeled as proprioception in a multimodal setting, and is guided by the perceptual prediction error, not by reinforcement. There are some points of similarity between our model and predictive coding.

We implement the model using a multimodal variational recurrent neural network. The agent is evaluated on three different kinds of data for generation and recognition tasks: (a) images of handwritten digits/alphabets, (b) videos of two person interactions and (c) emotional speech. We investigate the model for different configurations and show that our model is more efficient and the accuracy is comparable to the state-of-the-art.

In order to validate our model with respect to human performance, we collect mouse-click attention tracking (mcAT) data from 382 participants trying to recognize handwritten numerals and alphabets (upper and lowercase) from images via sequential sampling. We show that the proposed model is more efficient in handwritten numeral/alphabet recognition than human participants as well as a highly-cited attention-based reinforcement model. This dataset can be further utilized for evaluating the attention mechanism in attention based models for classifying handwritten numerals and digits.

1.1 Outline

This dissertation will proceed as follows:

In Chapter 2, we investigate the optimal modality selection problem for time-series data in the context of late fusion. Multimodal emotion or action recognition is used as a testbed. Widely-used features and classifier are used for each modality drawn from five benchmark datasets. We experimented with four widely-used late fusion methods. From the classification accuracies obtained for all possible combinations of modalities in each dataset, we observe that the accuracy does not always improve with increase in number of modalities. We further show that expected information gain increases monotonically with classification accuracy in an useful interval and hence, can be used for selecting a subset of modalities for late fusion to achieve a high classification accuracy.

In Chapter 3, we propose a general-purpose agent model consisting of proprioceptive and perceptual pathways. The agent actively samples its environment via a sequence of glimpses. It completes the partial propriocept and percept sequences observed till each sampling instant, and learns where and what to sample by minimizing prediction error, without reinforcement or supervision (class labels). The model is evaluated by exposing it to two kinds of stimuli: images of fully-formed handwritten numerals and alphabets, and videos of gradual formation of numerals. It yields state-of-the-art prediction accuracy upon sampling only 22.6% of the scene on average. The model saccades when exposed to images and tracks when exposed to videos. This is the first known attention-based agent to generate realistic handwriting with state-of-the-art accuracy and efficiency by interacting with and learning end-to-end from static and dynamic environments.

Multiple attention-based models that recognize objects via a sequence of glimpses have reported results on handwritten numeral recognition. However, no eye-tracking data for handwritten numeral or alphabet recognition is available. Availability of fixation data would allow attention-based models to be evaluated in comparison to human performance.

We collect mouse-click attention tracking (mcAT) data from 382 participants trying to recognize handwritten numerals and alphabets (upper and lowercase) from images via sequential sampling. Images from benchmark datasets are presented as stimuli. The collected data consists of a sequence of sample (click) locations, predicted class label(s) at each sampling, and the duration of each sampling. In Chapter 4, we present the details for this dataset and we show that on average, participants observe only 12.8% of an image for recognition. We propose a baseline model to predict the location and the class(es) a participant will select at the next sampling. When exposed to the same stimuli and experimental conditions as our participants, a highly-cited attention-based reinforcement model falls short of human efficiency.

A number of attention-based models for either classification or generation of handwritten numerals/alphabets have been reported in the literature. However, generation and classification are done jointly in very few end-to-end models. In Chapter 5, we extend the proposed model in Chapter 3 for classification such that the model learns where and what to sample by jointly minimizing the classification and generation errors. Three variants of this model are evaluated for handwriting generation and recognition on images of handwritten numerals and alphabets from benchmark datasets. We show that the proposed model is more efficient in handwritten numeral/alphabet recognition than human participants in a recently published study as well as a highly-cited attention-based reinforcement model. This is the first known attention-based agent to interact with and learn end-to-end from images for recognition via generation, with high degree of accuracy and efficiency.

In Chapter 6, we extend our model proposed in Chapter 3 to human interaction generation application. The model is exposed to videos of two-person interactions, where one person is the modeled agent and the other person's actions constitute its visual observation. For each interaction class, the model learns to selectively attend to locations in the other person's body. The proposed attention-based agent is the first of its kind to interact with and learn end-to-end from human interactions, and generate realistic interactions with performance comparable to models without attention and using significantly more computational resources.

In Chapter 7, we extend our model in Chapter 6 for human interaction classification. The model is exposed to videos of two-person interactions under two settings - (1) one person is the modeled agent and the other person's body movements constitute its visual observation, (2) a third person is the modeled agent and the two interacting persons body movements constitute its visual observation. Our model predicts both the interaction sequence and the interaction class for the two settings. Three variants of the proposed model, and three ways of implementing action selection (where to attend to) for each variant are analyzed using benchmark datasets. We show that classification accuracy is comparable when sampling locations are determined from sensory prediction error or from learned weights (without involving prediction error), but the latter is less efficient in terms of model size. This is the first known attention-based agent to interact with and learn end-to-end from two-person interaction environments for recognition via generation, with high degree of accuracy and efficiency.

In Chapter 8, we extend the proposed model in Chapter 7 to speech emotion recognition (SER) problem. The model uses MFCC as the input speech representation. Most SER models use a spectrogram as the input speech representation. Since the MFCC is of lower dimension than a spectrogram, the model is size- and data-efficient. At each instant, the model infers the emotion class and generates the next observation, computes the generation error, and selectively samples (attends to) the location of highest error. This simple and efficient model provides interesting design insights when evaluated for SER on the RAVDESS and IEMOCAP benchmark datasets.

In Chapter 9, we summarize the key findings from this dissertation.

1.2 List of Publications

Journal Papers:

1. **M. Baruah** and B. Banerjee (2022). An Attention-based Predictive Agent for Static and Dynamic Environments. *IEEE Access*.
2. **M. Baruah** and B. Banerjee. A Dataset for Handwritten Numeral and Alphabet Recognition via Sequential Sampling. *Under review*.
3. **M. Baruah** and B. Banerjee. Intent prediction in Human-Human Interactions. *Under review*.
4. **M. Baruah** and B. Banerjee. An Attention-based Predictive Agent for Handwritten Numeral/Alphabet Recognition via Generation. *In preparation*.
5. **M. Baruah** and B. Banerjee. An Attention-based Multimodal Predictive Agent: Exploiting Perception and Proprioception for Interaction Recognition via Generation. *In preparation*.

Conference/Workshop Papers:

6. **M. Baruah** and B. Banerjee (2020). The Perception-Action Loop in a Predictive Agent, *CogSci*, July 29- August 21.
7. **M. Baruah** and B. Banerjee (2020). A Multimodal Predictive Agent Model for Human Interaction Generation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, June 19.
8. **M. Baruah** and B. Banerjee (2020). Modality Selection for Classification on Time-series Data, *KDD Workshop on Mining and Learning from Time Series, MileTS*, August 24.
9. B. Banerjee, M. H. Kapourchali, **M. Baruah**, M. Deb, K. Sakauye, M. Olufsen (2021). Synthesizing Skeletal Motion and Physiological Signals as a Function of a Virtual Human's Actions and Emotions, *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, April 29-May 1.

10. **M. Baruah** and B. Banerjee. Speech Emotion Recognition via Generation using an Attention-based Variational Recurrent Neural Network. *Under review*.

Chapter 2

Modality Selection for Classification on Time-series Data

Abstract: In this paper, we investigate the optimal modality selection problem for time-series data in the context of late fusion. Multimodal emotion or action recognition is used as a testbed. Widely-used features and classifier are used for each modality drawn from five benchmark datasets. We experimented with four widely-used late fusion methods. From the classification accuracies obtained for all possible combinations of modalities in each dataset, we observe that the accuracy does not always improve with increase in number of modalities. We further show that expected information gain increases monotonically with classification accuracy in an useful interval and hence, can be used for selecting a subset of modalities for late fusion to achieve a high classification accuracy.

2.1 Introduction

Real-world time-series data is often multimodal. Learning from multiple modalities facilitates learning a richer representation which helps to make more accurate inference [Ngiam et al., 2011, Castellano et al., 2008]. Each modality is expected to provide unique information, otherwise it would be redundant. A challenge associated with multimodal data classification is to select a subset of modalities to maximize accuracy. This optimal modality selection problem is investigated in areas such as multimedia [Wu et al., 2004, Kankanhalli et al., 2006, Atrey et al., 2007] and wireless sensor networks [Pahalawatta et al., 2004, Lam et al., 2004, Isler and Bajcsy, 2005]. This problem has been explored for early fusion (see [Atrey et al., 2010] for review) but rarely for decision or late fusion.

In this paper, we investigate the optimal modality selection problem for time-series data in the context of late fusion. We experimentally evaluate the effect of combining different modalities on emotion recognition accuracy using two benchmark datasets (DEAP [Koelstra et al., 2012], HCI Tagging [Soleymani et al., 2012]) containing physiological signals, and on action classification accuracy using three benchmark datasets (PAMAP2 [Reiss and Stricker, 2012b], UTD MHAD [Chen et al., 2015b], Berkeley MHAD [Ofli et al., 2013]) containing inertial, motion capture and depth data. Several methods have been reported for action or emotion recognition using these benchmark datasets (see for example [Wiem and Lachiri, 2017, Wiem and Lachiri, 2016, Ofli et al., 2013, Busso et al., 2004, Chowdhury et al., 2018, Koelstra et al., 2012]). However, they did not report on how to select a subset of modalities, and their experiments are limited to a few datasets. Modality selection criteria, such as independence of classifiers [Kuncheva et al., 2000] and classifier correlation [Goebel et al., 2002], have been used in late fusion models, but were not evaluated on a number of benchmark datasets.

We compare emotion or action recognition accuracy from four late fusion methods and all possible subsets of modalities from each of the five benchmark datasets. We selected candidates for the different components of a late fusion model from the literature based on their wide usage: (1) features extracted from the signal in each modality, (2) classifier, (3) late fusion methods, and (4) modality selection criterion. These candidates allow us to experiment with multiple late fusion models and draw general conclusions.

The contributions of this paper are as follows:

(1) Our experimental results reveal that, contrary to expectation, emotion or action classification accuracy does not always increase with increase in number of modalities for different late fusion methods. More data might confuse a model as data is not always informative.

(2) We empirically show that information gain increases monotonically with classification accuracy in the accuracy interval $[a, 1] \times 100\%$ where $a \in [0, 0.5]$ for two or more classes.

(3) Our experimental results show that information gain is an useful metric for selecting the optimal subset of modalities for late fusion. For the five benchmark datasets

and both action and emotion recognition, subsets selected using information gain yield results comparable to the highest classification accuracy.

2.2 Models and Methods

Our recognition model consists of four functions: feature extraction, classification, modality selection, and late fusion (see Fig. 1.). We use widely-used feature extraction, classification and fusion methods, and modality selection metrics.

Let $\{M_1, \ldots, M_n\}$ be n modalities (or signals), and x_i be the feature vector and λ_i be the classifier for modality M_i ($i = 1, \ldots, n$). Let there be m classes, $\{\omega_1, \ldots, \omega_m\}$, such that $P(\omega_k|x_i)$ represent the posterior probability for class ω_k from classifier λ_i. [1]

Definition 2.2.1. ***Correlation degree.*** *The correlation-based classifier selection criteria for n classifiers is [Goebel et al., 2002, Niu et al., 2007]:*

$$\rho_n = \frac{nN^f}{N - N^f - N^r + nN^f} \tag{2.1}$$

where ρ_n is the correlation degree, N^f is the number of samples misclassified by all classifiers, N^r is the number of samples classified correctly by all the classifiers, and N is the total number of samples.

Definition 2.2.2. ***Information gain.*** *If T' denotes the predicted class labels (might be from a set of modalities after fusion or the individual modalities before fusion) and T the*

[1] Typically late fusion models, including the one in this paper, ignore temporal dependencies across modalities as they fuse modalities only after the classifiers have made their decisions. As a result, late fusion models are to some extent indifferent to the heterogeneity of and synchronicity between the different modalities. They allow modality-specific parameters (e.g., sampling rate, window length, feature vector dimension) and representation (e.g., space-time vs. frequency-time) to be chosen for each modality independently. One way to exploit temporal dependencies across modalities in late fusion models is by learning a mapping between the representations of each pair of modalities, either directly (e.g., [Najnin and Banerjee, 2015]) or via intermediate joint representations (e.g., [Hu et al., 2019a]), such that the signal in each modality can be generated from the signal in another. These features can then be used for classification in each modality. Relevant topics include challenges of learning features from time series [Kapourchali and Banerjee, 2018], generative models for learning features from time series [Banerjee and Dutta, 2014a, Najnin and Banerjee, 2019, Najnin and Banerjee, 2017, Dutta et al., 2016, Dutta and Banerjee, 2015, Dutta and Banerjee, 2014], generative models for learning joint representations from multimodal time series [Baruah and Banerjee, 2020a, Baruah and Banerjee, 2020b], and opportunistic sensor selection [Kapourchali and Banerjee, 2019, Kapourchali and Banerjee, 2020]. Exploiting temporal dependencies across modalities is beyond the scope of this paper.

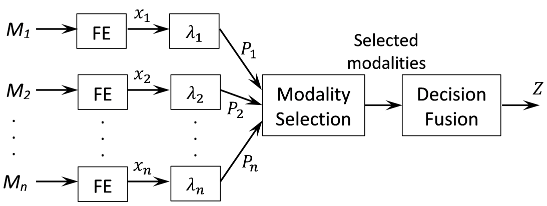


Figure 1.: Block diagram for multimodal time-series classification using late fusion. ‘FE’ refers to feature extraction.

true class labels, the information gain of T' relative to T can be defined as [Mitchell, 1997]:

$$G(T,T') \equiv H(T) - \sum_{v \epsilon Values(T')} \frac{|T_v|}{|T|} H(T_v) \tag{2.2}$$

where $Values(T')$ is the set of all possible values for T' and T_v is the subset of T for which T' has value v and $H(T)$ is the entropy of T and $\sum_{v \epsilon Values(T')} \frac{|T_v|}{|T|} H(T_v) = H(T|T')$.

Theorem 1. *Expected information gain is a convex function of classification accuracy. Given the number of classes c, there exists a unique positive real number a such that the minimum of expected information gain occurs at classification accuracy a. As $c \to \infty$, $a \to 0$.*

For the trivial case $c = 1$, $a = 1$. For the nontrivial case $c = 2$, $a = 0.5$ (computed from Eq. 2.2). Given a particular c, for each classification accuracy $\{0, 0.1, 0.2, ..., 1\}$, we randomly generated at least 10^6 confusion matrices and computed the corresponding mean (or expected) information gain. The expected information gain as a function of classification accuracy is shown in Fig. 2.2 for $c = 2, 3, 7, 25$. In each case, the expected information gain is a convex function of classification accuracy. As c increases, the unique location of the minimum of this function decreases. Since classification accuracy cannot be negative, the location is lower bounded by zero.

This theorem entails that, for any given c, in the accuracy interval $[a, 1]$, expected information gain increases monotonically with classification accuracy. Hence, information

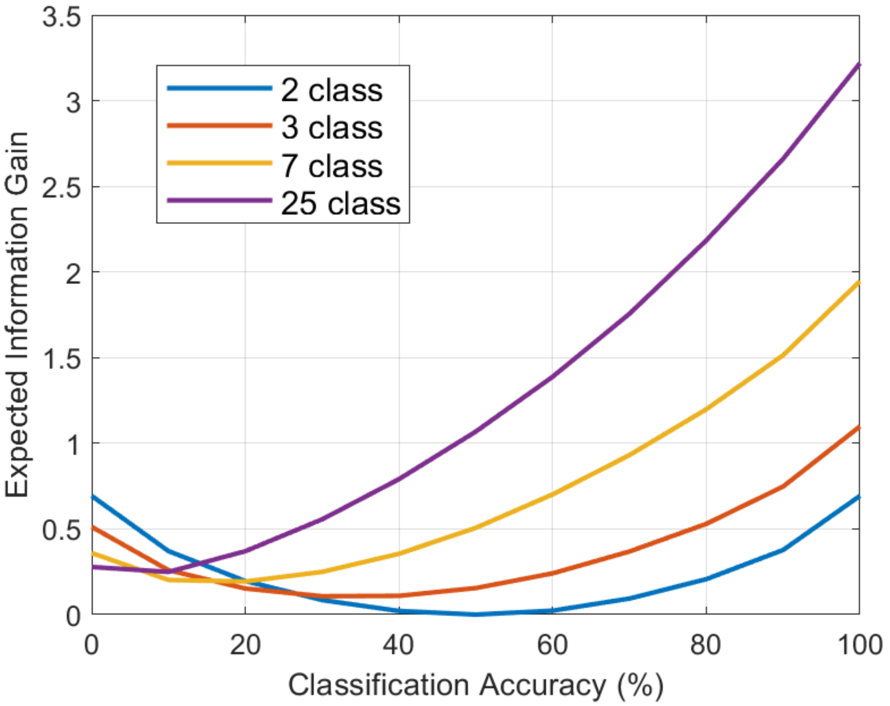


Figure 2.: Expected information gain as a function of classification accuracy is shown for 2, 3, 7 and 25 classes from randomly generated confusion matrices.

gain is probabilistically a sound measure of classification accuracy in $[a, 1]$ where a decreases as c increases.

We experiment with four methods to fuse the posterior probabilities obtained from each classifier at the decision level: product, average, Bayesian, and majority voting. Let Z be the pattern to be assigned to one of the m classes after fusion.

Definition 2.2.3. ***Product [Busso et al., 2004, Gunes and Piccardi, 2005, Kittler et al., 1998].*** *Assign* $Z \rightarrow \omega_j$ *if*

$\prod_{i=1}^{n} P(\omega_j|x_i) = \max_{k=1}^{m} \prod_{i=1}^{n} P(\omega_k|x_i)$ *[Kittler et al., 1998]. The product rule for fusion assumes that the joint probability distribution of the measurements obtained from the classifiers are conditionally independent.*

Definition 2.2.4. ***Average [Busso et al., 2004, Gunes and Piccardi, 2005, Poria et al., 2016, Kittler et al., 1998].*** *Assign* $Z \rightarrow \omega_j$ *if*

$\frac{1}{n}\sum_{i=1}^{n} P(\omega_j|x_i) = \max_{k=1}^{m} \frac{1}{n}\sum_{i=1}^{n} P(\omega_k|x_i)$ *[Kittler et al., 1998]. The average rule for fusion assumes that the prior is equal.*

Definition 2.2.5. ***Bayesian [Ivanov et al., 2005].*** *Assign $Z \to \omega_j$ if $P(\omega_j|x_1, \ldots, x_n)$ $\approx \max_{k=1}^{m} \sum_{i=1}^{n} \sum_{l=1}^{m} P(\omega_k|\tilde{\omega_l}, \lambda_i) P(\tilde{\omega_l}|\lambda_i, x_i) P(\lambda_i|x_i)$ [Ivanov et al., 2005], where $\tilde{\omega_l}$ denotes the predicted class. The probabilities, $P(\omega_k|\tilde{\omega_l}, \lambda_i)$ and $P(\lambda_i|x_i)$, can be approximated from the confusion matrix of λ_i.*

Definition 2.2.6. ***Majority voting [Mangai et al., 2010, Kittler et al., 1998].*** *Assign $Z \to \omega_j$ if $\sum_{i=1}^{n} \Delta_{ji} = \max_{k=1}^{m} \sum_{i=1}^{n} \Delta_{ki}$ [Kittler et al., 1998]. The term $\sum_{i=1}^{n} \Delta_{ki}$ adds the votes for the class ω_k from the individual classifiers. In case of equal votes for multiple classes, the class with the highest posterior probability is selected.*

2.3 Experimental Setup

2.3.1 Experiments

Classification accuracy with respect to number of modalities

To evaluate classification accuracy with increase in number of modalities through an exhaustive comparison, we construct a tree with n leaf nodes, each represents a set containing one modality, and the root node represents the set of n modalities. The tree has n levels. The i^{th} level has $\binom{n}{i}$ nodes, $i = 1, \ldots, n$. A non-leaf node representing a set Q containing q modalities has q child nodes, each representing a set $Q'_j \subset Q$ $(j = 1, \ldots, q)$ containing $q - 1$ modalities. Therefore, the total number of children of all nodes is: $\#Cases = \sum_{i=1}^{n-1} \binom{n}{i}(n - i)$.

Selection of modalities

The selection of modalities is carried out in three ways using information gain and correlation:

1) Use information gain, G, (Eq. 2.2) as a measure to compute the optimal combinations of $1, 2, \ldots, n$ modalities adding one at a time (ref. Algorithm 1). This is a greedy approach where the combination with the highest value of G is always selected.

Table 1.: Classification accuracy (%) reported for the best subset of modalities (i.e. the combination of modalities that yields highest accuracy from all fusion methods) for each of the three modality selection methods (G, ρ, G_s). The "Exhaustive" column reports the highest classification accuracy obtained by considering all possible combinations of modalities (total # combinations $= \sum_{i=1}^{n} \binom{n}{i}$). All modalities present in a dataset are mentioned below the dataset name. The fusion method and the subset of modalities yielding the highest accuracy are reported. The baselines for similar experimental conditions as ours are shown. The highest accuracy for each dataset is highlighted.

Dataset	Exhaustive	G	ρ	G_s	Baseline
DEAP (valence) eeg, gsr, resp, temp, plet, emg, eog	**75.72** - Product eeg, eog, emg, resp	**75.72** - Product eeg, eog, emg, resp	75.41 - Bayesian eeg, eog, emg, plet	**75.72** - Product eeg, eog, emg, resp	57.6 [Koelstra et al., 2012]
DEAP (arousal) eeg, gsr, resp, temp, plet, emg, eog	**66.41** - Bayesian eeg, eog, emg, gsr, resp	66.40 - Bayesian eeg, eog, gsr	65.30 - Bayesian eeg, eog, emg	66.31 - Bayesian eeg, eog	62 [Koelstra et al., 2012]
HCI Tagging (valence) eeg, ecg, gsr, resp, temp	**68.63** - Product eeg, ecg, gsr, resp	**68.63** - Product eeg, ecg, gsr, resp	**68.63** - Product eeg, ecg, gsr, resp	**68.63** - Product eeg, ecg, gsr, resp	56.83 [Wiem and Lachiri, 2017]
HCI Tagging (arousal) eeg, ecg, gsr, resp, temp	**66.36** - Average eeg, ecg, gsr, resp	**66.36** - Average eeg, ecg, gsr, resp	66.19 - Average eeg, ecg, gsr, temp	**66.36** - Average eeg, ecg, gsr, resp	54.73 [Wiem and Lachiri, 2017]
PAMAP2 s1, s2, s3	**90.98** - Product s1, s2, s3	83.57 - Product s1, s2	**90.98** - Product s1, s2, s3	**90.98** - Product s1, s2, s3	89.24 [Reiss and Stricker, 2012a]
UTD-MHAD depth, skel, iner	**92.40** - Product depth, skel, iner	**92.40** - Product depth, skel, iner	78.39 - Product skel, iner	**92.40** - Product depth, skel, iner	79.1 [Chen et al., 2015b]
Berkeley MHAD depth, skel, iner	97.05 - Average depth, skel, iner	97.05 - Average depth, skel, iner	94.22 - Product depth, skel	97.05 - Average depth, skel, iner	**98.23** [Chen et al., 2015a]

Table 2.: Percentage of $\#Cases$ (ref. Section 2.3.1) where classification accuracy decreases as a modality is added. The range of accuracy (%), stated within parentheses, is obtained from all possible combination of modalities (# combinations $= \sum_{i=1}^{n} \binom{n}{i}$).

Method	DEAP (valence)	DEAP (arousal)	HCI Tagging (valence)	HCI Tagging (arousal)	PAMAP2	UTD -MHAD	Berkeley MHAD
Product	47.85 (15.68)	64.17 (8.35)	18.67 (25.83)	16 (23.56)	0 (20.26)	0 (29.19)	0 (16.66)
Average	47.85 (15.60)	64.40 (8.35)	18.67 (25.44)	14.67 (23.72)	33.33 (13.32)	0 (24.08)	33.33 (18.91)
Bayesian	34.01 (15.66)	48.30 (8.66)	6.67 (22.62)	6.67 (20.62)	22.22 (11.69)	0 (23.87)	22.22 (17.57)
Voting	47.39 (15.15)	49.21 (8.41)	46.67 (19.9)	41.33 (20.15)	0 (9.16)	0 (22.50)	0 (14.19)

2) Use correlation degree, ρ, [Goebel et al., 2002, Niu et al., 2007] as a measure to compute the optimal combinations of $1, 2, \ldots, n$ modalities adding one at a time. Select the subset with lowest ρ, similar to the approach in [Goebel et al., 2002].

3) Assign a score, G_s, equal to its information gain (Eq. 2.2) to each modality and select the modalities with score greater than an appropriate threshold (0.2) which is determined experimentally and applies to all the datasets.

2.3.2 Training procedure

A total of 33 multilayer perceptron (MLP) classifiers (7×2 for DEAP, 5×2 for HCI Tagging, 3 for PAMAP2, 3 for UTD MHAD and 3 for Berkeley MHAD) are trained, one for each modality for each dataset. The hyperparameters (batchsize, number of layers, activation functions (tanh, relu), number of neurons in each layer, learning rate, dropout (for MLPs with more than one hidden layer)) for each MLP are fixed experimentally. We use a softmax classifier at the final layer and binary cross entropy as the cost function during training.

DEAP [Koelstra et al., 2012]: As in [Tripathi et al., 2017], the individual rated

Algorithm 1 Selecting modalities using information gain

```
**Inputs:** $P_1, \dots, P_n$.
**Output:** Set $s$.
**Initialize:** $S \longleftarrow \{M_{best}\}$, $S' \longleftarrow \{M_1, M_2, \dots, M_n\} - S$, $M_{best}$ is the modality yielding highest recognition accuracy.
Compute $G^S(1) \longleftarrow G(T', T)$ using Eq. 2.2, where $T'$ are the predicted class labels from $M_{best}$.
**for** $i = 2$ $to$ $n$ **do**
    **for** $j = 1$ $to$ $|S'|$ **do**
        Compute predicted class labels, $T' \longleftarrow f(\{P_{M_i} : \lambda_{M_i} \epsilon S \bigcup S'_j\})$ such that $f$ represents any late fusion operation and $S'_j$ denotes the $j^{th}$ element in set $S'$.
        Compute information gain $G^{S'}(j) \longleftarrow G(T', T)$ using Eq. 2.2.
    **end for**
    $G^S(i) \longleftarrow \max G^{S'}$
    $k \longleftarrow \arg_j \max G^{S'}$
    $S \longleftarrow S \bigcup S'_k$
    $S' \longleftarrow S' - S'_k$
**end for**
$k \longleftarrow \arg_i \max G^S$
**Return** $s \longleftarrow S_{1:k}$.
```

scales (1-9) are mapped to two levels of each valence and arousal states such that 1-5 is mapped to low and 5-9 to high emotion labels. The valence and arousal classification is done separately. As in [Tripathi et al., 2017, Koelstra et al., 2012], the train/test set split is obtained by leaving one subject out cross-validation such that data from 31 subjects constitute the train set and data from the remaining subject constitute the test set. The mean classification accuracy from 32-fold cross-validation is reported. The preprocessed dataset available in MATLAB format in the downloaded dataset is used in our experiments. We extract six statistical features for all the 7 modalities considered in our experiments: means and standard deviations of the raw signals, means of absolute values of the first and second differences of the raw signals, and means of absolute values of the first and second differences of the normalized signals [Picard et al., 2001], from a 6-second window without overlap [Tripathi et al., 2017] such that each window constitutes a datapoint.

HCI Tagging [Soleymani et al., 2012]: The individual rated scales (1-9) are mapped to three levels of each valence and arousal states such that 1-3 is mapped to low, 4-6 to medium and 7-9 to high, as in [Wiem and Lachiri, 2017]. The valence and arousal classification is performed separately. Two-third of the dataset is used for training, as in [Wiem and Lachiri, 2016]. The experiments are repeated 10 times. The mean classification accuracy is reported.We extract 11 statistical features for all the 5 modalities considered in our experiments: the six features as in the DEAP dataset, skewness, kurtosis, min, max, and median [Picard et al., 2001,Tripathi et al., 2017] from 6-second windows without overlap [Tripathi et al., 2017] such that each window constitutes a datapoint.

PAMAP2 [Reiss and Stricker, 2012b]: We consider 12 actions for recognition, as in [Reiss and Stricker, 2012b]. The train/test set split is obtained by leaving one subject out cross-validation; data from 8 subjects constitute the train set, as in [Reiss and Stricker, 2012b]. The mean classification accuracy from 9-fold cross-validation is reported. We extract features from a 2-second window without overlap. Six statistical features as in the DEAP dataset are extracted from sensors 1 and 2. Three features are extracted from sensor 3 as in [Chen et al., 2015b]: mean, variance, and standard deviation. The features are extracted for each dimension of the inertial signals and then concatenated.

UTD MHAD [Chen et al., 2015b]: There are 27 action classes. Data from subjects $\{1, 3, 5, 7\}$ is used for training and subjects $\{2, 4, 6, 8\}$ for testing, as in [Chen et al., 2015b]. The experiments are repeated 10 times and the mean of the classification accuracy over all the experiments is reported. We extract 8 statistical features from each 6-second window, as in [Chen et al., 2015b], for inertial data across all the dimensions: mean, median, max, min, standard deviation, variance, skewness, kurtosis. Variance is extracted from 8 windows, as in [Chen et al., 2015b], for each video for each dimension of the motion capture data. The features are computed over all dimensions, then concatenated over all the dimensions and windows. Depth motion map (DMM) features, as in [Chen et al., 2015b] are extracted from the depth videos.

Berkeley MHAD [Ofli et al., 2013]: There are 11 action classes. Data from the first 7 subjects are used for training and the last 5 for testing, as in [Ofli et al., 2013, Du et al., 2015]. The experiments are repeated 10 times and mean classification accuracy is reported. Each inertial sequence is divided into 30 windows and each accelerometer sequence into 60 windows, as in [Chen et al., 2015a]. Variance is extracted for inertial and accelerometer data from each window and across all the dimensions which is then concatenated over all the dimensions and windows. As in [Chen et al., 2015a], DMM features constitute our feature vector for the depth videos.

2.4 Experimental Results

Classification accuracy with respect to number of modalities.

Our results show that an increase in the number of modalities may not increase the classification accuracy, especially for emotion recognition from physiological signals (ref. Table 1. and Table 2.). Adding more modalities for fusion might add noise and create confusion leading to misclassification. The decrease in classification accuracy with increasing modalities is observed for all the fusion methods and all datasets except UTD-MHAD dataset, as shown in Table 2.. Of the four fusion methods, Bayesian fusion yields the smallest decrease in accuracy with increase in number of modalities compared to other fusion methods for all the datasets. This is because Bayesian fusion takes into account the uncertainty of the classifiers for each class by combining the classifiers as a weighted combination of the error distribution over the classes.

Evaluation of different metrics for selecting modalities.

Our results (ref. Table 1.) show that, for all the datasets, the classification accuracy obtained from the subset of modalities selected using information gain is closer to the highest accuracy than that selected using correlation degree. Hence, information gain outperforms correlation degree as a criterion for modality selection. The lowest absolute difference between the true highest accuracy (“Exhaustive”) and the accuracy obtained using selected modalities, added over all datasets, is 1.79 for product fusion, 1.7 for average fusion, 0.03 for Bayesian fusion, and 0.01 for majority voting, obtained using metrics G_s, G, G and G respectively. Since information gain increases monotonically with classification accuracy (ref. Theorem 1), it is a useful metric for selecting a subset of modalities that will yield high accuracy. As shown in Table 1., the subset of modalities selected using different metrics always yields a classification accuracy comparable, if not equal, to the highest accuracy.

The correlation degree criteria initially selects the modality highly correlated with the true class labels, then it selects modalities least correlated with the selected modalities. This helps in reducing redundancy. However, it can reduce relevant information as well which can lower the classification accuracy, as observed from our results. On the other hand, selecting modalities based on their individual score using information gain and filtering them using a threshold, allows selection of modalities highly correlated with the true class labels. This preserves relevant information but might have high redundancy. However, it outperforms correlation degree as a selection criteria, as seen from our experiments.

Information gain modality selection method (ref. Algorithm 1) selects the combination of modalities after fusion that has the highest correlation with the true class label. This yields the highest classification accuracy, comparable with the true best combination in our experiments. Algorithm 1 requires fusing the modalities before computing the information gain. Hence, it depends on the fusion method while the other two, correlation degree and filtering using information gain, are independent of the fusion method.

2.5 Conclusions

In this paper, we investigated the optimal modality selection problem for time-series data in the context of late fusion. We analyzed multimodal emotion or action classification using four late fusion methods and five benchmark datasets. Our experimental analysis on product, average, Bayesian and majority voting late fusion methods show that the fusion methods perform differently based on the posterior distribution estimated by each modality. Our results show that for different fusion methods, increasing the number of modalities might not necessarily increase the classification accuracy. We analyze multiple methods for selecting a subset of modalities for late fusion and observe that information gain is an useful measure for selecting modalities which is consistent for all the datasets. The classification accuracy obtained from the selected subset of modalities is comparable to the highest accuracy in all cases.

Chapter 3

An Attention-based Predictive Agent for Static and Dynamic Environments

Abstract: Real-world applications of intelligent agents demand accuracy and efficiency, and seldom provide reinforcement signals. Currently, most agent models are reinforcement-based and concentrate exclusively on accuracy. We propose a general-purpose agent model consisting of proprioceptive and perceptual pathways. The agent actively samples its environment via a sequence of glimpses. It completes the partial propriocept and percept sequences observed till each sampling instant, and learns where and what to sample by minimizing prediction error, without reinforcement or supervision (class labels). The model is evaluated by exposing it to two kinds of stimuli: images of fully-formed handwritten numerals and alphabets, and videos of gradual formation of numerals. It yields state-of-the-art prediction accuracy upon sampling only 22.6% of the scene on average. The model saccades when exposed to images and tracks when exposed to videos. This is the first known attention-based agent to generate realistic handwriting with state-of-the-art accuracy and efficiency by interacting with and learning end-to-end from static and dynamic environments.

3.1 Introduction

Perception and action are inextricably tied together as, in the real world, efficiency is as important as accuracy. Nature has evolved the visual system such that, to minimize resources, it learns to selectively attend to a few locations that provide information for the task at hand. We propose a predictive agent model, which observes its visual environment via a sequence of glimpses. It predicts, learns and acts by minimizing sensory prediction error in a closed loop.

The agent is evaluated on handwriting generation. The model is exposed to images of fully-formed handwritten numerals and alphabets (MNIST, EMNIST datasets) and videos of gradual formation of numerals (SMNIST dataset). This allows evaluation of the agent in static (image) and dynamic (video) environments. In handwriting generation, the agent learns to sequentially sample its visual environment.

Related work. Attention-based models can be hard or soft [Xu et al., 2015, Elsayed et al., 2019]. Hard-attention models make decisions by processing a part of the data, sampled via a sequence of glimpses. These models are reinforcement-based (e.g., [Elsayed et al., 2019, Mnih et al., 2014]), unsupervised (e.g., [Gregor et al., 2015, Eslami et al., 2016]) or supervised (e.g., [Zheng et al., 2015]). Soft-attention models process the entire data but weigh the features. Supervised (e.g., [Fukui et al., 2019]) and unsupervised (e.g., [Sang et al., 2020]) variants of these models have been reported. We propose an unsupervised (no class label) hard-attention model.

A number of models have been proposed for handwriting generation, such as [Gregor et al., 2014, Gregor et al., 2015, Oord et al., 2016, Zhao et al., 2017, Maaloe et al., 2019, Ma et al., 2019, Sadeghi et al., 2019, Vahdat and Kautz, 2020, Standvoss et al., 2020]. Only one of them, DRAW [Gregor et al., 2015], is an unsupervised hard-attention model. In DRAW, attention is explicitly learned. In our model, attention emerges as a consequence of minimizing the prediction error, similar to the model in [Standvoss et al., 2020]. However, our prediction error computation function is different from that in [Standvoss et al., 2020]. Our function selects the location with maximum information gain at each glimpse. Also, this model is supervised (uses class labels). This model and DRAW have reported results only on images while our model operates on images and videos. Though the role of attention is to foster efficiency, most works on attention-based models, including this model and DRAW, do not report on their efficiency. We evaluate the efficiency of our model with respect to its size (number of trainable parameters) and the number of glimpses, or equivalently, fraction of scene, required for accurate prediction.

Novelties of our agent model. **(1)** It implements the perception-action loop as the optimization of an objective function. Action/attention is modeled as proprioception in a multimodal setting, and is guided by the perceptual prediction error, not by reinforcement.

(2) The same model can be used for static and dynamic environments. We show applications on image and video. Behaviorally, the agent saccades and tracks when exposed to images and videos respectively. **(3)** This end-to-end model is efficient in terms of size and number of glimpses required for accurate prediction. It learns by sampling locations with maximum information gain at each glimpse. Consequently, it yields state-of-the-art prediction accuracy upon sampling only 22.6% of the scene on average. By the fourth glimpse which corresponds to 11.2% of the scene, the prediction error drops by 60.4%. **(4)** It yields state-of-the-art accuracy in handwriting generation. In particular, it yields 4.9% lower error than the DRAW model on the binarized MNIST benchmark.

3.2 Models and Methods

3.2.1 Preliminaries

Agent. Anything that perceives from and acts upon its environment using sensors and actuators respectively is called an agent [Russell and Norvig, 2020].

Perception is the mechanism of interpreting sensory signals from the external environment by an agent [Han et al., 2016].

Proprioception is a form of perception in which the agent's environment is its own body. Internal perception of position, movement, and motion of body parts is due to proprioception [Han et al., 2016].

Generative model. Given a set of data points x, a generative model p_{model} with parameters θ maximizes the log-likelihood, $\mathcal{L}(x;\theta)$, of the data.

Evidence lower bound (ELBO). Let the data x be generated by a latent continuous random variable z. Then, computing the log-likelihood requires integrating the marginal likelihood, $\int p_{model}(x,z)dz$, which is intractable [Kingma and Welling, 2013]. In variational inference, an approximation of the intractable posterior is optimized by defining an evidence lower bound (ELBO) on the log-likelihood, $\mathcal{L}(x;\theta) \leq \log p_{model}(x;\theta)$.

Variational autoencoder (VAE) is a multilayered generative model. It assumes an isotropic Gaussian prior, $p_\theta(z)$, and i.i.d. data samples. VAE maximizes the following ELBO [Kingma and Welling, 2013]:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}[q_\phi(z|x), p_\theta(z)] \tag{3.1}$$

where $p_\theta(x|z)$ and $q_\phi(z|x)$ are generative and recognition models respectively, $\mathbb{E}$ denotes expectation, and D_{KL} denotes Kullback-Leibler divergence. The first and second terms capture accuracy and complexity respectively. The negative of this ELBO is also known as *variational free energy*, minimization of which has been hypothesized as a general principle guiding brain function [Friston, 2010].

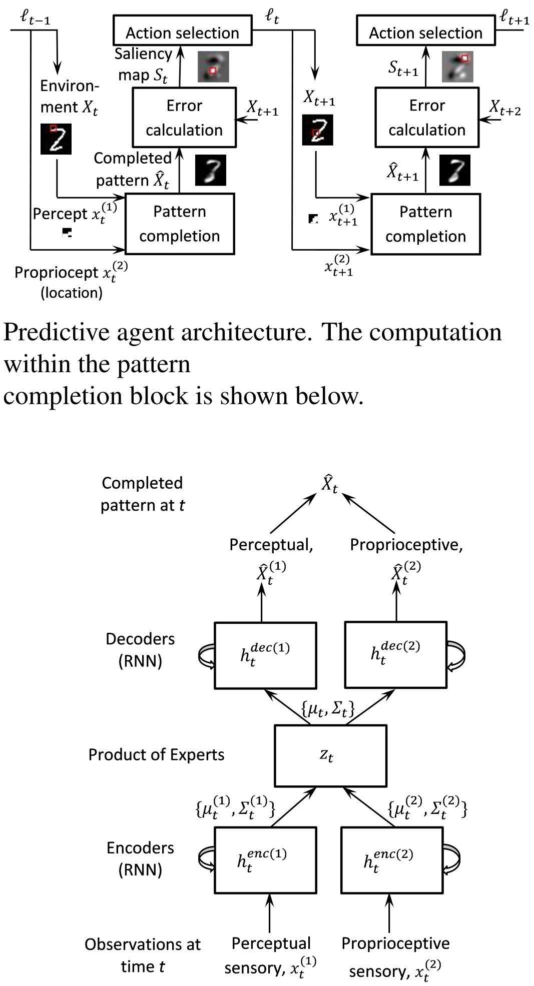
Saliency lies in the eyes of an agent. Saliency of a location in an environment is a function of its neighborhood and an agent's internal model (see [Spratling, 2012, Friston et al., 2009]).

3.2.2 Problem Statement

Let an environment in m modalities be represented by a set of observable variables $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(m)}\}$. The variable representing the i-th modality is a sequence: $\mathbf{X}^{(i)} = \langle X_1^{(i)}, X_2^{(i)}, \ldots, X_T^{(i)} \rangle$, where T is the sequence length. Let $\mathbf{x}_{\leq t} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\}$ be a partial observation of $\mathbf{X}$ such that $\mathbf{x}^{(i)} = \langle x_1^{(i)}, \ldots, x_t^{(i)} \rangle$, $1 \leq t \leq T$. As in [Baruah and Banerjee, 2020b], we define *pattern completion* as the problem of accurately generating $\mathbf{X}$ from its partial observation $\mathbf{x}_{\leq t}$. Given $\mathbf{x}_{\leq t}$ and a generative model p_θ with parameters θ and latent variables $z_{\leq t}$, the generative process of $\mathbf{X}$ is given as $p_\theta(\mathbf{X}|\mathbf{x}_{\leq t}) = \int p_\theta(\mathbf{X}|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_\theta(z_{\leq t}) dz$. The objective for pattern completion at any time t is to maximize the log-likelihood of $\mathbf{X}$, i.e. $\arg\max_\theta \int log(p_\theta(\mathbf{X}|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_\theta(z_{\leq t})) dz$.

Table 3.: Variable dimensions as used in this paper. Here $(.)^{(1)}$, $(.)^{(2)}$ refer to visual perception and visual proprioception respectively; T is maximum number of glimpses, t is glimpse index or time, $n \times n$ is patch size, $M \times M$ is image size.

$x_t^{(1)}$	$x_t^{(2)}$	$X_t^{(1)}$	$X_t^{(2)}$	$S_t^{(1)}$
$\{0,1\}^{n \times n}$	$\mathbb{R}^2$	$\{0,1\}^{M \times M}$	$\mathbb{R}^{2 \times T}$	$\mathbb{R}^{M \times M}$



Predictive agent architecture. The computation within the pattern
completion block is shown below.

Pattern completion model.

Figure 3.: Different components of the proposed agent.

3.2.3 Agent Architecture

As shown in the block diagram in Fig. 3.a, environment, observation, pattern completion, action selection, and learning are the five components of the proposed agent architecture.

1. Environment. Two kinds of environment, or source of sensory data, are considered: static (images) and dynamic (videos).

2. Observation. Our agent sequentially samples its environment in two modalities: visual perception and visual proprioception. The 2D coordinates of the fixation location in the environment constitutes the proprioceptive observation while the visual stimuli at that location constitutes the corresponding perceptual observation, as in [Friston et al., 2012]. See Table 3. for variable dimensions.

3. Pattern completion. Patterns in the two modalities are completed using a multimodal variational recurrent neural network (VRNN). Recognition and generation are the two processes involved in the operation of a VRNN [Chung et al., 2015].

Recognition (Encoder). The recognition model, $q_\phi(z_t|\mathbf{x}_{\leq t})$, is a probabilistic encoder [Kingma and Welling, 2013]. It produces a Gaussian distribution over the possible values of the code z_t from which the given observations $\mathbf{x}_{\leq t}$ could have been generated. Two RNNs, each with one layer of long short-term memory (LSTM) units, constitute the recognition model. Each RNN generates the parameters for the approximate posterior distribution for each modality. The parameters for all modalities are combined using product of experts (PoE) [Wu and Goodman, 2018] to generate the joint distribution parameters for the approximate posterior $q_\phi(z_t|\mathbf{x}_{\leq t})$. The prior can be sampled from a standard normal distribution $p_\theta(z_t) \sim \mathcal{N}(0, 1)$ as in [Gregor et al., 2015]. The function of the encoder is shown in Lines 1–5 in Algorithm 3, where RNN_ϕ^{enc} represents the function of a LSTM unit, φ^{enc} is a function that returns the mean and the logarithm of the standard deviation as a linear function of the hidden state, as in [Chung et al., 2015].

Generation (Decoder). The generative model, $p_\theta(\mathbf{X}_t|\mathbf{x}_{<t}, z_{\leq t})$, generates the data from the latent variables, z_t, at each time step. The generative model has two RNNs with one layer of hidden LSTM units. Each RNN generates the parameters of the distribution of the sensory data for a modality. The sensory data is sampled from this distribution, which can be multivariate Gaussian or Bernoulli. In our model, $X_t^{(1)}$ is sampled from a multivariate Bernoulli distribution (as the perceptual observation is binary) with means generated by the perceptual decoder RNN, and $X^{(2)}$ is sampled from a multivariate Gaussian distribution (as the proprioceptive observation is real) with means and variances as output of the proprioceptive decoder RNN (see Fig. 3.b). The pattern, $p_\theta(\mathbf{X}|\mathbf{x}_{<t}, z_{\leq t})$, is completed at every time step. In order to generate the perceptual data at any time step, the

output from the perceptual RNN at the previous time step is added to the current perceptual RNN output before applying the sigmoid function, as in [Gregor et al., 2015] (ref. Line 8 of Algorithm 3). The decoder equations are shown in Lines 7–11 of Algorithm 3, where RNN_θ^{dec} and φ^{dec} are same as RNN_ϕ^{enc} and φ^{enc}.

4. Action selection. In our model, action selection is to decide the location in the environment to sample from. At any time t, a saliency map S_t is computed which assigns a salience score $S_t^{(\ell)}$ to each location ℓ:

$$S_t^{(\ell)} = D_{KL}(p(X_{t+1,\ell}^{(1)})||p_\theta(X_{t+1,\ell}^{(1)}|z_{\leq t}, \mathbf{x}_{\leq t})) \tag{3.2}$$

where $p(X_{t+1,\ell}^{(1)})$ is the true data distribution at location ℓ and is sampled from a Bernoulli distribution. KL divergence, also known as *relative entropy*, is a measure of information gain achieved by using the true distribution, $p(X_{t+1,\ell}^{(1)})$, instead of the predicted distribution, $p_\theta(X_{t+1,\ell}^{(1)}|z_{\leq t}, \mathbf{x}_{\leq t})$. Thus, the saliency map is a function of the prediction error. The most salient location is computed from this saliency map, which constitutes the sampling location.

The saliency map is smoothed using a Gaussian kernel $\mathcal{N}(., \sigma)$. The sampling location is chosen as:

$$\ell_t = \underset{\ell_t \in \{1,2,\ldots,M^2\}}{\text{argmax}} \text{ conv}(\mathcal{N}(., \sigma), S_t) \tag{3.3}$$

where $\sigma = 2$. Each sample is a $n \times n$ patch centered at ℓ_t.

The salient location ℓ_t at any time t is the proprioceptive observation $x_{t+1}^{(2)}$ for time $t+1$. Therefore, the salient locations at $t = 1, 2, \ldots, T$ constitutes the proprioceptive pattern $\mathbf{X}^{(2)}$. Hence, prediction error (saliency) guides the sampling of the scenes in our model. Unlike typical multimodal models, the two modalities in our model interact at the observation level as the perceptual prediction error provides the observation for the visual proprioceptive modality. The agent learns a policy to generate the proprioceptive pattern or the sequence of expected salient locations by minimizing the proprioceptive prediction error. This error, at any time, is a function of the difference between predicted fixation location from the learned policy and the most salient location in the scene. The most salient location is the location that yields the maximum information gain in the environment. These are the locations where the agent's prediction error is the highest given all the past observations. The agent attends to these locations to update its internal model.

5. Learning. The recognition and generative model parameters are jointly learned by maximizing the ELBO for the multimodal variational RNN. This objective, obtained by modifying the objective for multimodal VAE [Wu and Goodman, 2018] with variational RNN [Chung et al., 2015], is to maximize

$$\begin{aligned}\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i \log p_\theta(X_t^{(i)}|z_{\leq t},\mathbf{x}_{<t})\Big]\\ -\sum_{t=1}^{T}\beta D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}),p_\theta(z_t)\big)\end{aligned} \tag{3.4}$$

where λ_1, λ_2, β are the weights balancing the terms.

Loss function derivation

Here we derive the objective function in Eq. 3.4. The generative and recognition models are factorized as:

$$p_\theta(\mathbf{X}_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^{T} p_\theta(\mathbf{X}_t|z_{\leq t},\mathbf{x}_{<t})p_\theta(z_t)$$

$$q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\leq t})$$

The variational lower bound (ELBO) on the log-likelihood of the generated data,

$\log p_\theta(\mathbf{X}_{\leq T}|\mathbf{x}_{\leq T})$, is derived as:

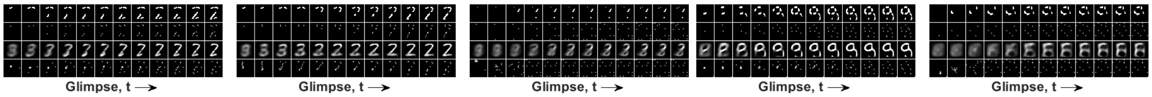
$$\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\log p_\theta(\mathbf{X}_{\leq T}|\mathbf{x}_{\leq T})\frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big]\\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\log \frac{p_\theta(\mathbf{X}_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T})}{p_\theta(z_{\leq T}|\mathbf{x}_{\leq T})}\frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big]\\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\log \frac{p_\theta(\mathbf{X}_t|z_{\leq t}, \mathbf{x}_{<t})p_\theta(z_t)}{p_\theta(z_t|\mathbf{x}_{\leq t})}\frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{q_\phi(z_t|\mathbf{x}_{\leq t})}\Big]\\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\Big[\log p_\theta(\mathbf{X}_t|z_{\leq t}, \mathbf{x}_{<t})\\
&\quad - \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{p_\theta(z_t)} + \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{p_\theta(z_t|\mathbf{x}_{\leq t})}\Big]\Big]\\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\log p_\theta(\mathbf{X}_t|z_{\leq t}, \mathbf{x}_{<t})\Big]\\
&\quad - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big)
\end{aligned}$$

We assume, the modalities $X_t^{(1)}$ and $X_t^{(2)}$ are conditionally independent given the common latent variables [Wu and Goodman, 2018] and all observations till the current time. Therefore,

$$\log p_\theta(\mathbf{X}_t|z_{\leq t}, \mathbf{x}_{\leq t}) = \sum_{i=1}^{2}\log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t})$$

Thus,

$$\begin{aligned}
&\log p_\theta(\mathbf{X}_{\leq T}|\mathbf{x}_{\leq T})\\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t})\Big]\\
&\quad - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big)\\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i\log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t})\Big]\\
&\quad - \sum_{t=1}^{T} \beta D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big)
\end{aligned}$$



MNIST SMNIST MNIST-random EMNIST EMNIST-random

Figure 4.: Pattern completion for (a) a random example (‘2’) from MNIST test set, (b) same example from SMNIST, (d) a random example (‘g’) from EMNIST; (c) and (e) correspond to examples (a) and (d) respectively when the observations are sampled randomly and not from the saliency map. Here patch size is 5×5. Each column in subfigures a–e corresponds to time or glimpse number. Rows 1, 2 show the perceptual and proprioceptive observation till the current glimpse in 28×28 space. Rows 3, 4 show the perceptual and proprioceptive pattern completion after each glimpse.

Algorithm 2 Learning the proposed network

```
1:  Initialize parameters of the generative model $\theta$, recognition model $\phi$, sequence length $T$.
2:  Initialize optimizer parameters: $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\eta = 0.001$, $\epsilon = 10^{-10}$.
3:  Initialize $x_1^{(1)} \leftarrow F(X_1^{(1)}, \ell_0)$, $x_1^{(2)} \leftarrow g_3(\ell_0)$, where $\ell_0$ is the initial sampling location (ref. Experimental setup in Section 3.3), $g_3$ is an identity function (ref. Action selection in Section 3.2.3), and the function $F$ extracts a sample $x^{(1)}$ (e.g., $5 \times 5$ patch) from the environment $X^{(1)}$ (e.g., $28 \times 28$ image) at location $\ell$ (e.g., center of the image).
4:  while true do
5:      for $\tau \leftarrow 1$ to $T$ do
6:          $\hat{X}_\tau^{(1:2)} \leftarrow PatternCompletion(x_{1:\tau}^{(1:2)})$
            Saliency Computation
7:          $S_\tau \leftarrow g_1(X_{\tau+1}^{(1)}, \hat{X}_\tau^{(1)})$        [ref. Eq. 3.2]
8:          $\ell_\tau \leftarrow g_2(S_\tau)$        [ref. Eq. 3.3]
9:          $x_{\tau+1}^{(2)} \leftarrow g_3(\ell_\tau)$
10:         $x_{\tau+1}^{(1)} \leftarrow F(X_{\tau+1}^{(1)}, \ell_\tau)$
            Learning
11:         Update $\{\theta, \phi\}$ by maximizing Eq. 3.4.
12:     end for
13: end while
```

where λ_1, λ_2, β are the weights balancing the terms.

We assume a one-to-one mapping between the agent’s body and its environment, i.e. between the oculomotor muscles to the locations in the image/video frame. This assumption allows us to map from the perceptual space ℓ to the proprioceptive space $x^{(2)}$ using a simple function g_3 (ref. Line 9 in Algorithm 2).

Algorithm 3 $PatternCompletion(x^{(1:2)}_{1:\tau})$

```
Recognition Model
for $i \leftarrow 1\ to\ 2$ do
    $h^{enc(i)}_{\tau} \leftarrow RNN^{enc}_{\phi}(x^{(i)}_{1:\tau}, h^{enc(i)}_{\tau-1})$
    $[\mu^{(i)}_{\tau}\,; \sigma^{(i)}_{\tau}] \leftarrow \varphi^{enc}(h^{enc(i)}_{\tau})$
end for
Product of Experts
$z_{\tau} \sim \mathcal{N}(\mu_{\tau}, \Sigma_{\tau})$, where
$\Sigma_{\tau} \leftarrow \Big(\sum_{i=1}^{2} \Sigma^{(i)}_{\tau}{}^{-2}\Big)^{-1},\ \mu_{\tau} \leftarrow \Big(\sum_{i=1}^{2} \mu^{(i)}_{\tau} \Sigma^{(i)}_{\tau}{}^{-2}\Big)\Sigma_{\tau}$
Generative Model
For perceptual modality:
$h^{dec(1)}_{\tau} \leftarrow RNN^{dec}_{\theta}(z_{\tau}, h^{dec(1)}_{\tau-1})$
$\hat{X}^{(1)}_{\tau} \leftarrow f_{\sigma}(h^{dec(1)}_{\tau}, \hat{X}^{(1)}_{\tau-1})$
For proprioceptive modality:
$h^{dec(2)}_{\tau} \leftarrow RNN^{dec}_{\theta}(z_{\tau}, h^{dec(2)}_{\tau-1})$
$[\mu^{(2)}_{x^{(2)},\tau}\,; \sigma^{(2)}_{x^{(2)},\tau}] \leftarrow \varphi^{dec}(h^{dec(2)}_{\tau})$
$\hat{X}^{(2)}_{\tau} \leftarrow \mu^{(2)}_{x^{(2)},\tau}$
```

3.3 Experimental Results

Our model is implemented using TensorFlow 1.3 in Python 3.5.4. All experiments are carried out in HPC using PowerEdge R740 GPU nodes equipped with Tesla V100 PCIE 16GB.

Datasets. Three datasets are used to evaluate our model:

(1) MNIST [LeCun et al., 1998] is a dataset of handwritten numerals $\{0, 1, \ldots 9\}$, consisting of 60,000 training and 10,000 test images (28 × 28 pixels).

(2) EMNIST [Cohen et al., 2017] is a balanced dataset of handwritten English alphabets in uppercase and lowercase, consisting of 124,800 training and 20,800 test images (28 × 28 pixels).

(3) MNIST stroke sequence dataset (SMNIST) [de Jong, 2016] was designed to learn sequences from MNIST images. It consists of a sequence of locations forming each MNIST numeral. We create a video for each image by selecting an equal number of more

or less equidistant locations. Each frame is 28×28 pixels. Such videos show the gradual formation of numerals.

Experimental setup. For each modality, the generative and recognition models consist of 512 and 64 hidden units respectively. The latent variable dimension is 10. These parameters are estimated experimentally, as shown in Fig. 7.. Maximum number of glimpses $T = 12$, and minibatch size is 100. The parameters β, λ_1, λ_2 are fixed to 1. The model is learned end-to-end using backpropagation and Adam optimization [Kingma and Ba, 2014] with a learning rate of 0.001. These hyperparameters are estimated via cross-validation using 10,000 images or videos from the training set. The first observation is sampled from the starting pixel of the numeral in a SMNIST video as obtained from [de Jong, 2016], and the center pixel of an image in MNIST and EMNIST. Fixing the first observation (or origin) on an object (egocentric reference) allows learning a position-invariant representation of the object.

The quality of generated images is measured using negative log-likelihood (NLL), as in [Gregor et al., 2015]. Efficiency of the model is evaluated with respect to number of trainable parameters and number of glimpses required for accurate prediction.

Evaluation for accuracy. At the initial time steps, the completed patterns are of poor quality (ref. Figs. 4.a, b, d) as the agent samples from the latent distribution of multiple classes. Within a few glimpses, the predictions improve significantly. The examples in Figs. 4.c, e show that when the agent samples the input space randomly, it may sample uninformative locations and will require more observations to determine the true class and generate the data accurately.

For static environment (image), the actual sequence of salient locations, $x^{(2)}_{1:T}$, and

Table 4.: Increase in number of trainable parameters with patch size in our model. Baseline patch size is 5×5 pixels.

Patch size	9×9	13×13	17×17	21×21
# additional parameters	57344	147453	270336	425984

Table 5.: Comparison of generation accuracy at the final time step (T) between variants of the proposed model. Perceptual (Perc.) and proprioceptive (Prop.) modalities are shown separately for each dataset. Best results are highlighted.

Dataset	Variants of proposed model	Perc. NLL	Prop. NLL
MNIST	w/ prop. w/o prop.	**1125.3** 1794.2	-761.6
EMNIST	w/ prop. w/o prop.	**1331.1** 2332.0	-694.5
SMNIST	w/ prop. w/o prop.	**1439.25** 2028.2	-745.11

Table 6.: Prediction error (negative log-likelihood or NLL) comparison on binarized MNIST dataset [Salakhutdinov and Murray, 2008]. Baseline refers to the case where the entire image is sampled by our model at any glimpse, i.e. it observes 100% of the ground truth.

Attention models		Non-attention models									
Ours	DRAW [Gregor et al., 2015]	Baseline (Ours w/o attn.)	DARN 1hl [Gregor et al., 2014]	Pixel CNN [Oord et al., 2016]	Info-VAE [Zhao et al., 2017]	Row LSTM [Oord et al., 2016]	Diagonal BiLSTM [Oord et al., 2016]	BIVA [Maaloe et al., 2019]	NVAE [Vahdat and Kautz, 2020]	Pixel-VAE++ [Sadeghi et al., 2019]	MAE [Ma et al., 2019]
$\leq \mathbf{76.99}$	$\leq$80.97	$\leq$79.5	$\approx$84.13	81.3	$\leq$80.76	80.54	79.2	$\leq$78.59	$\leq$78.01	$\leq$78	$\leq$77.98

that predicted by our model, $\hat{X}^{(2)}$, are randomly distributed over the shape of an object (numeral or alphabet). Hence, with increase in number of glimpses, the distribution of salient locations for an object resembles its shape (ref. Figs. 5.a, b). For dynamic environment (video), the sequence of salient locations, both actual and predicted, follow the motion. For example, in Figs. 5.c, d, the salient location for '0' starts from top-left and ends at top after traversing in clockwise and anticlockwise directions as both formations of '0' are present in SMNIST. Thus, an interesting behavior emerges in our agent—it saccades while observing images and tracks the formation of objects while observing videos.

For both static and dynamic cases, the actual and predicted proprioceptive pattern distributions for each object class, obtained by averaging the actual and predicted salient locations from the test set, are quite similar. Thus in both cases, the distribution of salient locations is learned by the agent from its own behavior.

Our model's prediction accuracy, reported at the final time step as in [Gregor et al., 2015], for MNIST is higher than the existing state-of-the-art (ref. Table 6.). In this comparison, we have considered all recent attentional and non-attentional models that have reported prediction accuracy on the binarized version of MNIST dataset [Salakhutdinov and Murray, 2008] in terms of NLL. Under similar conditions, the NLL for EMNIST is 76.6. NLL on the EMNIST dataset is being reported for the first time in this work.

The results in Table 6. are obtained with our model's encoder and latent variable dimensions as 128 and 20 respectively. It yields $(1 - 76.99/80.97) = 4.9\%$ lower error than the DRAW model. Our model (with attention) observes at most 23.4$\%$ of the ground truth (ref. Fig. 6.b). When encoder and latent variable dimensions are 64 and 10 respectively, our model's NLL is ≤ 79.25. Comparison to [Standvoss et al., 2020] was not possible since they did not report prediction error or accuracy. Our NLL was lower when saliency was computed using KL divergence (ref. Eq. 3.2) as compared to rectification or Euclidean norm, as used in [Standvoss et al., 2020].

There are two key differences between the DRAW [Gregor et al., 2015] and our model:

(1) At any instant, prediction error of our model drives its attention (sampling location). In DRAW, attention weights are learned explicitly which are not driven by the model's prediction error alone.

(2) Our model considers the patch and its location as separate modalities, perception and proprioception respectively, resulting in two input modalities which are combined. DRAW considers the patch and its location together as input in one modality.

Our improvement in generation accuracy (NLL) suggests that these differences are playing an important role.

Ablation study. Here we evaluate the contribution of proprioceptive modality in our model. We define a variant of our model by eliminating the proprioceptive modality at input (observation) and output (generation), keeping rest of the model unchanged. That is, $\mathbf{x}_{<t} = \{x^{(1)}_{<t}\}$ and $i = 1$ in Eq. 3.4. For all datasets, the NLL is lower when the proprioceptive modality is used (ref. Table 5.). Thus, the proprioceptive modality facilitates more accurate pattern completion.

Intuitively, our agent senses its body (via proprioception) in addition to sensing its environment (via perception). This allows it to learn the relations between its perceptual and proprioceptive signals, which is the key to its accuracy. In recent years, artificial intelligence and related areas have been flooded with attention-based models for numerous applications. Our work is unique as it models action/attention as proprioception, similar to perception, and validates its role in attaining state-of-the-art accuracy.

Evaluation for efficiency. The number of trainable parameters increases exponentially with patch size (ref. Table 4.). The results in Table 6. are obtained with 5×5 patch size which allows our model to yield high accuracy while being efficient in model size.

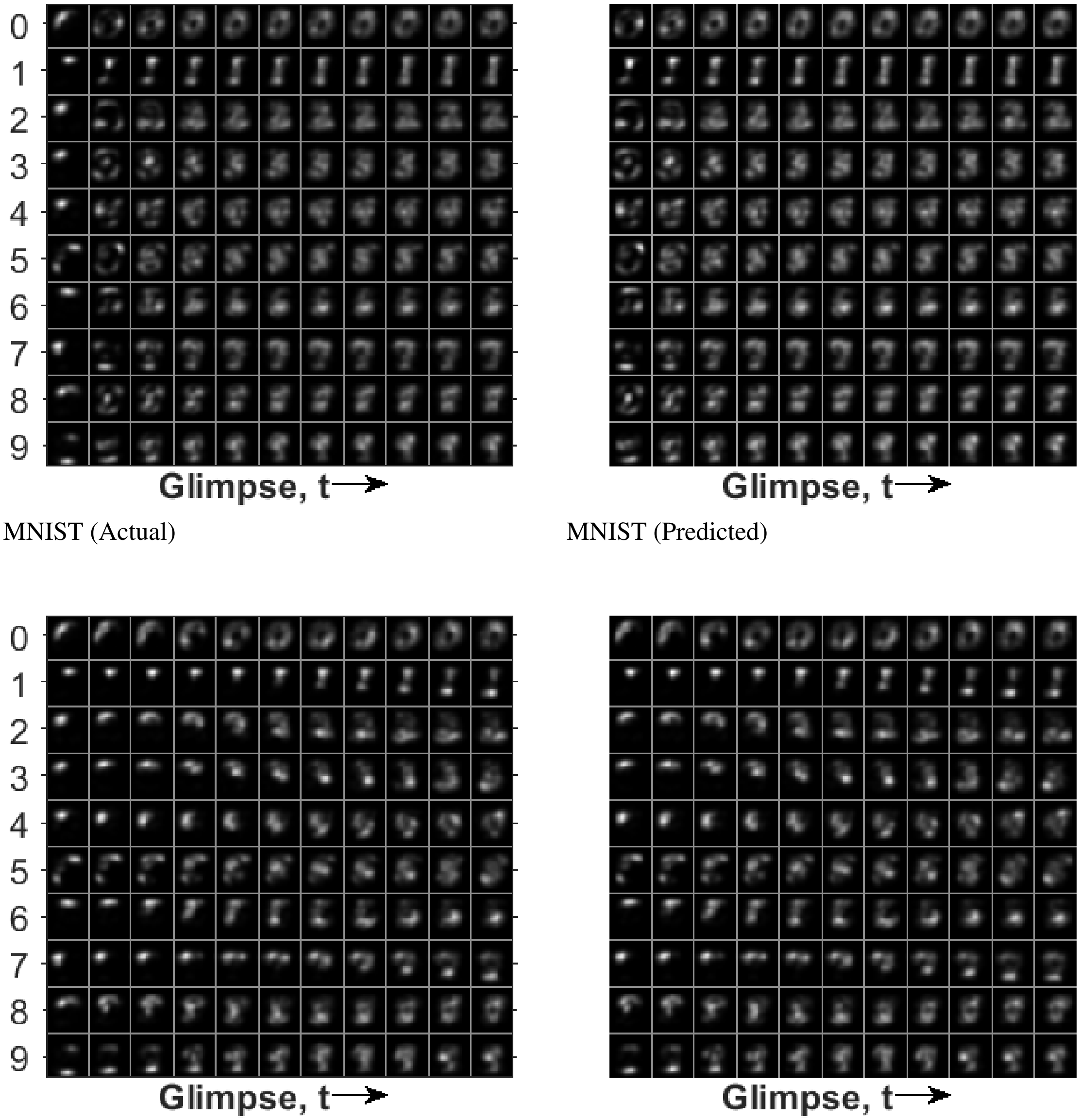
Our experiments show that prediction accuracy (NLL) improves exponentially with increase in number of glimpses and our model starts yielding high accuracy within a few glimpses (ref. Figs. 6.a, 6.c, 6.e). On average, by the fourth glimpse which corresponds to 11.2% (std. 1.1) of the scene, the prediction error drops by 60.4% (std. 10.1).

Figs. 6.b, 6.d, 6.f show that for 21×21 patch size, more than 80% of the scene is viewed within the sixth glimpse for all three datasets (MNIST, EMNIST, SMNIST). In contrast, for 5×5 patch size, less than 25% of the scene is viewed till the last (12^{th}) glimpse for all the datasets. However, the prediction accuracy for 5×5 is only slightly lower than that for 21×21 (ref. Figs. 6.a, 6.c, 6.e). This is because our model learns by sampling locations with maximum information gain at each glimpse. It yields state-of-the-art accuracy upon viewing only 22.6% of the scene on average over all three datasets (std. 2.7).

The proposed model yields state-of-the-art accuracy while being size and sample efficient. However, there is still room for improving its accuracy and efficiency. Our future work will include applying this model to other kinds of data, and learning this model using class labels in addition to perceptual and proprioceptive inputs.

3.4 Conclusions

A predictive agent model is proposed that sequentially samples and interacts with its environment. At each instant, it samples the location with maximum information gain to minimize its sensory prediction error in a greedy manner. The agent operates as a closed-loop system involving perceptual (‘what’) and proprioceptive (‘where’) pathways which are learned end-to-end, without supervision (class labels) or reinforcement. The same model can be used for static and dynamic environments. Experiments on handwriting generation reveal that the model is sample and size efficient, and yields state-of-the-art accuracy. Conceptually, this work is unique due to its modeling action/attention as proprioception, using it with perception in a multimodal setting, and experimentally validating its role in yielding state-of-the-art accuracy in an end-to-end model.



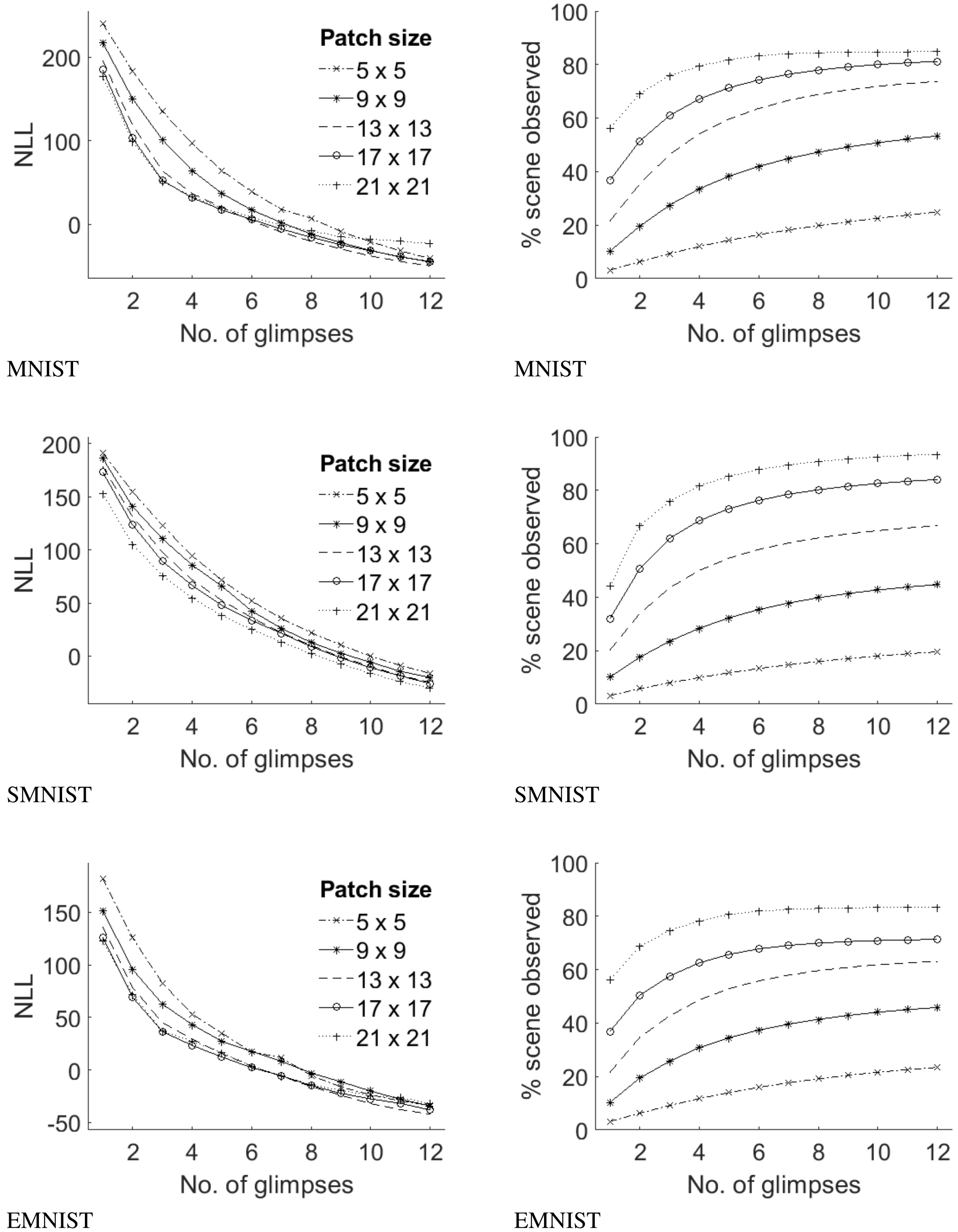
MNIST (Actual)

MNIST (Predicted)

SMNIST (Actual)

SMNIST (Predicted)

Figure 5.: Distribution of salient locations for MNIST and SMNIST datasets, averaged over all examples of a class in the test set. Actual salient locations are obtained from the saliency map and predicted salient locations are the predictions for visual proprioception. Each row and column correspond to a class and a glimpse number respectively.

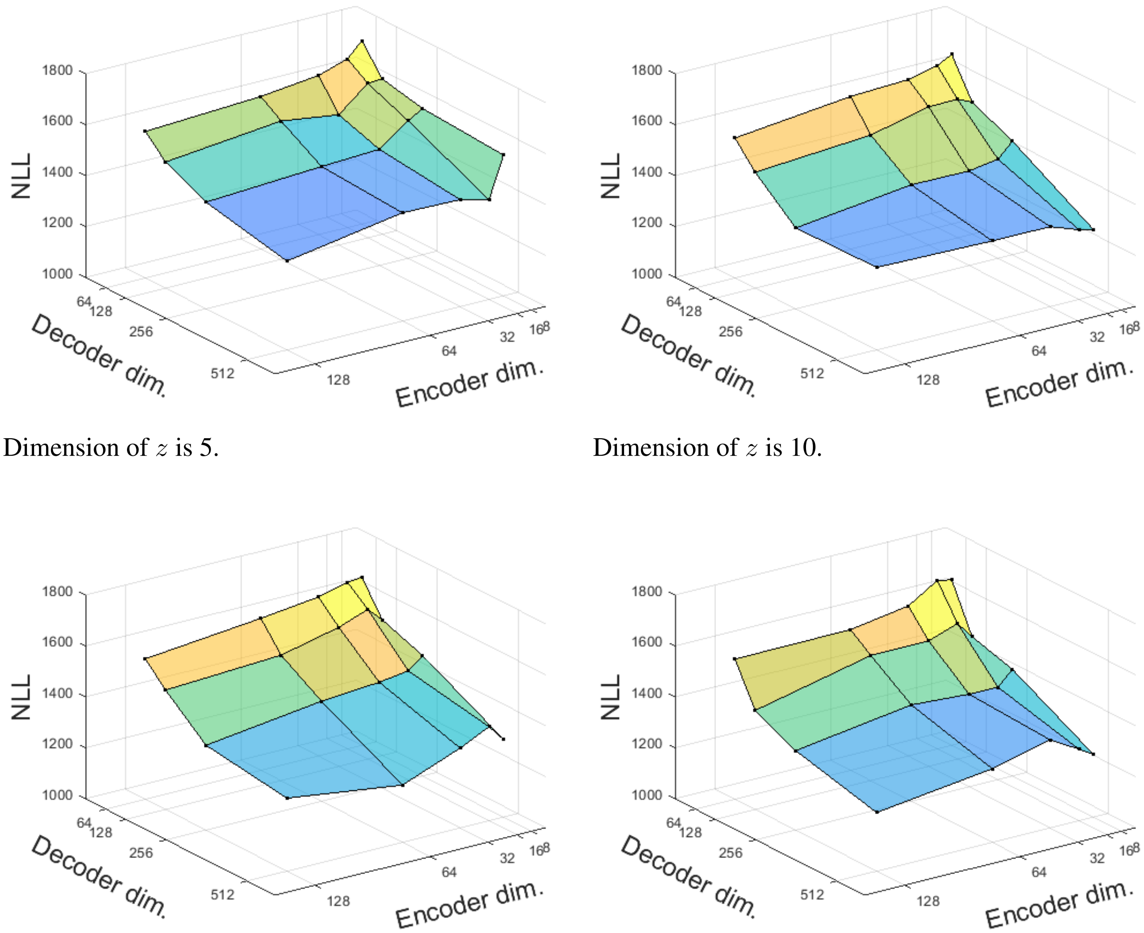


MNIST MNIST

SMNIST SMNIST

EMNIST EMNIST

Figure 6.: Comparison of prediction error (a, c, e) and efficiency (b, d, f) for different patch sizes.



Dimension of z is 5.

Dimension of z is 10.

Dimension of z is 20.

Dimension of z is 50.

Figure 7.: Prediction error (NLL) decreases up to a certain extent with increase in model size (i.e. encoder, decoder and latent variable (z) dimensions) for the MNIST dataset. In most cases, prediction does not improve beyond encoder dimension 64 and z dimension 10.

Chapter 4

A Dataset for Handwritten Numeral and Alphabet Recognition via Sequential Sampling

Abstract: Multiple attention-based models that recognize objects via a sequence of glimpses have reported results on handwritten numeral recognition. However, no eye-tracking data for handwritten numeral or alphabet recognition is available. Availability of fixation data would allow attention-based models to be evaluated in comparison to human performance. We collect mouse-click attention tracking (mcAT) data from 382 participants trying to recognize handwritten numerals and alphabets (upper and lowercase) from images via sequential sampling. Images from benchmark datasets are presented as stimuli. The collected data consists of a sequence of sample (click) locations, predicted class label(s) at each sampling, and the duration of each sampling. We show that on average, participants observe only 12.8% of an image for recognition. We propose a baseline model to predict the location and the class(es) a participant will select at the next sampling. When exposed to the same stimuli and experimental conditions as our participants, a highly-cited attention-based reinforcement model falls short of human efficiency.

4.1 Introduction

Machine learning (ML) models that recognize objects via a sequence of glimpses have gained interest in recent years due to their scalability and efficiency. Many of these models, such as [Ranzato, 2014, Ba et al., 2015, Mnih et al., 2014, Ba et al., 2014, Dutta and Banerjee, 2017, Larochelle and Hinton, 2010, Elsayed et al., 2019], have reported experimental results on the benchmark MNIST dataset for handwritten numeral recognition. Unfortunately, no attention tracking data for the MNIST is available. This prevents the evaluation of attention-based models in comparison to human performance.

We fill in that gap by collecting a dataset from individuals trying to recognize handwritten numerals and alphabets from images via sequential sampling. Unlike eye-tracking, the individual clicks the location in the image that he wants to see (a.k.a. *mouse-click attention tracking* (mcAT)). At the same time, he selects the class(es) that he predicts the object might belong to based on his observations till the current time. Thus, at each sampling, our data consists of the image location selected, class label(s) predicted, and time taken since last sampling by the individual. After each image, the individual receives a reward based on his performance (accuracy and efficiency).

Advantages of our data collection method over eye-tracking. (1) Eye movements contain significant variability, especially for static stimuli (images) [van Beers, 2007]. So a large amount of eye fixation data is needed to reach statistically significant conclusions. The variability is much lower in mcAT. **(2)** Eye movements are caused by a mixture of conscious and unconscious processing [Baumeister et al., 2011], including mind wandering [Smallwood and Schooler, 2015]. To facilitate task-dependent decision-making, we present the participants with adequate time, context and reinforcement signal, which can also be presented to an ML model. **(3)** The precision and accuracy of eye-tracking data are dependent on the eye-tracker while the same of our data are independent of any device. **(4)** It is a challenge to synchronize one's eye movements with his class selection. To overcome this, in our case the sampling location and class(es) are selected at the same time. **(5)** Finally, our method allows data collection using Amazon Mechanical Turk (MTurk), as in [Jiang et al., 2015, Kim et al., 2017], which is cost- and time-effective, and easily reproducible.

Related work. The temporal sequence of mouse clicks in mcAT is analogous to the eye movement scanpath [Egner et al., 2018]. mcAT can effectively substitute eye-movement attention tracking (emAT) as they are significantly correlated [Egner et al., 2000, Egner et al., 2018, Navalpakkam et al., 2013, Jiang et al., 2015, Kim et al., 2017, Matzen et al., 2021].

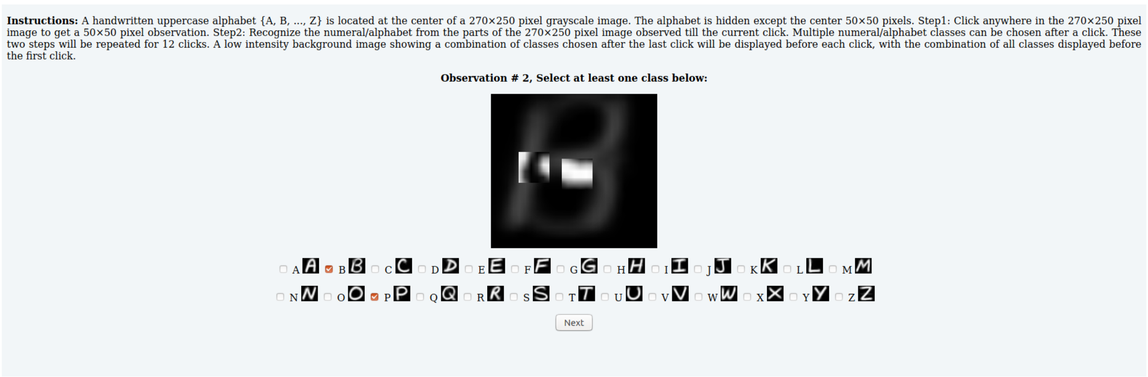
Different kinds of stimuli have been used in mcAT studies, such as images of animate and inanimate objects [Egner et al., 2018], images of natural scenes [Jiang et al., 2015, Kim et al., 2017], static webpages [Kim et al., 2017], search page layouts [Navalpakkam et al., 2013], and two lists of alphanumeric strings for visual comparison [Matzen et al., 2021]. However, mcAT has not been used for handwritten numeral/alphabet classification tasks or evaluation of attention-based classification models.

mcAT studies have used features such as time to contact, relative fixation frequency in areas of interest (AOIs), and relative proportion of subjects that clicked at least once in an AOI [Egner et al., 2018], number of fixations per trial, refixations within trials, dwell times, and scanpaths [Matzen et al., 2021], fixation maps [Kim et al., 2017, Jiang et al., 2015], AOI and information flow pattern [Navalpakkam et al., 2013]. The sequence of time-stamped click locations and predicted class labels constitute the raw data necessary to evaluate the efficiency and accuracy of attention-based models or humans in classification tasks. Different features can be derived from this data.

Contributions. We collect an mcAT dataset using MTurk from 382 participants, rewarded for accurately and efficiently recognizing handwritten numerals and alphabets (upper and lowercase) from images via sequential sampling. Images from benchmark datasets (MNIST, EMNIST) are presented as stimuli. On average, 169.1 responses per numeral/alphabet class are recorded. Using this dataset, we show the following:

1. On average, participants require 4.2, 4.7 and 4.9 samples to recognize a numeral, uppercase and lowercase alphabet, which correspond to only 11.3%, 13.4% and 13.7% of image area respectively. Classification accuracy increases with samples.

2. A model, presented as the baseline, can predict the class(es) and location a participant will select at the next sampling instant with 74.4% and 67.7% accuracy respectively, both averaged over all samplings and datasets. Class prediction accuracy increases and location prediction accuracy decreases with increase in samples.



Instructions: A handwritten uppercase alphabet {A, B, ..., Z} is located at the center of a 270×250 pixel grayscale image. The alphabet is hidden except the center 50×50 pixels. Step1: Click anywhere in the 270×250 pixel image to get a 50×50 pixel observation. Step2: Recognize the numeral/alphabet from the parts of the 270×250 pixel image observed till the current click. Multiple numeral/alphabet classes can be chosen after a click. These two steps will be repeated for 12 clicks. A low intensity background image showing a combination of classes chosen after the last click will be displayed before each click, with the combination of all classes displayed before the first click.

Observation # 2, Select at least one class below:

□ A ☑ B □ C □ D □ E □ F □ G □ H □ I □ J □ K □ L □ M

□ N □ O ☑ P □ Q □ R □ S □ T □ U □ V □ W □ X □ Y □ Z

Figure 8.: Our MTurk interface as seen by a participant. The second sampling for an EMNIST uppercase alphabet is shown.

3. When exposed to the same stimuli and conditions as our participants, a highly-cited reinforcement-based recurrent attention model (RAM) [Mnih et al., 2014] requires 3.7, 8.5, 7.6 samples to recognize a numeral, uppercase and lowercase alphabet, which correspond to 8.9%, 21.0%, 18.7% of image area respectively. Other attention-based reinforcement models (e.g., [Ranzato, 2014, Ba et al., 2015, Ba et al., 2014, Sermanet et al., 2014, Dutta and Banerjee, 2017, Elsayed et al., 2019]) can be similarly evaluated in comparison to human performance.

4.2 Data

Our data consists of a sequence of time-stamped samples. Each sample consists of: (1) the location in the image selected by the participant, (2) the class(es) selected by the participant, and (3) the time taken by the participant to register the current sample (i.e. the time elapsed between registering the last and current samples). This section will explicate our data collection process that includes stimuli selection, participants, visual task, performance scoring, and data filtering.

4.2.1 Stimuli selection

Stimuli are selected from images in two benchmark datasets, MNIST and EMNIST.

MNIST [LeCun et al., 1998] dataset consists of $70{,}000$ labeled images (28×28 pixels) of 10 handwritten numerals $\{0, 1, \ldots, 9\}$.

EMNIST [Cohen et al., 2017] dataset consists of $145{,}600$ images (28×28 pixels) of handwritten English alphabets in uppercase and lowercase, forming a balanced class. All images are labeled with one of 26 classes $\{a, b, \ldots, z\}$. However, uppercase or lowercase label is not associated with any image.

From each category, we select 15 well-formed numerals from MNIST and 15 well-formed alphabets each from EMNIST uppercase and EMNIST lowercase datasets. A well-formed numeral or alphabet is one that is closer to the norm of its class but might have some variation. Thus, we present stimuli from a set of $15(10 + 26 + 26) = 930$ unique images, with 15 images belonging to each of the 62 classes.

The well-formed 930 images are selected as follows:

Step 1: Normalize each image using min-max to scale the intensity between 0 and 1.

Step 2: Label well-formed EMNIST images as uppercase or lowercase. For each alphabet class, a well-formed alphabet from both uppercase and lowercase images is manually selected and labeled. The cosine similarity of all images belonging to that class with the two labeled images is computed. The images that are above the cosine similarity threshold (empirically chosen as 0.8) are assigned the uppercase or lowercase label.

Step 3: Compute the mean of the images belonging to each class. The mean image of a class constitutes its norm. An image is eligible to be a stimulus if its cosine similarity with the mean image of its class is greater than an empirically-determined threshold (0.7 for MNIST, 0.75 for EMNIST).

Step 4: Among the eligible images, 15 images from each class are selected manually based on how well-formed it is.

Each image, originally 28×28 pixels, is reduced to 27×25 by removing the pixels near the boundaries as they have no intensity variation. The mean of these 15 images is computed for each of the 62 classes. We denote these mean images as $I_1, I_2, \ldots, I_n$ for n classes in each dataset.

4.2.2 Participants

A total of 382 distinct adult individuals participated in our study. No selection criteria were used. A participant could respond to multiple images. For each of the 62 classes, an average of 169.1 responses were recorded.

4.2.3 Visual task

The MTurk interface for our visual task is shown in Fig. 8.. A canvas of size 270×250 displays a low-intensity background image at all time. The background and stimulus images are upsampled to 270×250. The center of the canvas is aligned with the center of the images.

Background. Initially, the background is the mean of all images in the dataset from which the stimulus is drawn. After the first sampling, the background is the mean of all images from the set of classes selected by the participant in the last sampling. In the real world, the context for location, size and orientation of a numeral or alphabet is obtained from the writing in its neighborhood, which is missing here. When our experiments were done with a blank background, the participants often sampled multiple locations of the image that do not contain any part of the object. This behavior was contained by presenting the mean image of the selected class(es) in a low-intensity background and reducing the size of all MNIST and EMNIST images from 28×28 pixels to 27×25.

Each time the participant selects a location in the canvas by clicking on it, a 50×50 pixel patch centered at that location from the stimulus image is revealed. A patch once revealed continues to be displayed till the final sampling.

A participant's task consists of three steps at each sampling, $t = 1, \ldots, T$:

Step 1: Click anywhere in the 270×250 canvas to reveal the patch he wants to sample. Only the first click is accepted.

Step 2: Recognize the numeral/alphabet from all the samples till the current time.

The participant can select multiple classes and will have to choose at least one class from the list of classes shown below the canvas.

Step 3: Click “Next” at the bottom of the screen to proceed.

In order to infer the class accurately and quickly, the participant will have to choose the locations judiciously given his observations till the current time. There is no time limit for a sampling. However, we limit the total time for T samplings of an image to six minutes.

4.2.4 Performance scoring

A score is assigned to the participant based on his accuracy and efficiency in terms of the number of samples observed. Let c_t be the set of classes he chose at any sampling t. Then, his score at t is:

$$P_t = \begin{cases} \frac{1}{|c_t|}, & \text{if correct class} \in c_t \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where $|.|$ denotes the cardinality of a set. Total score awarded in T samplings is: $h = \sum_{t=1}^{T} P_t$. Therefore, the maximum one can score in T samplings is T if he always chooses only the correct class. The minimum one can score in T samplings is zero if he always chooses a set of classes that does not include the correct class. So, $0 \leq h \leq T$.

Sooner a participant selects the correct class, the higher his score will be. Thus, this scoring mechanism takes into account recognition accuracy and sampling efficiency. Trying to maximize score by choosing only one class from the very first sampling will be risky as a score of zero will be awarded if it is not the correct class, whereas a score greater than zero will be awarded if the participant chooses multiple classes (even all classes) that include the correct class. This will motivate the participant to respond based on the probable classes in his mind at any sampling. The score awarded at each sampling instant is disclosed only at the end of T sampling instants to refrain from providing any hint to the participant. In MTurk, the remuneration received by a participant for an image is proportional to his total score, h.

4.2.5 Data filtering

If a participant's score at the final (i.e. T-th) sampling for a stimulus image is zero, his data recorded for that image is discarded. The data is also discarded if a participant leaves the task incomplete. With this selection criteria, we obtained responses on 1736 stimuli from MNIST, 4431 stimuli from EMNIST uppercase, and 4315 stimuli from EMNIST lowercase; that is, 169.1 responses per class on average.

4.3 Models and Methods for Utilizing Data

In this section, we illustrate the utility of the collected data by: (4.3.1) providing a baseline model for predicting the behavior of a participant, and (4.3.2) showing how an existing attention-based reinforcement model can be compared to human numeral/alphabet recognition performance.

4.3.1 Baseline for behavior prediction

Behavior at any sampling instant t consists of location selection and class selection. Let n be the number of classes in a dataset, η_t be the singleton set containing the true class for the stimulus image at t, c_t be the set of classes and l_t be the location selected by a participant at t, o_t be his observation at t, and $1{:}t$ denotes the sequence $1, 2, \ldots, t$. Till any t, the observations of a participant are $o_{1:t}$ and the locations he selected are $l_{1:t}$.

We formulate the problem of a participant's behavior prediction as follows:

Class prediction. Estimate the probability of $i \in c_t$ ($i = 1, 2, \ldots, n$) given his $o_{1:t}$ and $l_{1:t}$, i.e. $P(i \in c_t | o_{1:t}, l_{1:t})$.

Location prediction. Estimate the probability of l_{t+1} given his $o_{1:t}$, $l_{1:t}$ and c_t, i.e. $P(l_{t+1} | o_{1:t}, l_{1:t}, c_t)$.

Class prediction

To predict the class a participant will choose at sampling t, we compute the probability that the image stimulus at t belongs to class i given the participant's selected locations $l_{1:t}$ and the corresponding observations $o_{1:t}$, as follows:

$$P(i|o_{1:t}, l_{1:t}) = \frac{\frac{I'}{\|I'\|} \cdot \frac{I_i}{\|I_i\|}}{\sum_{j \in \{1,\ldots,n\}} \frac{I'}{\|I'\|} \cdot \frac{I_j}{\|I_j\|}} \tag{4.2}$$

where I_i is the mean of the stimuli images (27×25) belonging to class i, I' is a 27×25 image containing $o_{1:t}$ at $l_{1:t}$, $\cdot$ denotes scalar product, and $\|.\|$ denotes Euclidean norm. All pixel intensities are non-negative.

At any sampling t, the k highest probable classes from the belief distribution $P(i|o_{1:t}, l_{1:t})$ constitute the set of classes, $\hat{c}_t$, predicted by our model, where $k = |c_t|$.

The classification accuracy is measured using the Jaccard index (JI). JI measures the similarity between two sets, X and Y, as: $J(X, Y) = |X \cap Y|/|X \cup Y|$. JI is bounded between 0 and 1; if $X = Y$, $J(X, Y) = 1$. At any sampling t, the classification accuracy of a participant is $J(\eta_t, c_t)$ while that of our model is $J(\eta_t, \hat{c}_t)$. Due to its denominator, JI penalizes more as the number of elements in the predicted set (c_t or $\hat{c}_t$) that are not in η_t increases, which is a desirable property for our case. The similarity between a participant's and our model's classification is measured by $J(c_t, \hat{c}_t)$.

Our model is also evaluated in terms of class selection and rejection accuracy with respect to each participant. Let $s_t = c_t - c_{t-1}$ be the set of new classes selected and $r_t = c_{t-1} - c_t$ be the set of classes rejected by a participant at t. Similarly, $\hat{s}_t = \hat{c}_t - c_{t-1}$ be the set of new classes selected and $\hat{r}_t = c_{t-1} - \hat{c}_t$ be the set of classes rejected by our model at t. Then the model's class selection and rejection can be compared to a participant's by $J(s_t, \hat{s}_t)$ when $|s_t| > 0$ and $J(r_t, \hat{r}_t)$ when $|r_t| > 0$, respectively.

Location prediction

Hypothesis. Ideally, the belief distribution over all classes should be unimodal (i.e., one peak only) and a thin Gaussian (i.e., small standard deviation) in shape indicating a participant is confident about the class (state) of the stimulus (environment). However, as evident from our data (ref. Fig. 10.), a participant is often confused between multiple classes, especially during the initial few sampling instants. In these cases, his belief distribution has multiple peaks or is a fat Gaussian. We hypothesize, a participant's goal is to converge to a unimodal and thin Gaussian to achieve which it selectively samples locations that reduce the probability of all classes except one. This hypothesis leads to minimization of uncertainty over the classes (environmental states) which is a well-known principle guiding brain function (see [Friston, 2009]).

The observations at certain locations in a stimulus image can discriminate between certain classes. The observation at a location l might indicate that the numeral/alphabet belongs to a class i and not to a class j. Such locations are more salient than others in achieving a participant's goal. To sample such locations, a saliency map, D_{ij}, is computed such that if l is salient, the observation at l is an evidence to increase the probability of class i and decrease that of j.

Mathematically, $D_{ij} = \mathcal{N}(., \sigma) * g(.)$, where $*$ is the convolution operator, $g(.)$ is a saliency scoring function, and $\mathcal{N}(., \sigma)$ is a 5×5 Gaussian kernel with standard deviation $\sigma = 6$ to smooth the saliency scores. We denote the set of all saliency maps as $\mathrm{D} = \{D_{ij} : i, j \in \{1, 2, \ldots, n\}, i \neq j\}$. A location l in a stimulus image is salient for class i with respect to class j if $D_{ij}(l) > \theta$, where the threshold $\theta = 0.5 \times \max(\mathrm{D})$ is an empirically determined scalar quantity.

We consider two asymmetric metrics, Kullback-Leibler (KL) divergence and difference, as candidates for the function g.

KL divergence. Given two normalized mean images, I_i and I_j, the KL divergence $KL(I_i, I_j)$ measures the loss of information when I_j is used to approximate I_i. This is calculated for each pixel k as [Bylinskii et al., 2018]: $KL(I_{i,k}, I_{j,k}) = I_{i,k} \log\left(\epsilon + \frac{I_{i,k}}{I_{j,k}+\epsilon}\right)$, where ϵ is a regularization constant. Lower KL divergence for k implies $I_{i,k}$ and $I_{j,k}$ are similar.

Difference. Given two normalized mean images, I_i and I_j, the difference for each

Table 7.: Average Pearson correlation coefficient (corr.) for fixation sequences for the same class. For any fixation, distance is Euclidean and direction is measured as the polar angle with respect to the center of stimuli as origin. Standard deviations are included in parenthesis.

Metric	MNIST	EMNIST upp.	EMNIST low.
Distance corr.	0.34 (0.21)	0.42 (0.22)	0.33 (0.21)
Direction corr.	0.27 (0.19)	0.28 (0.21)	0.29 (0.2)

Table 8.: Evaluation of fixation maps from RAM for the stimuli presented in the MTurk experiments, averaged over all classes and samplings. Standard deviations are included in parenthesis.

Metric	MNIST	EMNIST upp.	EMNIST low.
KL	22.50(±7.48)	22.96(±7.24)	22.23(±7.16)
CC	0.01(±0.00)	0.01(±0.00)	0.01(±0.00)
SIM	0.17(±0.09)	0.16(±0.07)	0.18(±0.09)

pixel k is: $\mathit{Diff}(I_{i,k}, I_{j,k}) = I_{i,k} - I_{j,k}$. Lower difference for k implies $I_{i,k}$ and $I_{i,k}$ are similar.

A participant is uncertain regarding the set of classes, c_t, he selected at the current sampling. Hence, for location prediction, we consider only those saliency maps in D that involve the classes in c_t. A location is predicted if it is salient based on these saliency maps and was never selected by the participant. Thus, given $o_{1:t}$, $l_{1:t}$ and c_t, the location l_{t+1} is predicted as follows:

$$\begin{aligned} \mathrm{D}' &= \{D_{ij} : D_{ij} \in \mathrm{D}, i \in c_t \text{ or } j \in c_t\} \\ \Gamma &= \{\langle \hat{l}, i, j \rangle : \hat{l} \notin l_{1:t}, D_{ij}(\hat{l}) > \theta, D_{ij} \in \mathrm{D}'\} \end{aligned} \tag{4.3}$$

where Γ is the set of 3-tuples containing the predicted location $\hat{l}$, the class it is salient for (i), and with respect to which class (j). The location is predicted correctly if there exists a $\langle \hat{l}, i, j \rangle \in \Gamma$ such that $\|\hat{l} - l_{t+1}\| < \epsilon$, $i \in c_{t+1}$ and $j \notin c_{t+1}$, where ϵ is the maximum Euclidean distance between the center pixel and any pixel in an observation patch. A pseudo code for the location prediction algorithm is included in the supplemental material.[1]

[1] The probability distribution, $P(l_{t+1}|o_{1:t}, l_{1:t}, c_t)$, may be computed by assuming the saliency score of locations not in Γ to be zero, and then normalizing the saliency score of all locations to sum to unity. However, this probability has not been used, as eq. 4.3 is sufficient for the purposes of this paper.

4.3.2 Evaluation of attention-based models

As a representative of attention-based models, we consider the highly-cited recurrent attention model (RAM) [Mnih et al., 2014] that reports experimental results on the MNIST dataset. This reinforcement model sequentially samples an image and decides where to sample next at each sampling instant, making it appropriate for evaluation using the collected data.

RAM classifies images using a sequence of glimpses. The next location is chosen stochastically from a distribution parameterized by a location network. The model is trained end-to-end by maximizing the following objective [Mnih et al., 2014]:

$$\frac{1}{M}\sum_{i=1}^{M}\sum_{t=1}^{T}\Delta_\theta \log \pi(u_t^i|x_{1:t}^i;\theta)(R_t^i - b_t) \tag{4.4}$$

where M is the number of episodes, T is the number of observations, $x_{1:t}^i$ are the interaction sequences obtained by running the current agent till i episodes, u_t^i is the current action, θ is the set of trainable parameters, R_t^i is the cumulative reward, b_t is a baseline, and $\pi(u_t^i|x_{1:t}^i;\theta)$ is the policy. RAM's behavior may be compared with the participants' by comparing the fixation maps obtained from the sequence of locations predicted by RAM and those chosen by the participants. A fixation map is computed by assigning each location a value equal to the frequency of its selection, and then normalizing those values to create a distribution over all locations.

Metrics for comparing fixation maps

For metrics comparing two fixation maps, P and Q, we closely follow [Bylinskii et al., 2018]. We use three distribution-based metrics: KL divergence (KL), Pearson correlation coefficient (CC), and Similarity (SIM), to compare the distribution of sampling locations from a model with that from the participants as recorded in the collected data.

KL (defined earlier) is highly sensitive to zero values.

CC can evaluate the linear relationship between two maps as [Bylinskii et al., 2018]: $CC(P,Q) = \frac{\sigma(P,Q)}{\sigma(P)\sigma(Q)}$, where σ is the variance or covariance. Since CC is symmetric, it fails to infer whether differences between fixation maps are due to false positives or false negatives.

SIM is measured as [Bylinskii et al., 2018]: $SIM(P,Q) = \sum_k \min(P_k, Q_k)$, where $\sum_k P_k = \sum_k Q_k = 1$. Like CC, SIM is symmetric and inherits the same drawback. Also, SIM is very sensitive to missing values, and penalizes predictions that fail to account for the ground truth density.

4.4 Experimental Results

4.4.1 Data analysis

The collected data can be visualized in terms of the sequence of distribution of selected locations (Fig. 9.), selected classes (Fig. 10.), and duration between consecutive samplings (Fig. 10.). These distributions are very similar for the three datasets.

For any numeral or alphabet, the distribution of selected locations after the final sampling resembles the distribution of pixel intensities of its class from the dataset. However, the sequence of locations selected is stochastic in nature.

The class distribution indicates confusion between categories with similar structures at the initial few samplings when the participants choose multiple classes. This confusion reduces with more sampling. There is a significant positive correlation between degree of confusion (# selected classes/total # classes) and sampling duration (see Fig. 3 in supplemental material). If the number of selected classes is high (low), the duration between consecutive samplings is high (low).

The CC of the sequence of locations selected by a participant for a class is not significant (Table 7.). This is expected due to inter-subject variability in sampling static images.

The average number of samplings required by a participant to accurately predict a class is quite low. On average, it takes 4.2, 4.7, 4.9 samples corresponding to 36, 44.1, 48.1 seconds to accurately classify MNIST, EMNIST uppercase and lowercase images respectively. The participants on average viewed only 11.3%, 13.4%, 13.7% of image area for classifying a numeral, uppercase and lowercase alphabet image accurately (see Fig. 2 in supplemental material). These results highlight the efficiency of the human visual reasoning system, albeit at a lower resolution than eye tracking data but with less noise and variability. These empirical results may be useful for designing attention-based models for real-world applications.

4.4.2 Behavior prediction

In this section, the performance of our baseline model is evaluated in terms of how accurately it can predict each participant's location and class selection. Since our experimental results using the two saliency scoring functions, KL divergence and difference, are quite similar, results are reported using difference only, unless otherwise stated.

Class prediction

The class prediction and its accuracy evaluation methods are described in Section 4.3.1. The class prediction accuracy, shown in Fig. 11., is computed over all classes for all samplings. The mean class prediction accuracy over all samplings and datasets is 74.4% (standard deviation 26.5).

Figs. 11.a, 11.b show that the set of classes selected by the participants and by our baseline model (eq. 4.2) are quite inaccurate at the initial sampling instants and improves with increase in samples. Fig. 11.c shows that, during the initial samplings, these two sets, c_t and $\hat{c}_t$, are quite dissimilar; similarity increases with increase in samples. The same applies to new class selections (ref. 11.f). However, class rejections are similar at the initial samplings; similarity increases further with more samples (ref. 11.e). Since $J(s_t, \hat{s}_t) = \frac{|(c_t \cap \hat{c}_t) - c_{t-1}|}{|(c_t \cup \hat{c}_t) - c_{t-1}|}$ and $J(r_t, \hat{r}_t) = \frac{|c_{t-1} - (c_t \cup \hat{c}_t)|}{|c_{t-1} - (c_t \cap \hat{c}_t)|}$, it can be inferred from Figs. 11.e, 11.f that at the initial samplings, the intersection between c_{t-1} and $c_t \cup \hat{c}_t$ is small, indicating that initially the participants and our baseline model make many changes in their class selection between consecutive samplings. Therefore, initially, the class selection process is highly stochastic.

While there are some dissimilarities between the participants' and our model's class prediction during the initial samplings, the behaviors become increasingly similar with more samples. During the first few (typically 4 to 7) sampling instants, highly salient parts of a stimulus are revealed. This helps to select only the correct class in the later samplings, which increases the prediction accuracy. Since there are many classes whose mean templates match the observed parts of the stimulus during the initial few instants, the class selection process is significantly more stochastic, leading to low classification accuracy from the participants as well as our model.

Location prediction

Our baseline model's (eq. 4.3) location prediction accuracy, averaged over all samplings and datasets, is 67.7% (standard deviation 14.1) (ref. Fig. 11.d). The trend of this prediction accuracy is opposite to that of class prediction accuracy. However, the explanation remains the same. Location prediction accuracy is high during the initial samplings because during these instants, the highly salient locations are selected, leaving the less salient locations to be selected in the later instants. Since there are many locations with low saliency, their selection process is highly stochastic and hence difficult to predict, leading to a decrease in prediction accuracy with increase in samplings. The decreasing trend is unique for each dataset (ref. Fig. 11.d) as the number of classes and the number of highly salient locations useful for discrimination vary between datasets. Lower the number of classes and highly salient discriminative locations, faster will be the decrease in location prediction accuracy with increase in samplings.

4.4.3 Evaluation of RAM

For each class and sampling, the fixation maps from RAM[2] and the collected data for the same stimuli presented in MTurk are compared. For a fair comparison with the

[2] We used the RAM implementation from github.com/hehefan/Recurrent-Attention-Model.

participants, in RAM we fixed the sequence length at $T = 12$, the first sampling location at the image center, the input observation to a 5×5 patch with the selected location as its center, and modified the reward function by eq. 4.1. The cumulative reward, R_t in eq. 4.4, is replaced by the cumulative score $\sum_{\tau=1}^{t} P_\tau$ obtained from eq. 4.1. As a participant can select multiple classes at any instant, for the RAM model, instead of predicting a single class based on highest probability, we consider the mean probability over all classes as a threshold and predict the set of classes c_t with probabilities greater than the threshold. This c_t is used for calculating the score using eq. 4.1.

Under these conditions, RAM requires 3.7, 8.5, 7.6 samples to recognize MNIST numerals, uppercase and lowercase EMNIST alphabets, which correspond to 8.9%, 21.0%, 18.7% of image area respectively. Thus, in comparison to our participants (ref. Section 4.4.1), RAM is less efficient.

Results from comparing the fixation maps from RAM and the collected data are shown in Table 8.. KL is higher due to its sensitivity to zero values. This implies several locations are sampled by the participants but not by RAM. These experiments can be used as a baseline for evaluating locations sampled by an attention model.

4.5 Conclusions

We introduced an mcAT dataset for recognizing handwritten numerals and alphabets via sequential sampling. The data is collected from 382 participants presented with images selected from benchmark datasets (MNIST, EMNIST). On average, 169.1 responses per numeral/alphabet class are recorded. The data is rigorously analyzed to reveal the efficiency of human visual recognition. The participants observed only 12.8% of an image for recognition. We proposed a baseline model to predict the location and class(es) a participant would select at the next sampling. We showed how our experimental conditions and data may be used to evaluate an attention-based reinforcement model in comparison to human performance. This mcAT dataset, with multiple benefits over eye-tracking data, fills an important gap in attention-based models research.

4.6 Supplemental Material

4.6.1 Location prediction pseudocode

As stated in Section 4.3.1 in our paper, we implement our hypothesis for location prediction as Algorithm 4.

According to our hypothesis, a participant is minimizing uncertainty from his current selection of classes, c_t. Initially, the saliency maps that involve the classes in c_t are selected (ref. line 4 in Algorithm 4).

To conform with our MTurk experimental setup, inhibition-of-return is utilized in the baseline model. That is, a location cannot be resampled if it has already been sampled (ref. line 6 in Algorithm 4). This is achieved by assigning a negative value to a 5×5 patch centering the previously sampled location.

A location(s) in each saliency map with saliency value greater than a threshold θ is selected (ref. line 9 in Algorithm 4) which is added to the set of 3-tuples Γ along with the classes involved in the current saliency map (ref. line 10 in Algorithm 4) .

In order to ensure this location is not resampled within the loop at the same sampling instant, inhibition-of-return is applied (ref. line 11 in Algorithm 4) before selecting the next location from that saliency map. This is done by lowering the saliency values from a 9×9 patch centering that location. Setting a global threshold allows comparison of all the saliency maps and allows selection of variable number of locations from each saliency map. The average number of locations selected from a saliency map is shown in Fig. 12.. This number is less than three in all cases.

If the participant selects a location which lies close to the predicted salient location $\hat{l} = \arg\max(D_{ij})$, and he selects class i and does not select class j, the predicted location is considered to be correct. The Euclidean distance between the predicted location and participant's selected location has to be less than $\sqrt{8}$, which is the maximum distance between the center pixel and any other pixel of a 5×5 patch.

Algorithm 4 PredictLocation(D, $\theta, l_{1:t}, c_t, l_{t+1}, c_{t+1}$)

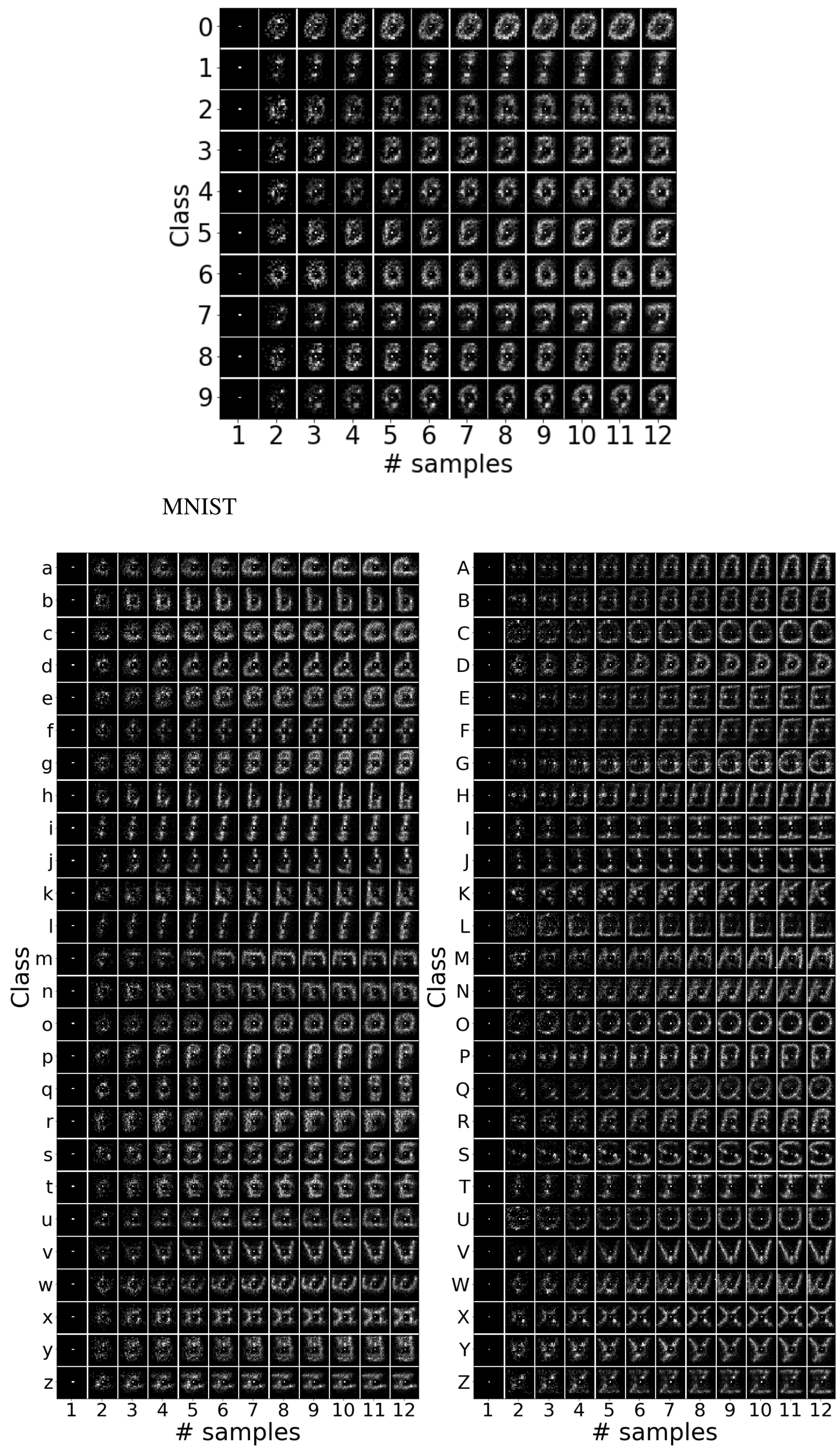
```
Initialize set of 3-tuples $\Gamma = \{\}$. $n$ = # classes in the dataset. Note that $l_{t+1}, c_{t+1}$ are not needed for location prediction. They are needed only for verifying if the location is predicted correctly.
for $i \leftarrow 1$ $to$ $n$ do
    for $j \leftarrow 1$ $to$ $n$ do
        if $i \in c_t$ or $j \in c_t$ then
            for $k \leftarrow 1$ $to$ $t$ do
                $D_{ij}[l_k - 2 : l_k + 2] \leftarrow -1$    //Inhibition of return
            end for
            while $\max(D_{ij}) > \theta$ do
                $\hat{l} \leftarrow \arg\max(D_{ij})$    //Predicted location
                $\Gamma \leftarrow \Gamma \cup \{\langle \hat{l}, i, j \rangle\}$
                $D_{ij}[\hat{l} - 4 : \hat{l} + 4] \leftarrow D_{ij}[\hat{l} - 4 : \hat{l} + 4] - \mathcal{N}(., \sigma)$    //Inhibition of return, $\mathcal{N}(., \sigma)$ is a $9 \times 9$ Gaussian kernel, $\sigma = 2$.
                if $\|\hat{l} - l_{t+1}\| < \sqrt{8}$ and $i \in c_{t+1}$ and $j \notin c_{t+1}$ then
                    Location is predicted correctly.
                end if
            end while
        end if
    end for
end for
```

4.6.2 Experimental Results: Data analysis

Detailed analysis of the participants' class selection for each dataset is shown in Fig. 13..

Fig. 14. shows the positive correlation between sampling duration and the degree of confusion for class selection by participants.

4.6.3 Data samples

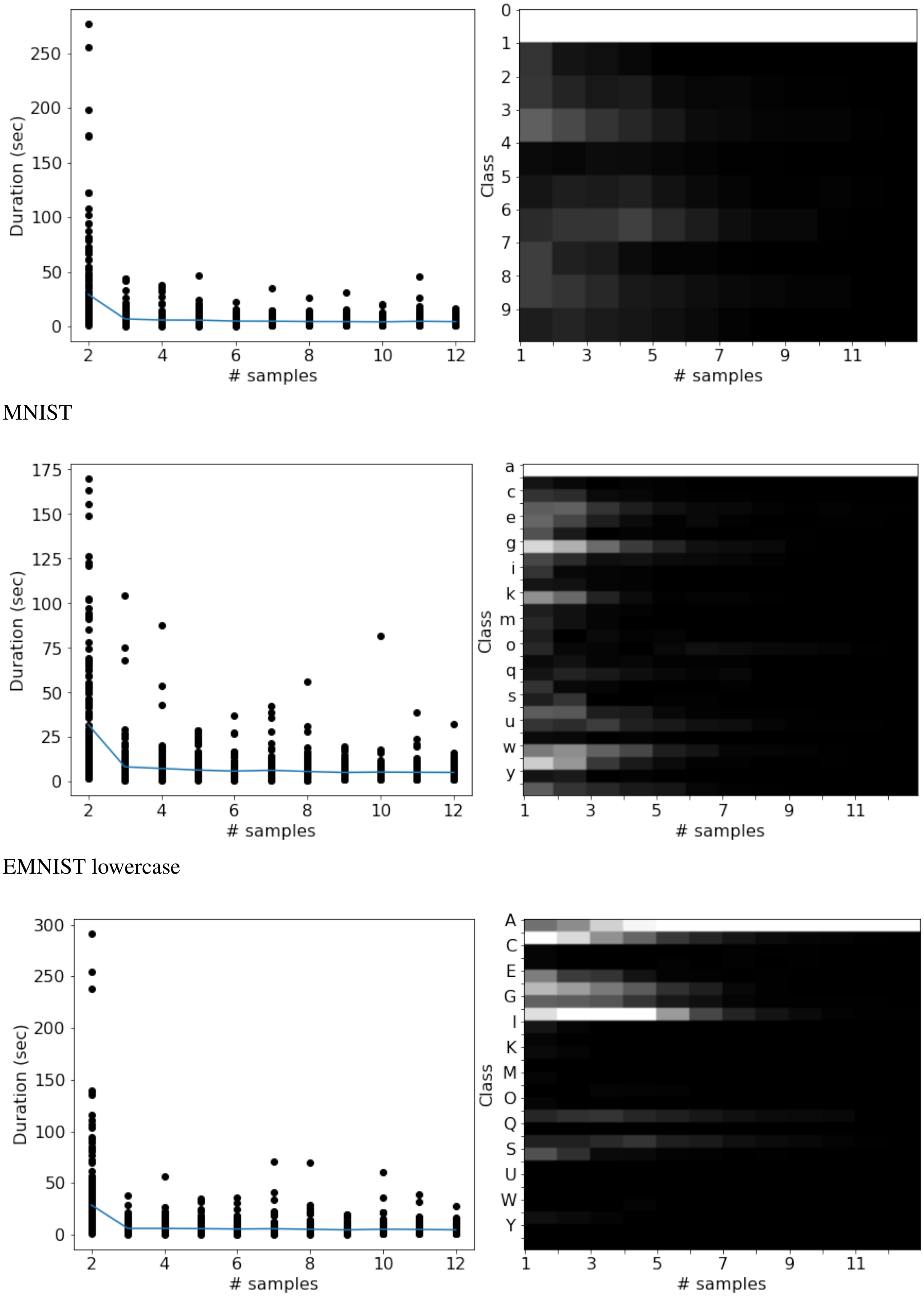


MNIST

EMNIST lowercase

EMNIST uppercase

Figure 9.: Distribution of sampling locations over all participants for each numeral/alphabet class and each sampling instant. Each row corresponds to a class, each column corresponds to a sampling instant which increases from left to right.



MNIST

EMNIST lowercase

EMNIST uppercase

Figure 10.: Duration and class distribution over all participants and stimuli belonging to categories ‘0’, ‘a’ and ‘A’.

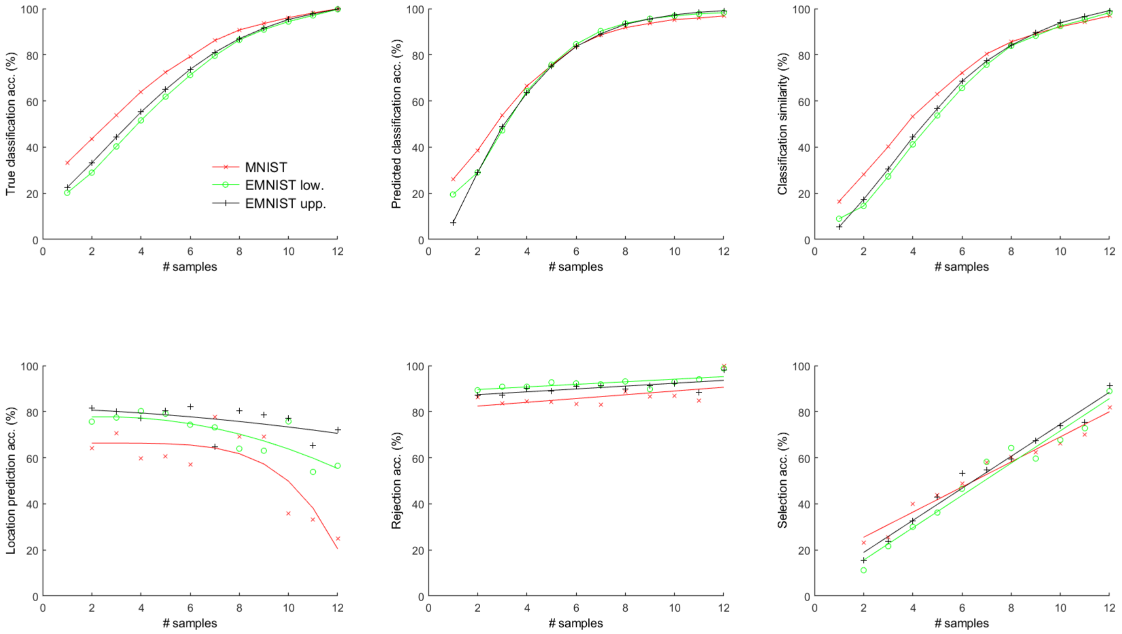
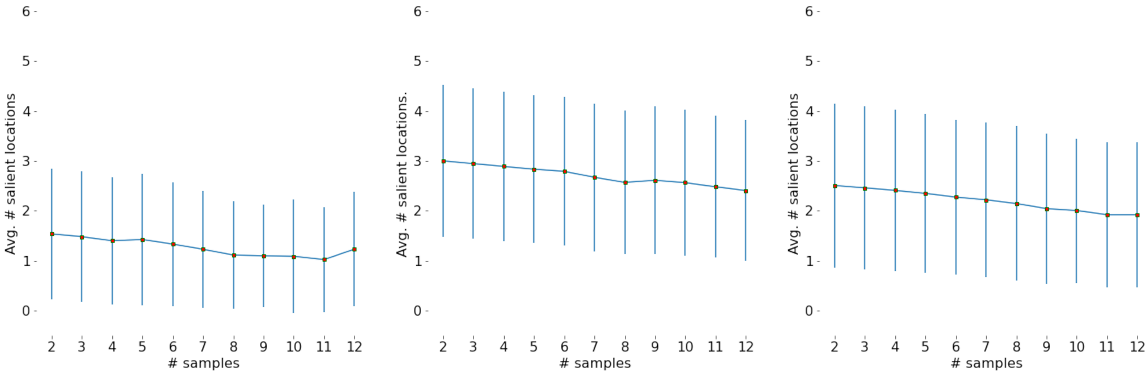


Figure 11.: Evaluation of our baseline model (ref. Section 4.3.1). (a) Classification accuracy (acc.) of the participants and (b) that of our baseline model with actual labels as ground truth. (c) Classification accuracy, (d) location prediction accuracy, (e) class rejection accuracy and (f) class selection accuracy of our baseline model with participants' data as ground truth. See Section 4.4.2 for details.

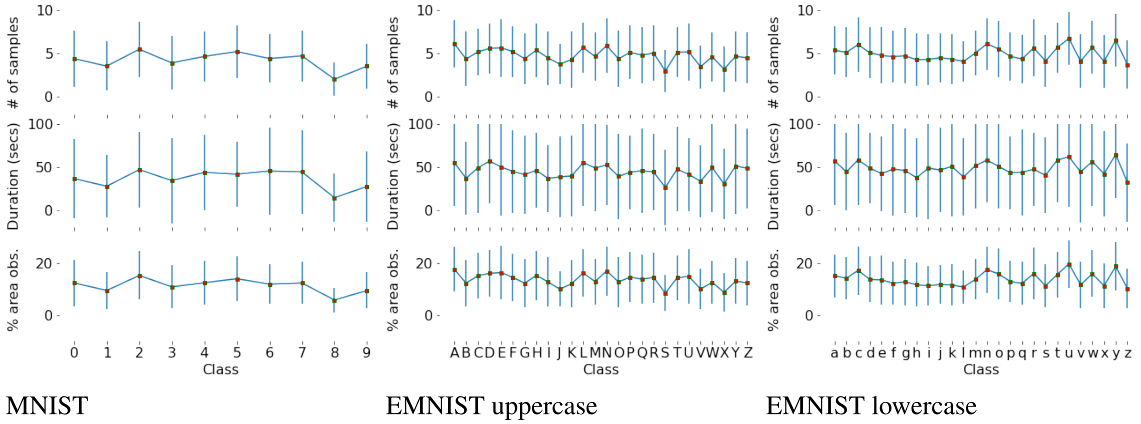


MNIST

EMNIST uppercase

EMNIST lowercase

Figure 12.: Errorbar plot showing the average number of salient locations selected from a saliency map for each sampling. Errorbars indicate standard deviation.



MNIST

EMNIST uppercase

EMNIST lowercase

Figure 13.: Minimum number of samples, corresponding time spent, and proportion of image area observed by the participants after which only the correct class is selected till the last sampling instant. Errorbars indicate standard deviation.

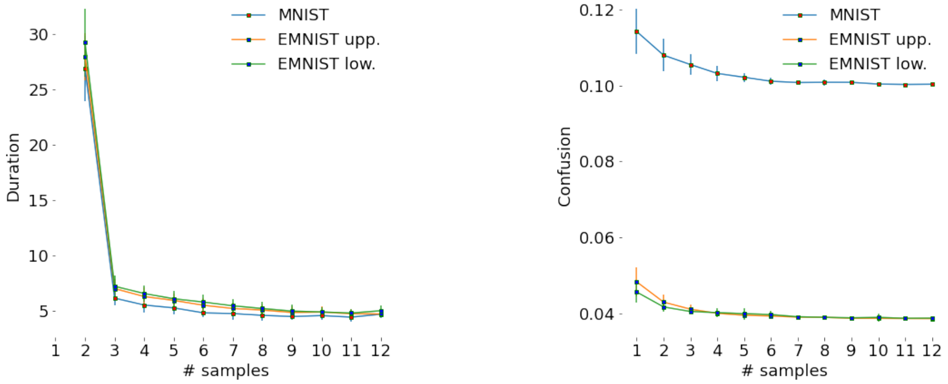


Figure 14.: (Left) Errorbar plot of time difference (seconds) between consecutive samples averaged over all classes. That is, value shown at sampling t is the time elapsed between a participant's responses to stimuli at $t-1$ and t. (Right) Errorbar plot of confusion averaged over all classes at each sampling. Here, confusion = # selected classes / total # classes. A significant positive correlation is observed between the duration and confusion plots. Errorbars indicate standard deviation.

Table 9.: One sample from our dataset for MNIST.

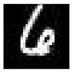
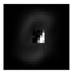
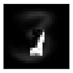
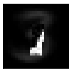
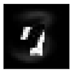
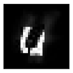
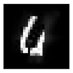
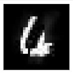
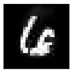
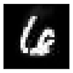
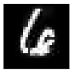
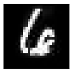
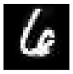
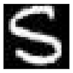
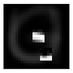
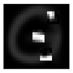
Stimulus MNIST	Observ-ation sequence	Sampled locations (x, y)	Duration between samples (sec)	Class(es) selected
		(14, 12)	-	3
		(13, 17)	29.35	3
		(14, 10)	8.52	3
		(9, 11)	5.78	4
		(7, 16)	7.54	4
		(12, 7)	4.66	4
		(18, 17)	7.25	6
		(18, 10)	10.22	6
		(20, 16)	4.58	6
		(13, 4)	3.97	6
		(21, 16)	4.41	6
		(9, 20)	3.38	6

Table 10.: One sample from our dataset for EMNIST-uppercase.

Stimulus EMNIST	Observation sequence	Sampled locations (x, y)	Duration between samples (sec)	Class(es) selected
		(14, 12)	-	G, U
		(17, 19)	9.37	G, O
		(20, 5)	12.79	G, O, P
		(20, 12)	21.77	G, O, P
		(9, 3)	11.47	G, O, P
		(21, 8)	16.84	G, O
		(3, 16)	17.69	G, S
		(21, 16)	39.32	S
		(5, 5)	6.21	S
		(21, 9)	6.51	S
		(22, 20)	7.9	S
		(7, 12)	6.12	S

Chapter 5

An Attention-based Predictive Agent for Handwritten Numeral/Alphabet Recognition via Generation

Abstract: A number of attention-based models for either classification or generation of handwritten numerals/alphabets have been reported in the literature. However, generation and classification are done jointly in very few end-to-end models. We propose a predictive agent model that actively samples its visual environment via a sequence of glimpses. The attention is driven by the agent's sensory prediction (or generation) error. At each sampling instant, the model predicts the observation class and completes the partial sequence observed till that instant. It learns where and what to sample by jointly minimizing the classification and generation errors. Three variants of this model are evaluated for handwriting generation and recognition on images of handwritten numerals and alphabets from benchmark datasets. We show that the proposed model is more efficient in handwritten numeral/alphabet recognition than human participants in a recently published study as well as a highly-cited attention-based reinforcement model. This is the first known attention-based agent to interact with and learn end-to-end from images for recognition via generation, with high degree of accuracy and efficiency.

5.1 Introduction

Perception and action are inextricably tied together as, in the real world, efficiency is as important as accuracy. Nature has evolved the visual system such that, to minimize resources, it learns to selectively attend to a few locations that provide information for the task at hand. This motivates our exploration of predictive agent models that observe the visual environment via a sequence of glimpses. Such agents predict, learn and act by minimizing sensory prediction error in a closed loop.

Our earlier work [Baruah and Banerjee, 2020b, Baruah and Banerjee, 2020a] explored attention-based predictive agents that learn to sequentially sample their environment for spatial and spatiotemporal data generation. In this paper, we propose an attention-based predictive agent for handwritten numeral and alphabet recognition in images. The attention (action) is driven by the agent's sensory prediction error.

Related work. Attention-based models can be hard or soft [Xu et al., 2015, Elsayed et al., 2019]. Hard-attention models make decisions by processing a part of the data, sampled via a sequence of glimpses. These models are reinforcement-based (e.g., [Elsayed et al., 2019, Mnih et al., 2014]), unsupervised (e.g., [Gregor et al., 2015, Eslami et al., 2016]) or supervised (e.g., [Zheng et al., 2015]). Soft-attention models process the entire data but weigh the features. Supervised (e.g., [Fukui et al., 2019]) and unsupervised (e.g., [Sang et al., 2020]) variants of these models have been reported. We propose an supervised (with class labels) hard-attention model.

Numerous attention-based models for either classification (e.g., [Mnih et al., 2014]) or generation (e.g., [Gregor et al., 2015, Baruah and Banerjee, 2020b]) of handwritten numerals/alphabets have been reported in the literature. However, generation and classification are done jointly in very few end-to-end models. Two models deserve mention: semi-supervised learning with generative models proposed in [Kingma et al., 2014], and a multimodal variational autoencoder robust to missing data introduced in [Wu and Goodman, 2018]. Though both models do generation and classification for handwritten numerals (MNIST), only classification accuracy is reported in [Kingma et al., 2014] while only generation accuracy in terms of negative log-likelihood is reported in [Wu and Goodman, 2018]. Further, none of them incorporate attention, i.e. an image is not sampled as a sequence of observations but presented in its entirety.

Contributions. In this paper, we propose an attention-based agent model that learns to classify handwritten numerals/alphabets from images by generating them. The novelty of this work is as follows:

1. The proposed model implements a perception-action loop as the optimization of an objective function. The action (attention) is modeled as proprioception in a

multimodal setting and is guided by perceptual prediction error, not by reinforcement.

2. At each sampling instant, the model simultaneously classifies and completes the partial sequence of observations. Pattern completion allows prediction error computation which decides the next sampling location.

3. Three variants of the component that maps the partial sequences of perceptual and proprioceptive observations to the class label and completed perceptual pattern are proposed. Their accuracies are comparable and correlate with the number of trainable parameters.

4. The proposed model is more efficient than the human participants in a recently published study [Baruah and Banerjee, 2021]. On average, the participants required 4.2, 4.7 and 4.9 samples to recognize a numeral, uppercase and lowercase alphabet, which correspond to 14.5%, 16.7% and 16.8% of image area, respectively. When exposed to the same stimuli and conditions as the participants, our model requires 2.0, 4.5, 4.2 samples which correspond to 8%, 15.7%, 14.1% of image area, respectively. A highly-cited attention-based reinforcement model [Mnih et al., 2014] falls short of human performance.

The rest of the paper is organized as follows. The proposed agent model is described in Section 5.2 and evaluated on various benchmark datasets in Section 5.3. The paper ends with concluding remarks in Section 5.4.

5.2 Models and Methods

5.2.1 Preliminaries

Agent. Anything that perceives from and acts upon its environment using sensors and actuators respectively is called an agent [Russell and Norvig, 2020].

Perception is the mechanism of interpreting sensory signals from the external environment by an agent [Han et al., 2016].

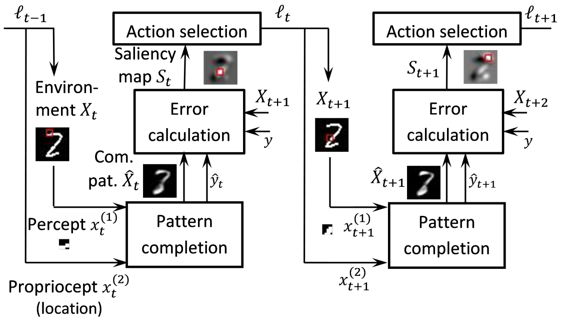
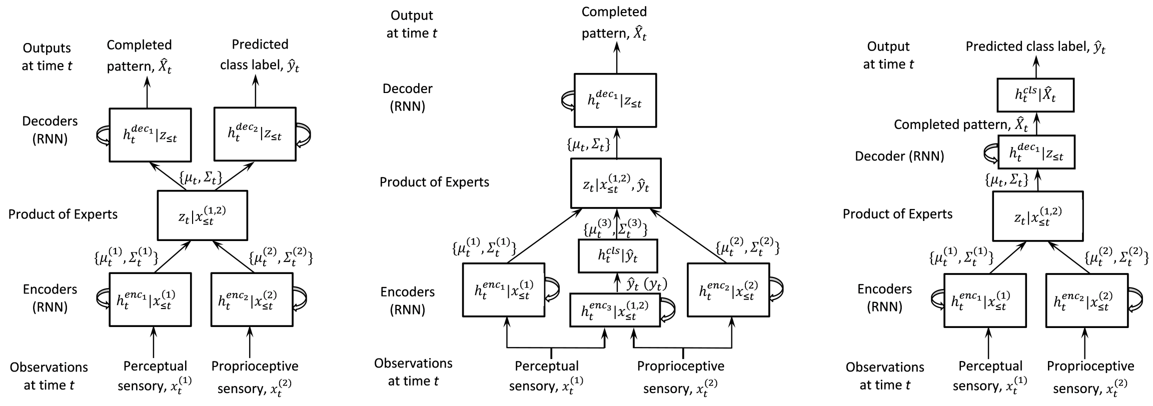


Figure 15.: Different components of the proposed agent. Implementation of the pattern completion block is shown in Fig. 16..


Outputs at time t
Completed pattern, $\hat{X}_t$
Predicted class label, $\hat{y}_t$
Decoders (RNN)
$h_t^{dec_1}|z_{\leq t}$
$h_t^{dec_2}|z_{\leq t}$
$\{\mu_t, \Sigma_t\}$
Product of Experts
$z_t|x_{\leq t}^{(1,2)}$
$\{\mu_t^{(1)}, \Sigma_t^{(1)}\}$
$\{\mu_t^{(2)}, \Sigma_t^{(2)}\}$
Encoders (RNN)
$h_t^{enc_1}|x_{\leq t}^{(1)}$
$h_t^{enc_2}|x_{\leq t}^{(2)}$
Observations at time t
Perceptual sensory, $x_t^{(1)}$
Proprioceptive sensory, $x_t^{(2)}$
Output at time t
Completed pattern, $\hat{X}_t$
Decoder (RNN)
$h_t^{dec_1}|z_{\leq t}$
$\{\mu_t, \Sigma_t\}$
Product of Experts
$z_t|x_{\leq t}^{(1,2)}, \hat{y}_t$
$\{\mu_t^{(3)}, \Sigma_t^{(3)}\}$
$h_t^{cls}|\hat{y}_t$
$\{\mu_t^{(1)}, \Sigma_t^{(1)}\}$
$\{\mu_t^{(2)}, \Sigma_t^{(2)}\}$
Encoders (RNN)
$h_t^{enc_1}|x_{\leq t}^{(1)}$
$\hat{y}_t\ (y_t)$
$h_t^{enc_3}|x_{\leq t}^{(1,2)}$
$h_t^{enc_2}|x_{\leq t}^{(2)}$
Observations at time t
Perceptual sensory, $x_t^{(1)}$
Proprioceptive sensory, $x_t^{(2)}$
Output at time t
Predicted class label, $\hat{y}_t$
$h_t^{cls}|\hat{X}_t$
Completed pattern, $\hat{X}_t$
Decoder (RNN)
$h_t^{dec_1}|z_{\leq t}$
$\{\mu_t, \Sigma_t\}$
Product of Experts
$z_t|x_{\leq t}^{(1,2)}$
$\{\mu_t^{(1)}, \Sigma_t^{(1)}\}$
$\{\mu_t^{(2)}, \Sigma_t^{(2)}\}$
Encoders (RNN)
$h_t^{enc_1}|x_{\leq t}^{(1)}$
$h_t^{enc_2}|x_{\leq t}^{(2)}$
Observations at time t
Perceptual sensory, $x_t^{(1)}$
Proprioceptive sensory, $x_t^{(2)}$

Model M1. Model M2. Model M3.

Figure 16.: Three variations for implementing the pattern completion block in Fig. 15..

Proprioception is a form of perception in which the agent's environment is its own body. Internal perception of position, movement, and motion of body parts is due to proprioception [Han et al., 2016].

Generative model. Given a set of data points x, a generative model p_{model} with parameters θ maximizes the log-likelihood, $\mathcal{L}(x;\theta)$, of the data.

Evidence lower bound (ELBO). Let the data x be generated by a latent continuous random variable z. Then, computing the log-likelihood requires integrating the marginal likelihood, $\int p_{model}(x,z)dz$, which is intractable [Kingma and Welling, 2013]. In variational inference, an approximation of the intractable posterior is optimized by defining an evidence lower bound (ELBO) on the log-likelihood, $\mathcal{L}(x;\theta) \leq \log p_{model}(x;\theta)$.

Variational autoencoder (VAE) is a multilayered generative model. It assumes an isotropic Gaussian prior, $p_\theta(z)$, and i.i.d. data samples. VAE maximizes the following ELBO [Kingma and Welling, 2013]:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\text{KL}}[q_\phi(z|x), p_\theta(z)] \tag{5.1}$$

where $p_\theta(x|z)$ and $q_\phi(z|x)$ are generative and recognition models respectively, $\mathbb{E}$ denotes expectation, and D_{KL} denotes Kullback-Leibler divergence. The first and second terms capture accuracy and complexity respectively. The negative of this ELBO is also known as *variational free energy*, minimization of which has been hypothesized as a general principle guiding brain function [Friston, 2010].

Saliency lies in the eyes of an agent. Saliency of a location in an environment is a function of its neighborhood and an agent's internal model (see [Spratling, 2012, Friston et al., 2009]).

5.2.2 Problem Statement

Let an environment in m modalities be represented by a set of observable variables $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(m)}\}$. The variable representing the i-th modality is a sequence: $\mathbf{X}^{(i)} = \langle X_1^{(i)}, X_2^{(i)}, \ldots, X_T^{(i)} \rangle$, where T is the sequence length. Let $\mathbf{x}_{\leq t} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}\}$ be a partial observation of $\mathbf{X}$ such that $\mathbf{x}^{(i)} = \langle x_1^{(i)}, \ldots, x_t^{(i)} \rangle$, $1 \leq t \leq T$. Let y represent the class label.

We define *pattern completion and classification* as the problem of accurately generating $\mathbf{X}$ and y from the partial observation $\mathbf{x}_{\leq t}$. Given $\mathbf{x}_{\leq t}$ and a generative model p_θ with parameters θ and latent variables $z_{\leq t}$, the objective for pattern completion and classification at any time t is to maximize the joint log-likelihood of $\mathbf{X}$ and y, i.e., $\arg\max_\theta \int log(p_\theta(\mathbf{X}, y|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_\theta(z_{\leq t}))dz$.

5.2.3 Models

We solve the problem in three distinct ways as follows.

Model M1 (ref. Fig. 16.a): The completed pattern and class label are generated from the latent variables. Mathematically,

$\arg\max_{\theta} \int log(p_{\theta}(\mathbf{X}|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_{\theta}(z_{\leq t}))dz + \arg\max_{\theta} \int log(p_{\theta}(y|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_{\theta}(z_{\leq t}))dz.$ The model is trained end-to-end.

Model M2 (ref. Fig. 16.b): The class label is inferred from the partial observation. The latent variables are inferred from the class label and partial observation, as in [Kingma et al., 2014]. Mathematically, $\arg\max_{\theta} \int log(p_{\theta}(\mathbf{X}|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_{\theta}(z_{\leq t}))dz + \arg\max_{\phi} \log q_{\phi}(y_t|\mathbf{x}_{\leq t})$, where q_{ϕ} is a recognition model. The model is trained end-to-end.

Model M3 (ref. Fig. 16.c): The class label is inferred from the completed pattern which is generated from the latent variables. The pattern completion model is trained first, $\arg\max_{\theta} \int log(p_{\theta}(\mathbf{X}|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_{\theta}(z_{\leq t}))dz$. Then the classification model is trained, $\arg\max_{\pi} log(p_{\pi}(y|\mathbf{X})$.

5.2.4 Agent Architecture

As shown in the block diagram in Fig. 15., environment, observation, pattern completion and classification, action selection and learning are the five components of the proposed agent architecture.

1. Environment. The environment is the source of sensory data. We consider static (images) environment in our experiments.

2. Observation. Our agent sequentially samples its environment in two modalities: visual perception and visual proprioception. The 2D coordinates of the fixation location in the environment constitutes the proprioceptive observation while the visual stimuli at that location constitutes the corresponding perceptual observation, as in [Friston et al., 2012]. See Table 11. for variable dimensions.

3. Pattern completion. At each sampling instant, the partial observation till that

Table 11.: Variable dimensions as used in this paper. Here $(.)^{(1)}$, $(.)^{(2)}$ refer to visual perception and visual proprioception respectively; T is maximum number of glimpses, t is glimpse index or time, $n \times n$ is patch size, $M \times M$ is image size.

$x_t^{(1)}$	$x_t^{(2)}$	X_t	S_t
$\{0,1\}^{n \times n}$	$\mathbb{R}^2$	$\{0,1\}^{M \times M}$	$\mathbb{R}^{M \times M}$

instant is completed using a multimodal variational recurrent neural network (MVRNN). Recognition and generation are the two processes involved in the operation of a MVRNN.

Recognition (Encoder). The recognition model, $q_\phi(z_t|\mathbf{x}_{\leq t})$ for M1 and M3, and $q_\phi(z_t|\mathbf{x}_{\leq t}, y_t)$ for M2, is a probabilistic encoder [Kingma and Welling, 2013]. It produces a Gaussian distribution over the possible values of the code z_t from which the given observations could have been generated.

Model M1: Two RNNs, each with one layer of long short-term memory (LSTM) units, constitute the recognition model. Each RNN infers the parameters of the approximate posterior distribution for each modality.

Model M2: In addition to the perceptual and proprioceptive modalities, the class label is an input modality. A fully-connected layer maps the class labels (inferred label $\hat{y}$ or given label y) to the parameters ($\mu^{(3)}$, $\Sigma^{(3)}$) of the approximate posterior density for the class label modality (ref. Fig. 16.b).

Model M3: Same as M1.

The parameters for all modalities are combined using product of experts (PoE) [Wu and Goodman, 2018] to generate the joint distribution for the approximate posterior, $q_\phi(z_t|\mathbf{x}_{\leq t})$ for M1 and M3, and $q_\phi(z_t|\mathbf{x}_{\leq t}, y_t)$ for M2.

The prior can be sampled from a standard normal distribution $p_\theta(z_t) \sim \mathcal{N}(0, 1)$, as in [Gregor et al., 2015]. The function of the encoder is shown in Lines 1–5 of Algorithm 6 and Lines 4–13 of Algorithm 7, where RNN_ϕ^{enc} represents the function of a LSTM unit, φ^{enc} is a function that returns the mean and the logarithm of the standard deviation as a linear function of the hidden state, as in [Chung et al., 2015].

Generation (Decoder).

Model M1: The model, $p_\theta(X_t, y_t|\mathbf{x}_{\leq t}, z_{\leq t})$, generates the perceptual data and the class label from the latent variables, z_t, at each time step. The generative model consists of two RNNs, each with one layer of hidden LSTM units.

Model M2: The model, $p_\theta(X_t|\mathbf{x}_{\leq t}, z_{\leq t})$, generates the perceptual data from the latent variables, z_t. The generative model consists of one RNN with a single layer of hidden LSTM units.

Model M3: Same as M2.

Each RNN generates the parameters of the data distribution for a modality. The data is sampled from this distribution which can be multivariate Gaussian or Bernoulli. In our model, both X_t and y_t are sampled from a multivariate Bernoulli distribution with means inferred by the corresponding decoder RNN. In order to generate the perceptual data at any time step, the output from the perceptual RNN at the previous time step is added to the current perceptual RNN output before applying the sigmoid function, as in [Gregor et al., 2015] (ref. Line 8 of Algorithm 6 and Line 20 of Algorithm 7). The decoder equations are shown in Lines 7–10 of Algorithm 6 and Lines 15–20 of Algorithm 7, where the functions RNN_θ^{dec} and φ^{dec} are same as RNN_ϕ^{enc} and φ^{enc} respectively.

4. Classification.

Model M1: The decoder infers the class label as a separate modality for each time step (ref. M1 in Generation (Decoder)).

Model M2: The class labels are inferred from the partial observations, $\mathbf{x}_{\leq t}$, at every time step. An RNN with LSTM units is used as a hidden layer, along with a softmax classifier. The function of the classifier is shown in Lines 1–3 of Algorithm 7.

Model M3: A classifier[1] is trained separately to infer the class labels from the perceptual data. During training, the input to the classifier is the true perceptual data. During testing, the input is the predicted perceptual data.

5. Action selection. In our model, action selection is to decide the location in the environment to sample from. At any time t, a saliency map S_t is computed which assigns a salience score $S_t^{(\ell)}$ to each location ℓ.

$$S_t^{(\ell)} = D_{KL}(p(X_{t+1,\ell})||p_\theta(X_{t+1,\ell}|z_{\leq t}, \mathbf{x}_{\leq t})) \tag{5.2}$$

[1] We used a CNN classifier with code borrowed from https://chromium.googlesource.com/external/github.com/tensorflow/tensorflow/+/r0.10/tensorflow/g3doc/tutorials/mnist/pros /index.md.

where $p(X_{t+1,\ell})$ is the true data distribution at location ℓ and is sampled from a Bernoulli distribution. KL divergence, also known as *relative entropy*, is a measure of information gain achieved by using the true distribution, $p(X_{t+1,\ell})$, instead of the predicted distribution, $p_\theta(X_{t+1,\ell}|z_{\leq t}, \mathbf{x}_{\leq t})$. Thus, the saliency map is a function of the prediction error. The most salient location is computed from this saliency map which constitutes the sampling location.

The saliency map is smoothed using a Gaussian kernel $\mathcal{N}(.,\sigma)$. The sampling location is chosen as:

$$\ell_t = \underset{\ell_t \in \{1,2,\ldots,M^2\}}{\operatorname{argmax}} \text{conv}(\mathcal{N}(.,\sigma), S_t) \tag{5.3}$$

where $\sigma = 2$. Each sample is a $n \times n$ patch centered at ℓ_t.

The salient location ℓ_t at any time t is the proprioceptive observation $x_{t+1}^{(2)}$ for time $t+1$. Hence, prediction error (saliency) guides the sampling of a scene in our model. Unlike typical multimodal models, the two modalities in our model interact at the observation level as the perceptual prediction error provides the observation for the visual proprioceptive modality. The most salient location is the location that yields the maximum information gain in the environment. These are the locations where the agent's prediction error is the highest given all the past observations. The agent attends to these locations to update its internal model.

6. Learning. The objective is to maximize Equ. 5.4, 5.5 and 5.6 for M1, M2 and M3 respectively. It can be derived from the objectives for multimodal VAE [Wu and Goodman, 2018], variational RNN [Chung et al., 2015] and VAE for classification [Kingma et al., 2014].

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\lambda_1 \log p_\theta(X_t|z_{\leq t},\mathbf{x}_{\leq t}) \\ +\lambda_2 \log p_\theta(y_t|z_{\leq t},\mathbf{x}_{\leq t})\Big] \\ -\sum_{t=1}^{T}\beta D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big) \tag{5.4}$$

where λ_1, λ_2, β are the weights balancing the terms.

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}\log p_\theta(X_t|z_{\leq t},\mathbf{x}_{\leq t}) + \log p_\theta(y_t)\Big] \\ -\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, y_t), p_\theta(z_t)\big) + \sum_{t=1}^{T}\alpha \log q_\phi(y_t|\mathbf{x}_{\leq t}) \tag{5.5}$$

where α controls the relative weight between generative and purely discriminative learning.

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\log p_\theta(X_t|z_{\leq t},\mathbf{x}_{\leq t})\Big] \\ -\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big) + \log q_\pi(y|X) \tag{5.6}$$

where $q_\pi(y|X)$ is the classification model whose input is the entire image (completed pattern) and not a sequence of observations. So the subscript t is dropped.

Loss function derivation: Model M1

Here we derive the objective function in Eq. 5.4. The generative and recognition models are factorized as:

$$
\begin{aligned}
p_\theta(X_{\leq T}, y_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T}) &= \prod_{t=1}^{T} p_\theta(X_t, y_t|z_{\leq t}, \mathbf{x}_{\leq t}) p_\theta(z_t) \\
q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}) &= \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\leq t})
\end{aligned}
$$

The variational lower bound (ELBO) on the joint log-likelihood of the generated data, $\log p_\theta(X_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T})$, is derived as:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\log p_\theta(X_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T}) \frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\log \frac{p_\theta(X_{\leq T}, y_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T})}{p_\theta(z_{\leq T}|\mathbf{x}_{\leq T})} \\
&\qquad \frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \log \frac{p_\theta(X_t, y_t|z_{\leq t}, \mathbf{x}_{\leq t}) p_\theta(z_t)}{p_\theta(z_t|\mathbf{x}_{\leq t})} \\
&\qquad \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{q_\phi(z_t|\mathbf{x}_{\leq t})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \Big[\log p_\theta(X_t, y_t|z_{\leq t}, \mathbf{x}_{\leq t}) \\
&\qquad - \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{p_\theta(z_t)} + \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{p_\theta(z_t|\mathbf{x}_{\leq t})}\Big]\Big] \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \log p_\theta(X_t, y_t|z_{\leq t}, \mathbf{x}_{\leq t})\Big] \\
&\qquad - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big)
\end{aligned}
$$

We assume, the modalities X_t and y_t are conditionally independent given the common latent variables [Wu and Goodman, 2018] and all observations till the current time. Therefore,

$$\begin{aligned}
&\log p_\theta(X_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T}) \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \lambda_1 \log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) && (5.7)\\
&+ \lambda_2 \log p_\theta(y_t|z_{\leq t}, \mathbf{x}_{\leq t})\Big] - \sum_{t=1}^{T} \beta D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big) && (5.8)
\end{aligned}$$

where λ_1, λ_2, β are the weights balancing the terms.

Loss function derivation: Model M2

Here we derive the objective function in Eq. 5.5. The generative and recognition models are factorized as:

$$p_\theta(X_{\leq T}, y_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^{T} p_\theta(X_t, y_t|z_{\leq t}, \mathbf{x}_{\leq t}) p_\theta(z_t)$$
$$q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T}) = \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\leq t}, y_t)$$

The variational lower bound (ELBO) on the log-likelihood of the generated data, $\log p_\theta(X_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T})$, when the true label is given is derived as:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\log p_\theta(X_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T}) \\
&\qquad\qquad \frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\log \frac{p_\theta(X_{\leq T}, z_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T})}{p_\theta(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})} \\
&\qquad\qquad \frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T} \log \frac{p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) p_\theta(z_t) p_\theta(y_t)}{p_\theta(z_t|\mathbf{x}_{\leq t}, y_t)} \\
&\qquad\qquad \frac{q_\phi(z_t|\mathbf{x}_{\leq t}, y_t)}{q_\phi(z_t|\mathbf{x}_{\leq t}, y_t)}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}\Big[\log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) + \log p_\theta(y_t) \\
&\qquad - \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t}, y_t)}{p_\theta(z_t)} + \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t}, y_t)}{p_\theta(z_t|\mathbf{x}_{\leq t}, y_t)}\Big]\Big] \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T} \log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) + \log p_\theta(y_t)\Big] \\
&\qquad - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, y_t), p_\theta(z_t)\big) \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}(\log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) + \log p_\theta(y_t))\Big] \\
&\qquad - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, y_t), p_\theta(z_t)\big)
\end{aligned}
$$

After adding the classification loss, the final objective function can be written as:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T} \log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) + \log p_\theta(y_t)\Big] \\
&- \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, y_t), p_\theta(z_t)\big) + \sum_{t=1}^{T} \alpha \log q_\phi(y_t|\mathbf{x}_{\leq t})
\end{aligned} \tag{5.9}
$$

where α controls the relative weight between generative and purely discriminative learning.

Loss function derivation: Model M3

Here we derive the objective function in Eq. 5.6. The generative and recognition models are factorized as:

$$\begin{aligned} p_\theta(X_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T}) &= \prod_{t=1}^{T} p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) p_\theta(z_t) \\ q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}) &= \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\leq t}) \end{aligned}$$

The variational lower bound (ELBO) on the log-likelihood of the generated data, $\log p_\theta(X_{\leq T}|\mathbf{x}_{\leq T})$, is derived as:

$$\begin{aligned} &\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\log p_\theta(X_{\leq T}|\mathbf{x}_{\leq T}) \frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big] \\ &= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\log \frac{p_\theta(X_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T})}{p_\theta(z_{\leq T}|\mathbf{x}_{\leq T})} \frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big] \\ &= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \log \frac{p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) p_\theta(z_t)}{p_\theta(z_t|\mathbf{x}_{\leq t})} \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{q_\phi(z_t|\mathbf{x}_{\leq t})}\Big] \\ &= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \Big[\log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t}) \\ &\qquad - \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{p_\theta(z_t)} + \log \frac{q_\phi(z_t|\mathbf{x}_{\leq t})}{p_\theta(z_t|\mathbf{x}_{\leq t})}\Big]\Big] \\ &\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{\leq t})\Big] \\ &\qquad - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big) \end{aligned}$$

After adding the classification loss, the final objective function can be written as:

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\log p_\theta(X_t|z_{\leq t},\mathbf{x}_{\leq t})\Big] \\ -\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big) + \log q_\pi(y|X)$$

where $q_\pi(y|X)$ is the classification model whose input is the entire image (completed pattern) and not a sequence of observations. So the subscript t is dropped.

We assume a one-to-one mapping between the agent's body and its environment, i.e. between the oculomotor muscles to the locations in the image. This assumption allows us to map from the perceptual space ℓ to the proprioceptive space $x^{(2)}$ using a simple function g_3 (ref. Line 9 of Algorithm 5).

5.2.5 Metrics for comparing fixation maps

In order to evaluate the action mechanism of our model, we compare the fixation map obtained from the sequence of locations sampled by our model with that of the fixation map obtained from participants' data in [Baruah and Banerjee, 2021]. The fixation map is computed by assigning each location a value equal to the frequency of its selection, and then normalizing the values to create a distribution over all locations.

For metrics comparing two fixation maps, P and Q, we closely follow [Bylinskii et al., 2018]. We use three distribution-based metrics: KL divergence (KL), Pearson correlation coefficient (CC), and Similarity (SIM), to compare the distribution of sampling locations from a model with that from the participants as recorded in the collected data.

KL divergence. [Bylinskii et al., 2018] Given two image distributions, P and Q, the KL divergence $KL(P, Q)$ measures the loss of information when Q is used to approximate P. This is calculated for each pixel k as: $KL(P_k, Q_k) = P_k \log\left(\epsilon + \frac{P_k}{Q_k+\epsilon}\right)$,

Algorithm 5 Learning the proposed network

```
Initialize parameters of the generative model $\theta$, recognition model $\phi$, sequence length $T$.
Initialize optimizer parameters: $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\eta = 0.001$, $\epsilon = 10^{-10}$.
Initialize $x_1^{(1)} \leftarrow F(X_1, \ell_0)$, $x_1^{(2)} \leftarrow g_3(\ell_0)$, where $\ell_0$ is the initial sampling location (ref. Experimental setup in Section 5.3), $g_3$ is an identity function (ref. Action selection in Section 5.2.4), and the function $F$ extracts a sample $x^{(1)}$ (e.g., $5 \times 5$ patch) from the environment $X$ (e.g., $28 \times 28$ image) at location $\ell$ (e.g., center of the image).
while true do
    for $\tau \leftarrow 1 \; to \; T$ do
        Model M1:
        $\hat{X}_\tau, \hat{y}_\tau \leftarrow PatComClassModel1(x_{1:\tau}^{(1:2)})$
        Model M2:
        $\hat{X}_\tau, \hat{y}_\tau \leftarrow PatComClassModel2(x_{1:\tau}^{(1:2)})$
        Model M3:
        $\hat{X}_\tau \leftarrow PatComClassModel1(x_{1:\tau}^{(1:2)})$
        $\hat{y}_\tau \leftarrow Classifier(X_\tau)$
        Saliency Computation
        $S_\tau \leftarrow g_1(X_{\tau+1}, \hat{X}_\tau)$        [ref. Eq. 5.2]
        $\ell_\tau \leftarrow g_2(S_\tau)$        [ref. Eq. 5.3]
        $x_{\tau+1}^{(2)} \leftarrow g_3(\ell_\tau)$
        $x_{\tau+1}^{(1)} \leftarrow F(X_{\tau+1}, \ell_\tau)$
        Learning
        Update $\{\theta, \phi\}$ or $\{\theta, \phi, \pi\}$ by maximizing Eq. 5.4, 5.5 or 5.6.
    end for
end while
```

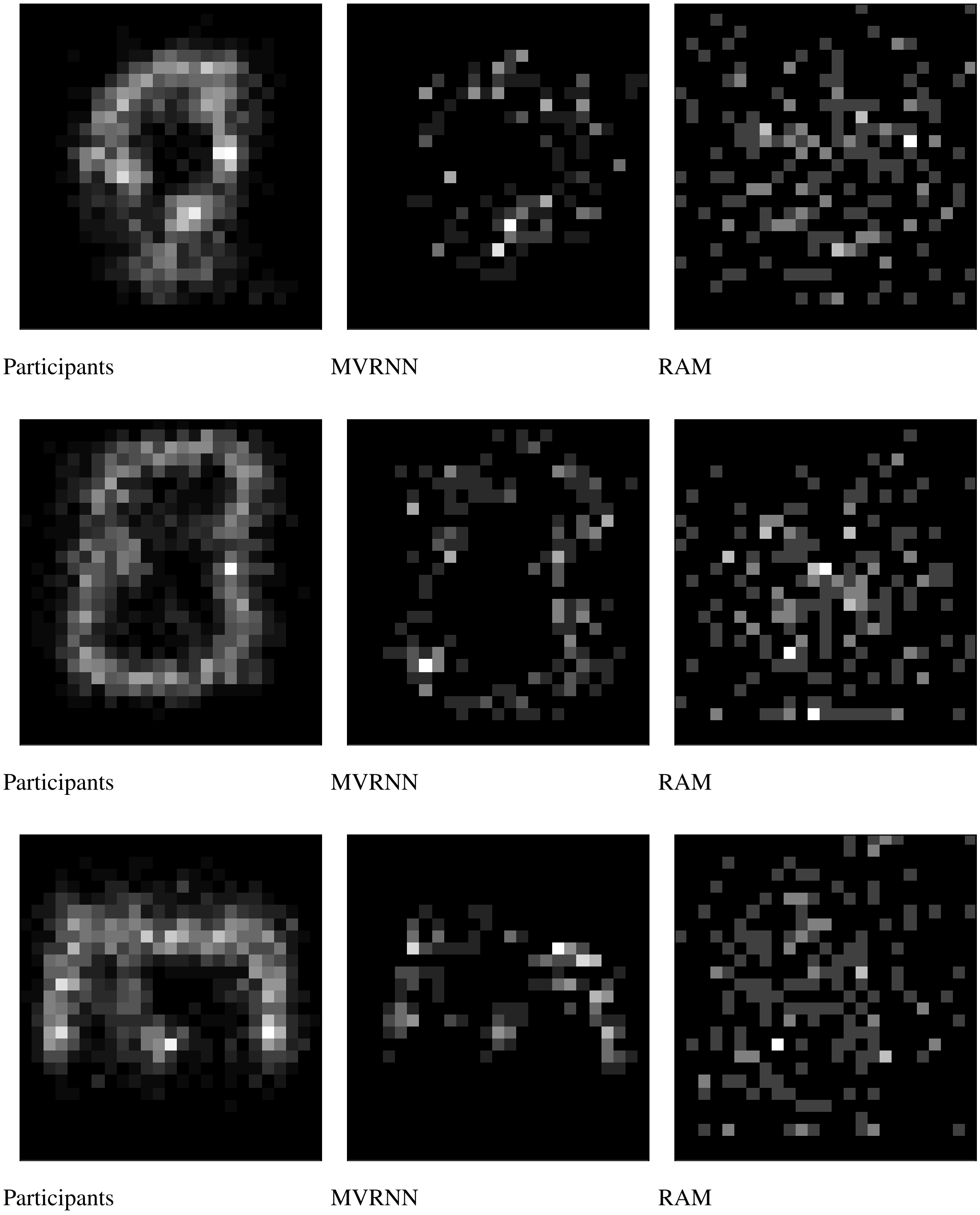
where ϵ is a very small real number. Lower KL divergence for k implies P_k and Q_k are similar. KL divergence is highly sensitive to zero values.

CC can evaluate the linear relationship between two maps as [Bylinskii et al., 2018]: $CC(P, Q) = \frac{\sigma(P,Q)}{\sigma(P)\sigma(Q)}$, where σ is the variance or covariance. Since CC is symmetric, it fails to infer whether differences between fixation maps are due to false positives or false negatives.

SIM is measured as [Bylinskii et al., 2018]: $SIM(P, Q) = \sum_k \min(P_k, Q_k)$, where $\sum_k P_k = \sum_k Q_k = 1$. Like CC, SIM is symmetric and inherits the same drawback. Also, SIM is very sensitive to missing values, and penalizes predictions that fail to account for the ground truth density.

Table 12.: Evaluation of fixation maps from RAM and our model (Model 1) for the stimuli presented in the MTurk experiments, averaged over all classes and samplings. Standard deviations are included in parenthesis.

Metric	MNIST		EMNIST uppercase		EMNIST lowercase	
	MVRNN	RAM	MVRNN	RAM	MVRNN	RAM
KL	$22.44(\pm 7.50)$	$22.50(\pm 7.48)$	$22.90(\pm 7.55)$	$22.96(\pm 7.24)$	$22.30(\pm 7.37)$	$22.23(\pm 7.16)$
CC	$0.02(\pm 0.01)$	$0.01(\pm 0.00)$	$0.02(\pm 0.01)$	$0.01(\pm 0.00)$	$0.02(\pm 0.01)$	$0.01(\pm 0.00)$
SIM	$0.18(\pm 0.11)$	$0.17(\pm 0.09)$	$0.16(\pm 0.10)$	$0.16(\pm 0.07)$	$0.18(\pm 0.10)$	$0.18(\pm 0.09)$



Participants MVRNN RAM

Participants MVRNN RAM

Participants MVRNN RAM

Figure 17.: Comparison of the distribution of the sequence of fixations over a class for different cases; classes ‘9’, ‘B’, ‘m’ are shown in rows 1 to 3 respectively. The fixations are scattered in case of RAM, our model shows similar pattern with the participants data.

Algorithm 6 $PatComClassModel1(x_{1:\tau}^{(1:2)})$

```
Recognition Model
for $i \leftarrow 1 \; to \; 2$ do
    $h_\tau^{enc_i} \leftarrow RNN_\phi^{enc}(x_{1:\tau}^{(i)}, h_{\tau-1}^{enc_i})$
    $[\mu_\tau^{(i)} ; \Sigma_\tau^{(i)}] \leftarrow \varphi^{enc}(h_\tau^{enc_i})$
end for
Product of Experts
$z_\tau \sim \mathcal{N}(\mu_\tau, \Sigma_\tau)$, where
$\Sigma_\tau \leftarrow \Big( \sum_{i=1}^{2} \Sigma_\tau^{(i)^{-2}} \Big)^{-1}, \; \mu_\tau \leftarrow \Big( \sum_{i=1}^{2} \mu_\tau^{(i)} \Sigma_\tau^{(i)^{-2}} \Big) \Sigma_\tau$
Generative Model
Pattern completion:
$h_\tau^{dec_1} \leftarrow RNN_\theta^{dec}(z_\tau, h_{\tau-1}^{dec_1})$
$\hat{X}_\tau \leftarrow f_\sigma(h_\tau^{dec_1}, \hat{X}_{\tau-1})$
Classification (Model M1):
$h_\tau^{dec_2} \leftarrow RNN_\theta^{dec}(z_\tau, h_{\tau-1}^{dec_2})$
$\hat{y}_\tau \leftarrow softmax(h_\tau^{dec_2})$
```

5.3 Experimental Results

5.3.1 Datasets

Our model is evaluated using the following datasets:

(1) MNIST [LeCun et al., 1998] is a dataset of handwritten numerals $\{0, 1, \ldots 9\}$, consisting of 60,000 training and 10,000 test images (28×28 pixels).

(2) EMNIST [Cohen et al., 2017] is a balanced dataset of handwritten English alphabets in uppercase and lowercase, consisting of 124,800 training and 20,800 test images (28×28 pixels).

(3) Sampled MNIST and EMNIST [Baruah and Banerjee, 2021] is a dataset consisting of a sequence of time-stamped samples from MNIST and EMNIST datasets and collected from participants using MTURK. Each sample consists of: (1) the location in the image selected by the participant, (2) the class(es) selected by the participant, and (3) the time taken by the participant to register the current sample (i.e. the time elapsed between registering the last and current samples). This data is recorded from 15 distinct

Algorithm 7 $PatComClassModel2(x^{(1:2)}_{1:\tau}, y_{1:\tau})$

```
Classification Model
$h^{cls}_{\tau} = RNN^{cls}_{\alpha}(h^{cls}_{\tau-1}, \mathbf{x}_{1:\tau})$
$\hat{y}_{\tau} = softmax(h^{cls}_{\tau})$
Recognition Model
for $i \leftarrow 1 \; to \; 2$ do
    $h^{enc_i}_{\tau} \leftarrow RNN^{enc}_{\phi}(x^{(i)}_{1:\tau}, h^{enc_i}_{\tau-1})$
    $[\mu^{(i)}_{\tau} \, ; \Sigma^{(i)}_{\tau}] \leftarrow \varphi^{enc}(h^{enc_i}_{\tau})$
end for
if labels are present then
    $h^{enc_3}_{\tau} \leftarrow tanh(y_{\tau})$
else
    $h^{enc_3}_{\tau} \leftarrow tanh(\hat{y}_{\tau})$
end if
$[\mu^{(3)}_{\tau} \, ; \Sigma^{(3)}_{\tau}] \leftarrow \varphi^{enc}(h^{enc_3}_{\tau})$
Product of Experts
$z_{\tau} \sim \mathcal{N}(\mu_{\tau}, \Sigma_{\tau})$, where
$\Sigma_{\tau} \leftarrow \Big(\sum_{i=1}^{3} \Sigma^{(i)^{-2}}_{\tau}\Big)^{-1}, \; \mu_{\tau} \leftarrow \Big(\sum_{i=1}^{3} \mu^{(i)}_{\tau} \Sigma^{(i)^{-2}}_{\tau}\Big)\Sigma_{\tau}$
Generative Model
Pattern Completion:
$h^{dec(1)}_{\tau} \leftarrow RNN^{dec}_{\theta}(z_{\tau}, h^{dec_1}_{\tau-1})$
$\hat{X}_{\tau} \leftarrow f_{\sigma}(h^{dec_1}_{\tau}, \hat{X}_{\tau-1})$
```

stimuli from each class for MNIST, EMNIST uppercase and EMNIST lowercase letters. The entire dataset is collected from 382 distinct participants. The dataset consists of 1743 samples from MNIST, 4461 samples from EMNIST uppercase, and 4114 samples from EMNIST lowercase; that is, 166.4 responses per class on average.

5.3.2 Experimental setup

The generative, recognition and classification models consist of 512, 128, 128 hidden units respectively. The latent variable dimension is 20. These parameters are estimated experimentally, and are consistent with model parameters reported in the literature. For example, the multimodal model in [Wu and Goodman, 2018] uses latent variable dimension of 64 and two MLP hidden layers of 512 units each for MNIST generation and classification, the model in [Gregor et al., 2015] uses latent variable dimension of 100 and an RNN hidden layer of 256 units for MNIST generation, and the model in [Mnih et al., 2014] uses an RNN hidden layer of 256 units for MNIST classification.

Maximum number of glimpses $T = 12$, and minibatch size is 100. The parameters β, λ_1, are fixed to 1, λ_2 and α is fixed as 5000. The model is learned end-to-end using backpropagation and Adam optimization [Kingma and Ba, 2014] with a learning rate of 10^{-3}. These hyperparameters are estimated via cross-validation using 10,000 images from the training set. The first observation is sampled from the center pixel of an image, as in the participants' data [Baruah and Banerjee, 2021].

We use a dropout probability of 0.7 to prevent overfitting. The dropout is applied at the decoder hidden layers for all the modalities in M1 and M3, and both the decoder hidden layer and the classification hidden layer for M2. Additionally, the KL divergence term in the objective function also acts as a regularizer [Kingma and Welling, 2013] that prevents overfitting.

Evaluation

The quality of the generated images is evaluated by negative log-likelihood (NLL), as in [Gregor et al., 2015] and the class prediction is evaluated by classification accuracy.

The three metrics, KL, CC and SIM, are used to evaluate the fixation maps obtained from the sequence of locations.

Efficiency of the model is evaluated with respect to the number of glimpses required for accurate prediction and is evaluated on the sampled MNIST and EMNIST datasets [Baruah and Banerjee, 2021].

We compare the efficiency and fixation maps with a highly-cited reinforcement model, recurrent attention model (RAM) [Mnih et al., 2014], that reports experimental results on the MNIST dataset. RAM classifies images using a sequence of glimpses. The next location is chosen stochastically from a distribution parameterized by a location

Table 13.: Classification accuracy and the NLL on the test set reported after the final glimpse.

Dataset	Variants of the proposed model	Accuracy %	NLL $\leq$
MNIST	M1	96.3	76.5
	M2	92.3	107.0
	M3 (pretrained)	94.6	76.1
	M4 (not end-to-end)	82.9	76.1
EMNIST	M1	90.2	125.8
	M2	80.4	82.6
	M3 (pretrained)	88.5	78.9
	M4 (not end-to-end)	75.4	78.9

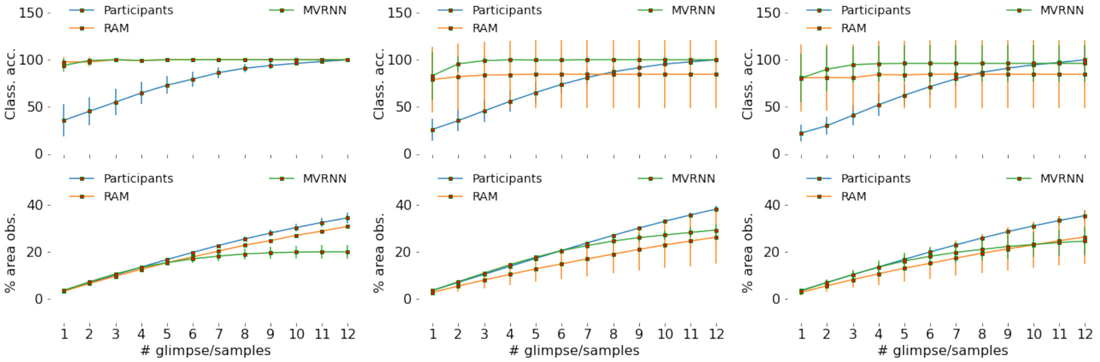
Table 14.: Classification accuracy and the NLL on the stimuli presented to the participants as in [Baruah and Banerjee, 2021] reported after the final glimpse.

Dataset	Variants of the proposed model	Accuracy %	NLL $\leq$
MNIST	M1	100	71.3
	M2	96	102.5
	M3 (pretrained)	98.7	71.8
	M4 (not end-to-end)	20.7	71.8
EMNIST upp.	M1	98.7	129.7
	M2	90.2	91.7
	M3 (pretrained)	98.7	83.9
	M4 (not end-to-end)	76.9	83.9
EMNIST low.	M1	95.6	111.0
	M2	85.4	66.8
	M3 (pretrained)	96.9	62.3
	M4 (not end-to-end)	74.9	62.3

network. For a fair comparison with the participants, in RAM[2] we fixed the sequence length at $T = 12$, the first sampling location at the image center, the input observation to a 5×5 patch with the selected location as its center, and modified the reward function according to the experimental setup in [Baruah and Banerjee, 2021] .

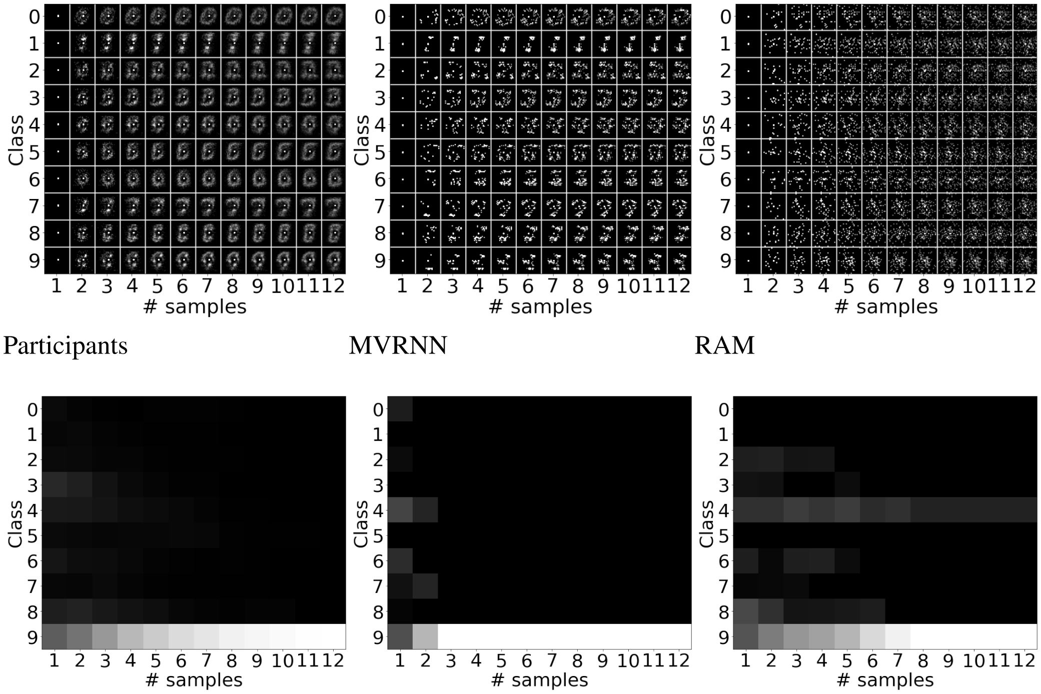
Apart from the 3 model variations, we add one more variation of our model in which initially the generative model is trained similar to M3 and then a RNN with LSTM units is used to classify the data from the latent variables. We refer to this model as M4.

[2] We use the RAM implementation from github.com/hehefan/Recurrent-Attention-Model.



MNIST EMNIST uppercase EMNIST lowercase

Figure 18.: Errorbar plot shown for the classification accuracy and the percentage of data seen by a participant for all glimpses.



Participants MVRNN RAM

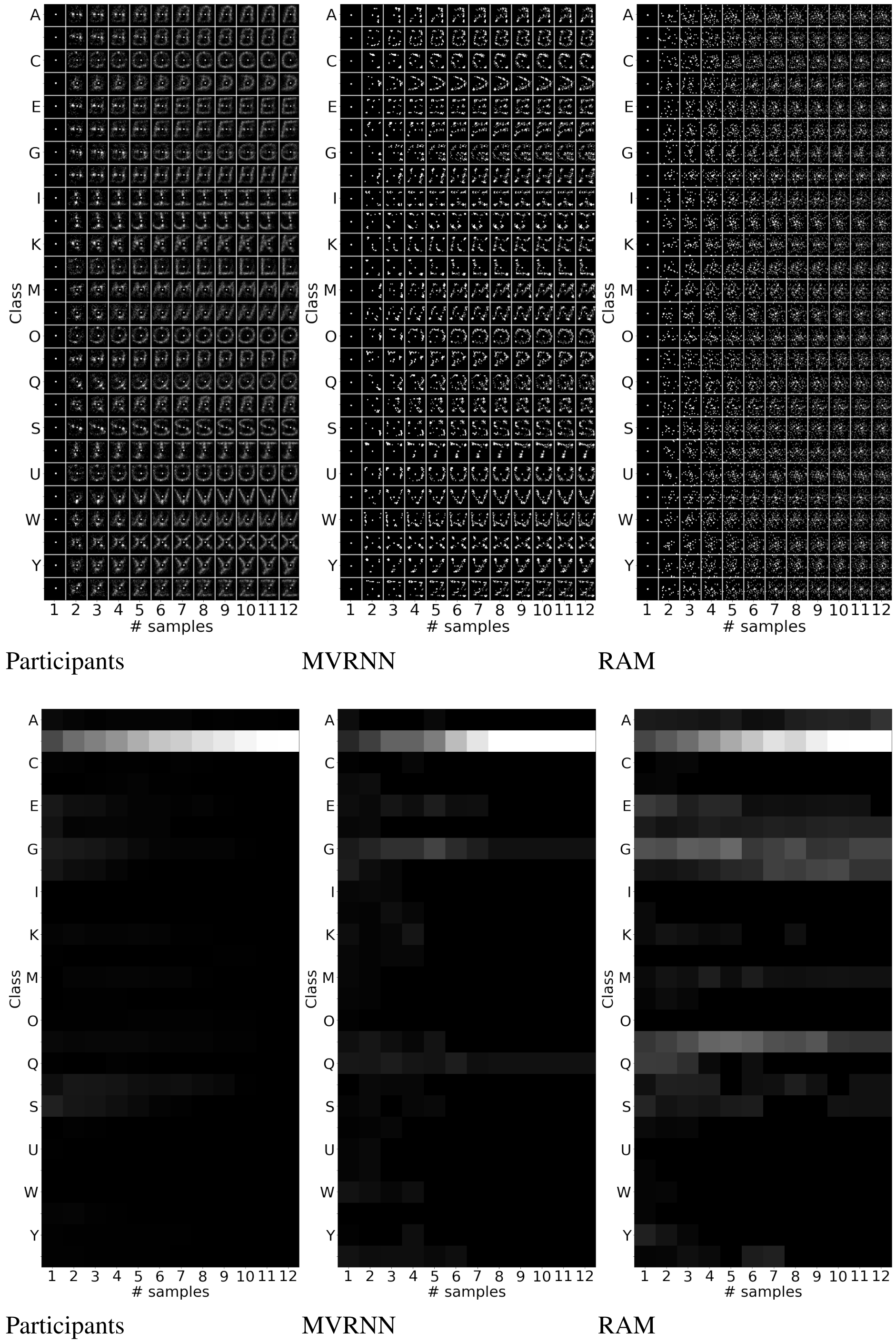
Participants MVRNN RAM

Figure 19.: (a)–(c) Distribution of sampling locations (or fixation maps) for each numeral and each sampling instant. (d)–(f) Class distribution for class ‘9’. Qualitatively, the participants’ fixation maps are more similar to MVRNN’s than RAM’s. The distributions are averaged over all stimuli (MVRNN and RAM) and all stimuli and participants (True) shown for MNIST. Each row corresponds to a class, each column corresponds to a sampling instant which increases from left to right.

5.3.3 Evaluation results

Evaluation for accuracy.

When both the classification and the pattern completion modality are trained end-to-end as in M1 and M2, NLL increases (ref. Tables 13.,14.). As the model is trained



Participants MVRNN RAM

Participants MVRNN RAM

Figure 20.: (a)–(c) Distribution of sampling locations (or fixation maps) for each numeral and each sampling instant. Qualitatively, the participants' fixation maps are more similar to MVRNN's than RAM's. (d)–(f) Class distribution for class 'B'. The distributions are averaged over all stimuli (MVRNN and RAM) and all stimuli and participants (True) shown for EMNIST uppercase. Each row corresponds to a class, each column corresponds to a sampling instant which increases from left to right.

to learn generation and classification tasks at the same time, the model is not able to perform well, due to which the accuracy in the generation modality lowers. When the

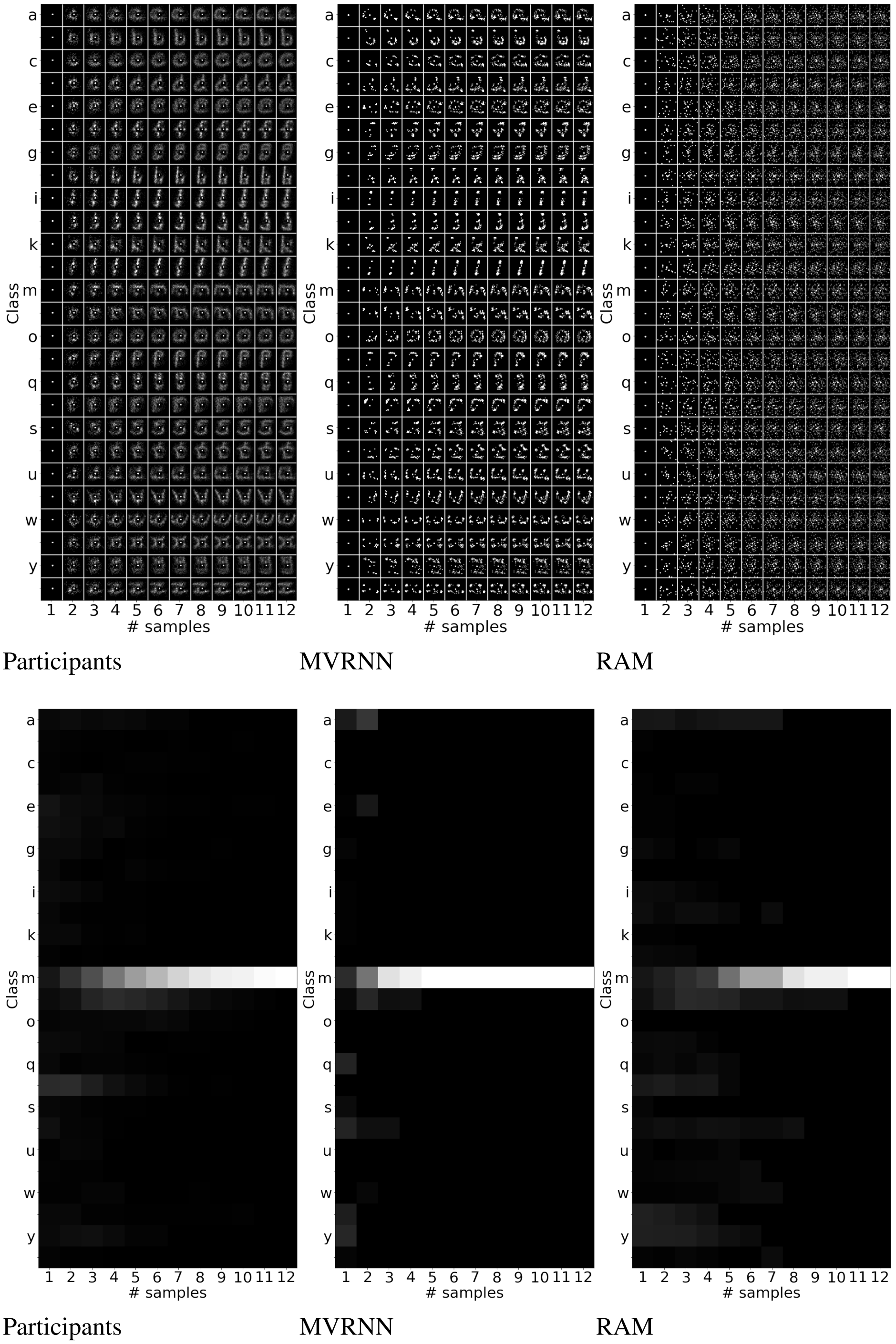


Figure 21.: (a)–(c) Distribution of sampling locations (or fixation maps) for each numeral and each sampling instant. Qualitatively, the participants' fixation maps are more similar to MVRNN's than RAM's. (d)–(f) Class distribution for class 'm'. The distributions are averaged over all stimuli (MVRNN and RAM) and all stimuli and participants (True) shown for EMNIST lowercase. Each row corresponds to a class, each column corresponds to a sampling instant which increases from left to right.

pattern completion and the classification modalities are trained separately, as the model is trained to learn the generation task only, the NLL is the lowest (ref. Table 13.,14.).

The classification accuracy from M1 is higher than M2 in all cases (ref. Table 13.,14.). In M1, the classification modality shares parameters with the generation modality, whereas in M2, the classification modality does not share parameters with the generation modality, though in both cases the generation modality shares parameters with the classification modality. Thus, the generation modality improves the classification results in M1 when compared to M2. The classification accuracy for M3 is very close to M1 and the classification accuracy for M4 is the lowest (ref. Table 13.,14.). M3 is a CNN based classifier, thus gives better classification accuracy when compared to M4, which is a RNN based classifier.

Evaluation for the sequence of locations.

Results from comparing the fixation maps from RAM and our model (M1) with the collected data are shown in Table 12.. KL is higher due to its sensitivity to zero values. This implies several locations are sampled by the participants (as there are multiple participants for one stimuli) but not by RAM or MVRNN. KL is lower for MVRNN than RAM for most cases. SIM and CC are either higher for MVRNN than RAM, or comparable for both the models. We obtain similar results for M2 and M3 as well (ref. supplemental material).

Clearly, between MVRNN and RAM, the fixation maps generated by the former are more similar to those generated by the participants. Visualization of the fixation maps in Fig. 17., 19., 20. and 21. shows higher similarity of the fixation map obtained from our model when compared to the participants data. As there are multiple participants for one stimuli, it appears like there are many more points for participants than for RAM or MVRNN in the visualization.

As MVRNN is based on saliency computed using prediction error and the human brain is closely linked with predictive coding [Friston, 2010], this can possibly explain higher similarity of the fixation maps for MVRNN. These experiments can be used as a baseline for evaluating locations sampled by an attention model.

Evaluation for efficiency.

As a participant can select multiple classes at any instant, for the proposed and RAM model, instead of predicting one class based on the highest probability, we consider the mean probability over all the classes as a threshold and predict the set of classes with probabilities greater than the threshold. We calculate the sampling number after which the participant and the models select only the correct class.

The average number of samplings required by a participant to accurately predict a class is quite low. On average, it takes 4.2, 4.7, 4.9 samples corresponding to 14.5%, 16.7% and 16.8% of image area respectively for MNIST, EMNIST uppercase and lowercase images. These results highlight the efficiency of the human visual reasoning system, albeit at a lower resolution than eye tracking data but with less noise and variability. These empirical results may be useful for designing attention-based models for real-world applications.

RAM requires 3.7, 8.5, 7.6 samples to recognize MNIST numerals, uppercase and lowercase EMNIST alphabets, which correspond to 12.0%, 23.4%, 21.4% of image area respectively. Thus, in comparison to the participants, under the same experimental conditions, RAM is less efficient.

Our model requires 2.0, 4.5, 4.2 samples to recognize MNIST numerals, uppercase and lowercase EMNIST alphabets, which correspond to 8.0%, 15.7%, 14.1% of image area respectively.

As seen from Fig. 18., in order to get the same accuracy, our model requires lesser percentage of observable data compared to RAM and the participants. Thus, our proposed model is more efficient in comparison to the participants and the RAM model. This is also validated by the class distribution plots shown in Figs. 19.–21.(d–f). We also observe that the classification accuracy over samples/glimpses plots for RAM and MVRNN is mostly flat (ref. Fig. 18.). This is because we are using a threshold to select multiple classes from these models as stated above, the true class gets selected in most of the glimpses, which does not change the classification accuracy over glimpses. The percentage area observation increases with glimpses/samples for RAM and the participants, but it saturates after a number of glimpses for MVRNN especially in Fig. 18.a. As there is no inhibition of return applied to our model during sampling, the model ends up selecting nearby locations of already sampled locations and thus this pattern is observed.

5.4 Conclusions

We proposed an attention-based agent model for handwritten numeral/alphabet recognition via a sequence of glimpses. The attention is driven by the agent's sensory prediction (or generation) error. Thus, at each sampling instant, the agent has to complete and classify the partial sequence observed till that instant. Very few end-to-end attention-based models reported in the literature perform generation and classification of handwritten numerals/alphabets jointly. Our agent model is learned by jointly minimizing the classification and generation errors. Three variants of this model are evaluated on benchmark datasets. Their accuracies are comparable and correlate with the model size. Our experiments reveal that the proposed model is more data-efficient in handwritten numeral/alphabet recognition than human participants as well as a highly-cited attention-based reinforcement model, under the same conditions and stimuli. Qualitatively, the participants' fixation maps are more similar to our model's fixation maps than the reinforcement model's. To the best of our knowledge, this is the first attention-based end-to-end agent of its kind for recognition via generation, with high degree of accuracy and efficiency.

Chapter 6

A Multimodal Predictive Agent Model for Human Interaction Generation

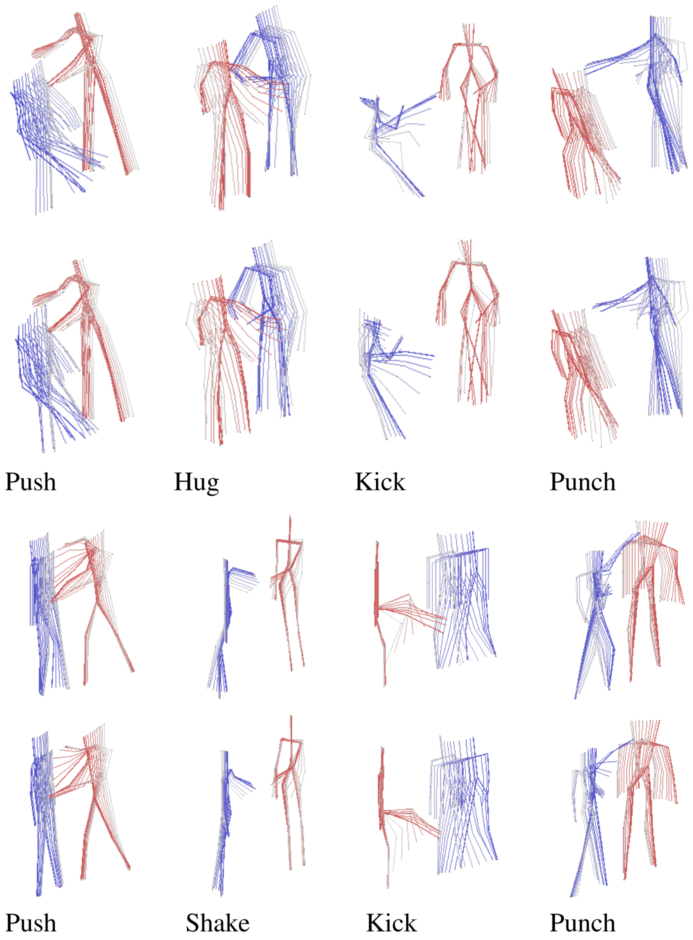
Abstract: Perception and action are inextricably tied together. We propose an agent model which consists of perceptual and proprioceptive pathways. The agent actively samples a sequence of percepts from its environment using the perception-action loop. The model predicts to complete the partial percept and propriocept sequences observed till each sampling instant, and learns where and what to sample from the prediction error, without supervision or reinforcement. The model is implemented using a multimodal variational recurrent neural network. The model is exposed to videos of two-person interactions, where one person is the modeled agent and the other person's actions constitute its visual observation. For each interaction class, the model learns to selectively attend to locations in the other person's body. The proposed attention-based agent is the first of its kind to interact with and learn end-to-end from human interactions, and generate realistic interactions with performance comparable to models without attention and using significantly more computational resources.

6.1 Introduction

An important property of human visual system that fosters efficiency is that one does not tend to process a whole spatiotemporal observation in its entirety at once. Instead humans focus attention selectively, in space and time, on parts of the observation to acquire information when and where it is needed, and combine information from different fixations over time to build up an internal representation of the observation [Rensink, 2000], guiding future eye movements and decision making.

Inspired by the human visual system, we propose a predictive agent[1] model which observes its visual environment via a sequence of glimpses. The agent is implemented in software; its actions are limited to sampling the visual environment and its own body

[1] An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators [Russell and Norvig, 2020]. There are many applications of such agent (e.g., [Banerjee and Chandrasekaran, 2010a,Banerjee and Chandrasekaran, 2010b,Najnin and Banerjee, 2017, Kapourchali and Banerjee, 2019, Kapourchali and Banerjee, 2020]).


Push

Hug

Kick

Punch

Push

Shake

Kick

Punch

Figure 22.: First and second rows show the actual and predicted data respectively for interactions push, hug, kick and punch from SBU Kinect interaction dataset. As the videos are short in length, continuous frames are shown. Third and fourth rows show the actual and predicted data respectively for interactions push, shake hands, kick and punch from K3HI interaction dataset. As the videos are longer in length, the frames are shown in intervals. Older frames are lighter in shade than more recent frames.

movements. The predictive agent actively makes inferences (predictive and causal), acts and learns by minimizing sensory prediction error in a perception-action loop. The model is unsupervised, and does not require reinforcement or utilities/values of states.

We apply the model for forecasting human interactions using 3D skeletal data. Interaction forecasting is a challenging problem as the model has to learn how the behavior of one person determines the behavior of the other. Spatiotemporal relations between different skeletal joints of a person as well as the two interacting persons have to be learned for accurate prediction. The ability to model dynamics of human interaction is useful for applications such as video surveillance, human-robot interaction, assistive robotics, and robotic surveillance. Though a large volume of work has been done on predicting actions using 3D skeletal data of a single person (e.g., [Ghosh et al., 2017, Li et al., 2018a, Chiu et al., 2019, Bütepage et al., 2018, Fragkiadaki et al., 2015, Xu et al., 2019]) as well as predicting human motion in crowded scenes (e.g., [Hoshen, 2017, Vemula et al., 2018, Varshneya and Srinivasaraghavan, 2017, Fernando et al., 2018]), much less has been done on predicting interaction of two persons using 3D skeletal data.

In this paper, we model the environment from the perspective of one of the interacting persons; the other person constitutes his environment. The novelty of our approach is threefold: (1) the modeled person (agent) learns to sample (or attend to) the most informative (or salient[2]) locations of the other persons body using a saliency map at each glimpse; (2) taking into account the past observations and its learned knowledge, the agent completes the entire perceptual and proprioceptive patterns after each glimpse; and (3) the pattern completion component in our agent is a multimodal generative model where the prediction error in a perceptual modality provides the observation for the proprioceptive modality. Attending the environment selectively introduces sparsity in the agent's observations, leading to efficiency. To the best of our knowledge, the proposed agent is the first of its kind to interact with and learn end-to-end from two-person interaction environments, with performance comparable to models without attention that uses significantly less sparse observations.

6.2 Related Work

A taxonomy of the models used for generating actions with 3D skeletons is presented below.

```
3D skeletal data generation models
└─Single-person action generation
```

[2] Saliency is a property of each location in a predictive agent's environment. The attention mechanism is a function of the agent's prediction error [Spratling, 2012, Banerjee and Dutta, 2014b, Najnin and Banerjee, 2017, Kapourchali and Banerjee, 2019, Kapourchali and Banerjee, 2020]. Other definitions of saliency (e.g., [Dutta and Banerjee, 2015, Dutta et al., 2016]) are not relevant to this paper.

    - Non-attentional models [Ghosh et al., 2017, Xu et al., 2019, Fragkiadaki et al., 2015, Li et al., 2018a, Chiu et al., 2019, Zhou et al., 2018, Bütepage et al., 2018, Barsoum et al., 2018, Lin and Amer, 2018, Gui et al., 2018]
    - Attentional models [Vinayavekhin et al., 2018]
  - Two-person interaction generation
    - Non-attentional models [Huang and Kitani, 2014]
    - Attentional models [Our proposed model]

The model in [Huang and Kitani, 2014] frames dual agent interaction as an optimal control problem by observing actions from one agent and predicting actions of the other agent. It does not model the observing agent's movement and predicts for short term only, unlike our proposed model. Work on predicting dual agent interactions using 3D skeletal data is limited. Most works report predicting motion of a single person using 3D skeletal data.

Few models have been proposed with attention mechanism for generating 3D skeletal data. The model in [Vinayavekhin et al., 2018] predicts the 3D skeletal data of a person using a temporal attention layer which generates an attention parameter at each time step. In this model, attention is defined by internal parameters and is not a function of the model's sensory prediction error, making it difficult to interpret the model's behavior. It also requires a fixed length of the input sequence to be observed in order to calculate an attention value for each time step, which may not be realistic for online application. We propose a novel attention mechanism based on sensory prediction error, that can complete the observation from any time step, with an interpretable behavior.

6.3 Models and Methods

This section defines the problem and describes the proposed agent model.

6.3.1 Problem Statement

Let $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(n)}\}$ be a set of observable variables representing an environment in n modalities. The variable representing the i-th modality is a sequence: $\mathbf{X}^{(i)} = \langle X_1^{(i)}, X_2^{(i)}, \ldots, X_T^{(i)} \rangle$, where T is the sequence length. Let

$\mathbf{x}_{\leq t} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ be a partial observation of $\mathbf{X}$ such that $\mathbf{x}^{(i)} = \langle x_1^{(i)}, \ldots, x_t^{(i)} \rangle$, $1 \leq t \leq T$. We define *pattern completion* as the problem of generating $\mathbf{X}$ as accurately as possible from its partial observation $\mathbf{x}_{\leq t}$.

Given $\mathbf{x}_{\leq t}$ and a generative model p_θ with parameters θ and latent variables $z_{\leq t}$, the generative process of $\mathbf{X}$ is:

$$p_\theta(\mathbf{X}|\mathbf{x}_{\leq t}) = \int p_\theta(\mathbf{X}|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_\theta(z_{\leq t}) dz \tag{6.1}$$

At any time t, the objective for pattern completion is to maximize the log-likelihood of $\mathbf{X}$, i.e. $\arg\max_\theta \int log(p_\theta(\mathbf{X}|\mathbf{x}_{\leq t}, z_{\leq t}; \theta) p_\theta(z_{\leq t}))dz$.

6.3.2 Agent Architecture

The proposed predictive agent architecture comprises of five components: environment, observation, pattern completion, action selection, and learning. See Fig. 23.a.

Environment. The environment is the source of sensory data and is dynamic (time-varying).

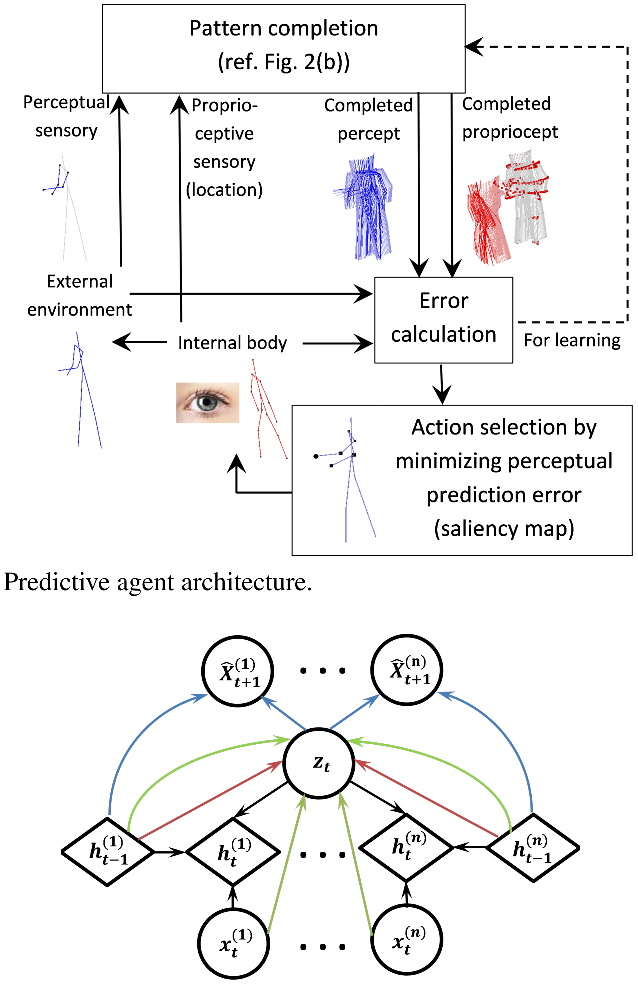
Observation. The agent interacts with the environment via a sequence of glimpses. The observations, sampled from the environment at each glimpse, are in two modalities: perceptual[3] and proprioceptive[4]. In the context of interaction generation, we define perceptual and proprioceptive sensory observations for an interacting person as follows.

Perceptual sensory observation. Perceptual sensory reports the visual observation at some location or region in the environment. $\mathbf{x}^{(1)} = \langle x_1^{(1)}, \ldots, x_T^{(1)} \rangle$, where $x_t^{(1)} \in \mathbb{R}^{3 \times N}$ denotes the other person's N 3D skeletal joints at time t.

Proprioceptive sensory observation. Proprioceptive sensory reports the

[3] Perception is the mechanism that allows an agent to interpret sensory signals from the external environment [Han et al., 2016].

[4] Proprioception is perception where the environment is the agent's own body. Proprioception allows an agent to internally perceive the location, movement and action of parts of its body [Han et al., 2016].



Predictive agent architecture.

Pattern completion model.

Figure 23.: (a) Components of the proposed agent. The red skeleton is the agent's own body while the blue is that of the other person. (b) Graphical illustration of all operations of the multimodal VRNN used for pattern completion. Red arrows show computation of the conditional prior, blue arrows show the generation process, black arrows show the updating process of the RNN's hidden states, and green arrows show the inference of the approximated posterior.

activations of the agent's joint muscles due to body movement and oculomotor muscles due to fixation. The activations of joint muscles over time (or body proprioception sequence) is $\mathbf{x}^{(2)} = \langle x_1^{(2)}, \ldots, x_T^{(2)} \rangle$, where $x_t^{(2)} \in \mathbb{R}^{3\times N}$ denotes N 3D skeletal joints at time t. The activation of oculomotor muscles over time (or visual proprioception sequence) is represented by the sequence of fixation locations in the environment, denoted as $\mathbf{x}^{(3)} = \langle x_1^{(3)}, \ldots, x_T^{(3)} \rangle$, where $x_t^{(3)} \in \{0,1\}^M$ is the activation at time t of skeletal joints reduced to M fixated regions (see Fig. 24.).

Pattern completion. A multimodal variational recurrent neural network (VRNN)

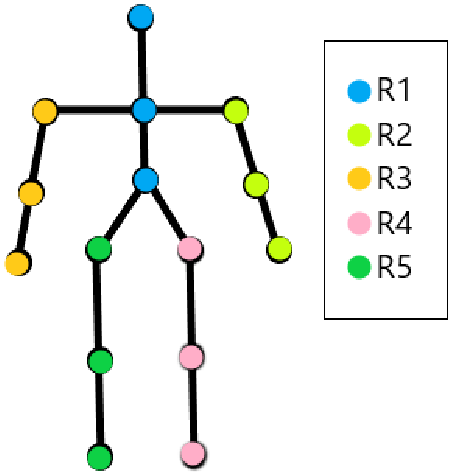


Figure 24.: The M (=5) regions in the 3D human skeleton.

for variable length sequences is used for completing the pattern for the three modalities (see Fig. 23.b). The two processes involved in the operation of a VRNN are recognition and generation [Chung et al., 2015].

Recognition (Encoder). The recognition model, $q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t})$, is a probabilistic encoder [Kingma and Welling, 2013]. Given the observations $\mathbf{x}_{\leq t}$, it produces a Gaussian distribution over the possible values of the code z_t from which the observations $\mathbf{x}_{\leq t}$ could have been generated. The recognition model consists of three RNNs, each with one layer of long-short term memory (LSTM) units. Each RNN generates the parameters for the approximate posterior distribution $(\mu_{z,t}^{(i)}, \sigma_{z,t}^{(i)})$ and the prior distribution $(\mu_{0,t}^{(i)}, \sigma_{0,t}^{(i)})$ for each modality i ($i = 1, 2, 3$), as in [Chung et al., 2015]. The parameters from each modality and for each distribution are combined using product of experts (PoE), as in [Wu and Goodman, 2018], to generate the joint distribution parameters (see Fig. 23.b) for both the prior $p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})$ and the approximate posterior $q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t})$ given by $(\mu_{0,t}, \sigma_{0,t})$ and $(\mu_{z,t}, \sigma_{z,t})$ respectively. The recognition model can be formulated as:

$$[\mu_{0,t}^{(i)}\ \sigma_{0,t}^{(i)}] = \varphi_{\tau}^{prior}(h_{t-1}^{(i)}), \quad [\mu_{z,t}^{(i)}\ \sigma_{z,t}^{(i)}] = \varphi_{\tau}^{enc}(x_t^{(i)}, h_{t-1}^{(i)})$$

$$z_t \sim \mathcal{N}(\mu_{0,t}, \sigma_{0,t}), \quad z_t|x_t \sim \mathcal{N}(\mu_{z,t}, \sigma_{z,t})$$

$$\sigma_{0,t} = \Big(\sum_i \sigma_{0,t}^{(i)^{-2}}\Big)^{-1}, \quad \sigma_{z,t} = \Big(\sum_i \sigma_{z,t}^{(i)^{-2}}\Big)^{-1}$$

$$\mu_{0,t} = \Big(\sum_i \mu_{0,t}^{(i)} \sigma_{0,t}^{(i)^{-2}}\Big)\sigma_{0,t}, \quad \mu_{z,t} = \Big(\sum_i \mu_{z,t}^{(i)} \sigma_{z,t}^{(i)^{-2}}\Big)\sigma_{z,t}$$

where φ_{τ}^{prior} and φ_{τ}^{enc} are functions representing neural networks. It is assumed that the prior z_t and the approximated posterior $z_t|x_t$ are sampled from an isotropic multivariate Gaussian distribution.

Generation (Decoder). The generative model, $p_\theta(\mathbf{X}_{t+1}|z_{\le t}, \mathbf{x}_{\le t})$, generates the data from the latent variables, $z_{\le t}$, at each time step. The generative model has three RNNs with one layer of hidden LSTM units. Each RNN generates the parameters of the distribution of the sensory data for a modality. The sensory data is sampled from this distribution which can be multivariate Gaussian or Bernoulli. In our model, $X_{t+1}^{(1)}|z_t, X_{t+1}^{(2)}|z_t$ are sampled from an isotropic multivariate Gaussian distribution and $X_{t+1}^{(3)}|z_t$ from a Bernoulli distribution. The generative model can be formulated as:

$$h_t^{(i)} = f_\theta(z_t, x_t^{(i)}, h_{t-1}^{(i)}), \quad [\mu_{x^{(i)},t}^{(i)}\ \sigma_{x^{(i)},t}^{(i)}] = \varphi_{\tau}^{dec}(z_t, h_t^{(i)}).$$

For Gaussian distribution, $X_{t+1}^{(i)}|z_t \sim \mathcal{N}(\mu_{x^{(i)},t}^{(i)}, \sigma_{x^{(i)},t}^{(i)})$. For Bernoulli distribution, $X_{t+1}^{(i)}|z_t = f_\sigma(h_t^{(i)})$. Here φ_{τ}^{dec}, f_θ are functions representing neural networks, and f_σ is a sigmoid function. The above equations facilitate one step ahead prediction. Beyond time t, for long term predictions or pattern completion, the input is the prediction from the previous time steps. Pattern completion is done at every time step.

Action selection. In the proposed agent model, action selection is to decide which location in the environment to sample from. The environment is a 3D skeleton of the interacting person. As the movement of a joint in the skeleton is dependent on its adjacent joints, we cluster the N skeletal joints into M regions (see Fig 24.). Location refers to all the skeletal joints in top k_t salient regions, $1 \leq k_t \leq M$, and k_t is not fixed. At any time step, the agent selects k_t regions using a threshold. At any time, there are $\sum_{k_t=1}^{M-1} \binom{M}{k_t}$ possible actions to choose from.

An action at time t is generated as a function of the saliency map. We denote the saliency map at time t as $S_t \in \mathbb{R}^N$ and the value of the saliency map at region ℓ as $S_t^{(\ell)}$. The saliency map is a function of the prediction error computed as $S_t = \|X_t^{(1)} - \hat{X}_t^{(1)}\|_1$, where $X_t^{(1)}, \hat{X}_t^{(1)} \in \mathbb{R}^{3\times N}$ are the true and predicted perceptual data (skeleton joint coordinates) respectively, and $\|.\|_1$ denotes L_1 norm. We consider saliency over $M = 5$ regions in the skeleton. This region-based saliency map, $S_t^\ell \in \mathbb{R}^M$, is obtained by averaging the saliencies over the joints in each region. The region ℓ is considered salient if $S_t^\ell \geq \frac{1}{M}\sum_{r=1}^{M} S_t^r$. Thus, at any time, at least one region will be salient. A variable number of salient regions at each time step is more effective. Setting the number of salient regions to a constant value might occasionally lead to selection of regions with low saliency or discard regions with high saliency as saliency, S_t, is a function of time, the agent's observations and its predictive model. In the proposed model, for the salient joints, the observation is sampled from the environment; for the non-salient joints, the observation is predicted from the last time step.

The salient regions at any time t is the proprioceptive observation $x_{t+1}^{(3)}$ for time $t+1$. Therefore, the salient regions at $t = 0, 1, 2, \ldots, T-1$ constitutes the proprioceptive pattern $\mathbf{X}^{(3)}$. Hence, prediction error (saliency) guides the sampling of the observations in our model. Unlike typical multimodal models, the modalities in our model interact at the observation level as the perceptual prediction error provides the observation for the proprioceptive modality.

The agent learns a policy to generate the proprioceptive pattern or the sequence of expected salient locations by minimizing the proprioceptive prediction error (first term in

Eq. 6.2 for $i = 3$). This error, at any time, is a function of the difference between predicted fixation location from the learned policy and the most salient location in the scene.

The most salient location is the most informative location in the environment. These are the locations where the agent's prediction error is the highest given all the past observations. The agent attends to these locations to update its internal model.

Learning. The recognition and generative model parameters are jointly learned by maximizing the ELBO for the multimodal VRNN. This objective function, obtained by modifying the objective for multimodal VAE (Eq. 2 in [Wu and Goodman, 2018]) with VRNN (Eq. 1 in [Chung et al., 2015]), is as follows:

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T-1}\Big[\sum_{i=1}^{n}\lambda_i \log p_\theta(X_{t+1}^{(i)}|z_{\leq t}, x_{\leq t}^{(i)}) \\ - \beta \mathrm{KL}[q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})]\Big]\Big] \quad (6.2)$$

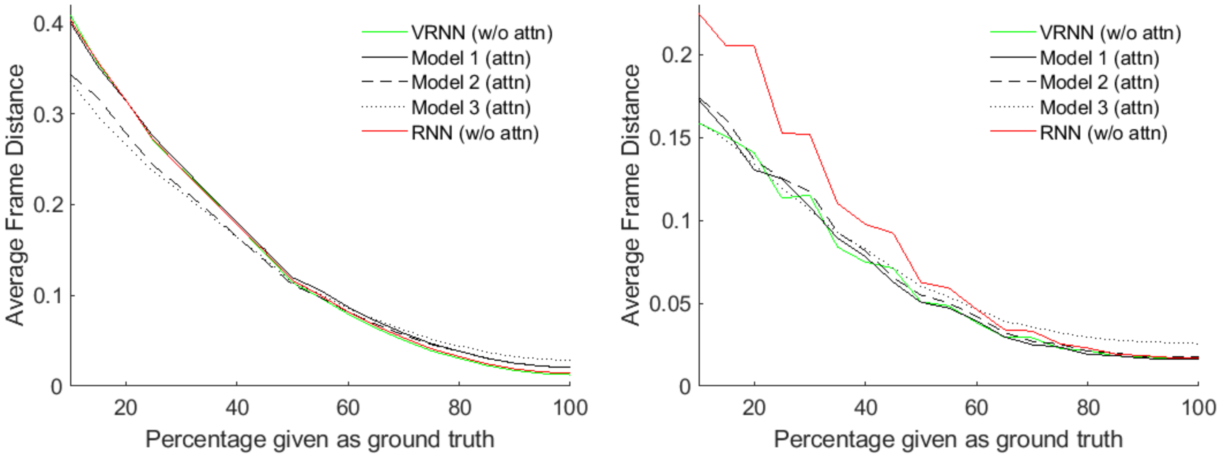
where n is the number of modalities, the first term for $i = 1, 2, 3$ is the expected negative prediction error for the three modalities. The KL-divergence is a regularizer to prevent overfitting during training.

The negative of the ELBO is also referred to as negative log-likelihood (NLL). In this paper, we refer to the negative of the first term in Eq. 6.2 for $i = 1$ and $i = 2, 3$ as perceptual NLL and proprioceptive NLLs respectively.

6.4 Experimental Results

6.4.1 Datasets

SBU Kinect Interaction Dataset [Yun et al., 2012] is a two-person interaction dataset comprising of eight interactions: approaching, departing, pushing, kicking, punching, exchanging objects, hugging, and shaking hands. The data is recorded from 7 participants forming a total of 21 sets such that each set consists of a unique pair of



SBU Kinect Interaction Dataset

K3HI Interaction Dataset

Figure 25.: AFD averaged over all actions and each dataset for different percentage of ground truth given as input. For any percentage p, $p\%$ of the actual data is given as input and the prediction is considered as input for the rest of the time steps.

Table 15.: Performance comparison for different versions of the proposed model and other models for different interactions on the SBU Kinect Interaction dataset for one step ahead prediction. The reported AFD is the average of the perceptual (Perc.) AFD and proprioceptive (Prop.) AFD averaged over all examples in the test set and all the train-test splits. The visual proprioceptive performance is shown in the last column.

Interaction	Perc. error and Body Prop. error (AFD)						Visual Prop. (%)
	[Huang and Kitani, 2014]	RNN (w/o attn)	VRNN (w/o attn)	Model 1 (VRNN, attn)	Model 2 (VRNN, attn+ true policy)	Model 3 (VRNN, attn+ pred. policy)	Model 2 (VRNN, attn+ true policy)
Approaching	-	0.0097	0.0082	0.0128	0.0138	0.0189	61.08
Departing	-	0.0117	0.0098	0.0140	0.0150	0.0199	61.41
Kicking	0.660	0.0210	0.0192	0.0358	0.0360	0.0411	61.77
Pushing	0.413	0.0142	0.0125	0.0212	0.0215	0.0267	64.79
Shaking	0.389	0.0094	0.0079	0.0130	0.0130	0.0276	62.25
Hugging	0.504	0.0197	0.0181	0.0273	0.0272	0.0412	63.80
Exchanging	0.574	0.0111	0.0095	0.0136	0.0145	0.0195	65.16
Punching	0.510	0.0175	0.0159	0.0252	0.0258	0.0326	63.19
Average	0.508	0.0143	0.0126	0.0204	0.0208	0.0284	62.93

participants performing all actions. The dataset has approximately 300 interactions of duration 9 to 46 frames. The dataset is divided into five distinct train test split.

K3HI: Kinect-based 3D Human Interaction Dataset [Hu et al., 2013] is a two-person interaction dataset comprising of eight interactions: approaching, departing, kicking, punching, pointing, pushing, exchanging an object, and shaking hands. The data is recorded from 15 volunteers. Each pair of participants performs all the actions. The dataset has approximately 320 interactions of duration 20 to 104 frames. The dataset is divided into three distinct train test split.

Table 16.: Performance comparison for different versions of the proposed model and other models for different interactions on the K3HI dataset for one step ahead prediction. The reported AFD is the average of the perceptual (Perc.) AFD and proprioceptive (Prop.) AFD averaged over all examples in the test set and all the train-test splits. The visual proprioceptive performance is shown in the last column.

Interaction	Perc. error and Body Prop. error (AFD)					Visual Prop. (%)
	RNN (w/o attn)	VRNN (w/o attn)	Model 1 (VRNN, attn)	Model 2 (VRNN, attn+ true policy)	Model 3 (VRNN, attn+ pred. policy)	Model 2 (VRNN, attn+ true policy)
Approaching	0.0844	0.0912	0.0714	0.0735	0.0796	72.68
Departing	0.0065	0.0067	0.0097	0.0102	0.0130	69.44
Exchanging	0.0022	0.0024	0.0036	0.0038	0.0072	73.54
Kicking	0.0047	0.0049	0.0078	0.0078	0.0117	69.43
Pointing	0.0025	0.0028	0.0048	0.0048	0.0089	69.77
Punching	0.0038	0.0040	0.0064	0.0067	0.0101	70.35
Pushing	0.0035	0.0038	0.0062	0.0065	0.0090	66.99
Shaking	0.0019	0.0021	0.0031	0.0034	0.0065	67.86
Average	0.0137	0.0147	0.0141	0.0146	0.0182	70.00

Table 17.: Average percentage of saliency joints for both SBU Kinect Interaction and K3HI dataset for all cases in an interaction. This percentage is also the proportion of joints that are sampled from the observation (ground truth).

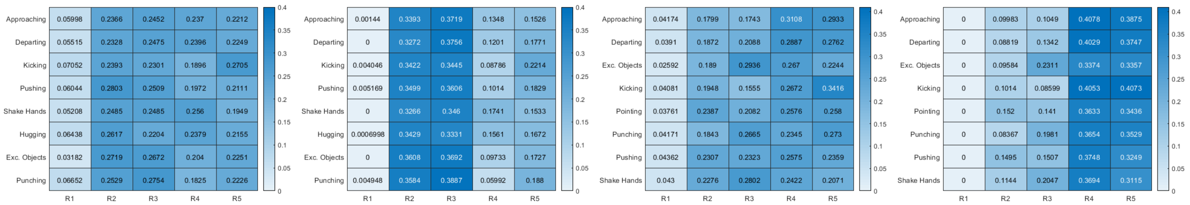
SBU	Approaching	Departing	Kicking	Pushing	Shaking	Exchanging	Punch	Hugging	Average
	46.21	46.99	44.87	47.54	47.23	47.36	47.80	47.45	46.93
K3HI	Approaching	Departing	Kicking	Pushing	Shaking	Exchanging	Punching	Pointing	Average
	45.19	46.27	43.79	48.06	48.11	50.00	47.57	47.70	47.09

6.4.2 Experimental setup

Each dataset consists of interactions where one person initiates an action and the other person reacts to it. In our experiments, we model one interacting person irrespective of its initiating or reacting nature. We consider 15 skeletal joints from each person for each dataset. Each skeletal joint is normalized before training.

Each modality in the agent architecture (ref. Fig. 23.b) has a recurrent hidden layer of 256 hidden units and a latent layer of 10 latent variables.

We use Adam optimizer with a learning rate of 0.001, and default hyper-parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ [Kingma and Ba, 2014]. A minibatch size of



	R1	R2	R3	R4	R5
Approaching	0.05998	0.2366	0.2452	0.237	0.2212
Departing	0.05515	0.2328	0.2475	0.2396	0.2249
Kicking	0.07052	0.2393	0.2301	0.1896	0.2705
Pushing	0.06044	0.2803	0.2509	0.1972	0.2111
Shake Hands	0.05208	0.2485	0.2485	0.256	0.1949
Hugging	0.06438	0.2617	0.2204	0.2379	0.2155
Exc. Objects	0.03182	0.2719	0.2672	0.204	0.2251
Punching	0.06652	0.2529	0.2754	0.1825	0.2226

SBU true dist.

	R1	R2	R3	R4	R5
Approaching	0.00144	0.3393	0.3719	0.1348	0.1526
Departing	0	0.3272	0.3756	0.1201	0.1771
Kicking	0.004046	0.3422	0.3445	0.08786	0.2214
Pushing	0.005169	0.3499	0.3606	0.1014	0.1829
Shake Hands	0	0.3266	0.346	0.1741	0.1533
Hugging	0.0006998	0.3429	0.3331	0.1561	0.1672
Exc. Objects	0	0.3608	0.3692	0.09733	0.1727
Punching	0.004948	0.3584	0.3887	0.05992	0.188

SBU predicted dist.

	R1	R2	R3	R4	R5
Approaching	0.04174	0.1799	0.1743	0.3108	0.2933
Departing	0.0391	0.1872	0.2088	0.2887	0.2762
Exc. Objects	0.02592	0.189	0.2936	0.267	0.2244
Kicking	0.04081	0.1948	0.1555	0.2672	0.3416
Pointing	0.03761	0.2387	0.2082	0.2576	0.258
Punching	0.04171	0.1843	0.2665	0.2345	0.273
Pushing	0.04362	0.2307	0.2323	0.2575	0.2359
Shake Hands	0.043	0.2276	0.2802	0.2422	0.2071

K3HI true dist.

	R1	R2	R3	R4	R5
Approaching	0	0.09983	0.1049	0.4078	0.3875
Departing	0	0.08819	0.1342	0.4029	0.3747
Exc. Objects	0	0.09584	0.2311	0.3374	0.3357
Kicking	0	0.1014	0.08599	0.4053	0.4073
Pointing	0	0.152	0.141	0.3633	0.3436
Punching	0	0.08367	0.1981	0.3654	0.3529
Pushing	0	0.1495	0.1507	0.3748	0.3249
Shake Hands	0	0.1144	0.2047	0.3694	0.3115

K3HI predicted dist.

Figure 26.: Salient region distribution (dist.) over all interactions shown in (a–d) averaged over all the examples in an interaction. True salient distribution obtained from the saliency map for SBU Kinect Interaction dataset is shown in (a) and for K3HI dataset is shown in (c). The predicted salient distribution obtained from the predicted joints as in proposed Model 2 are shown for SBU Kinect Interaction dataset and K3HI dataset in (b) and (d) respectively.

100 is used and number of training iterations is fixed at 25,000 (SBU Kinect) and 10,000 (K3HI). To avoid overfitting, we use a dropout probability of 0.8 at the generation layer (final layer). All the hyperparameters are determined experimentally.

For evaluation, we consider three variants of our model:

- **Model 1: VRNN with 2 modalities**. Perceptual and body proprioceptive are the two modalities. Here, $i = 1, 2$ in Eq. 6.2.

- **Model 2: VRNN with 3 modalities**. Ref. Section 6.3.2.

- **Model 3: VRNN with 3 modalities and perceptual input sampled from predicted visual proprioception**. This is a special case of Model 2. Here the perceptual input is sampled from the prediction, $\hat{X}_t^{(3)}$, instead of the true saliency map, S_t, at all time steps.

The difference between Model 1 and the other two models is the addition of the third modality $\mathbf{x}^{(3)}$ to the model. This difference will show the effect of adding modalities in the model. The difference between Model 3 and the other two models is the data from which the perceptual input is sampled. This difference will show how well the model learns to predict the salient joints.

We evaluate the model by comparing it with models without attention. The perceptual observation is sampled from the ground truth, $X_t^{(1)}$, at all time steps.

1. **RNN (without attention).** We use a standard LSTM encoder-decoder model and to generate data for two modalities: perceptual and body proprioceptive. The two modalities interact at the latent layer, where the latent variables are concatenated; it thus has a total of 20 latent variables.

2. **VRNN (without attention).** We use a variational LSTM autoencoder model to generate data for two modalities: perceptual and body proprioceptive. The two modalities interact at the latent layer, where the latent variables are combined using PoE.

For fair comparison, the number of layers and number of neurons are kept consistent for both models with respect to the proposed models.

We evaluate results from the perceptual modality ($i = 1$) and body proprioceptive modality ($i = 2$) using average frame distance (AFD), as in [Huang and Kitani, 2014]: $\frac{1}{T-1}\sum_{t=2}^{T}\|X_t^{(i)} - \hat{X}_t^{(i)}\|^2$, where $X_t^{(i)}$ and $\hat{X}_t^{(i)}$ are the true and predicted skeletal joint coordinates respectively at time t, and T is the sequence length.

We use percentage measure to evaluate the two proprioceptive modalities in Model 2. The measure reflects how good the learned policy (generated sequence of salient regions) is when compared to the true policy (true sequence of salient regions). At each time step, the true policy can generate multiple salient regions. We define the average percentage as:

$$\frac{1}{T-1}\sum_{t}\frac{No.\ of\ correctly\ predicted\ salient\ regions}{Total\ no.\ of\ salient\ regions}$$

6.4.3 Evaluation Results

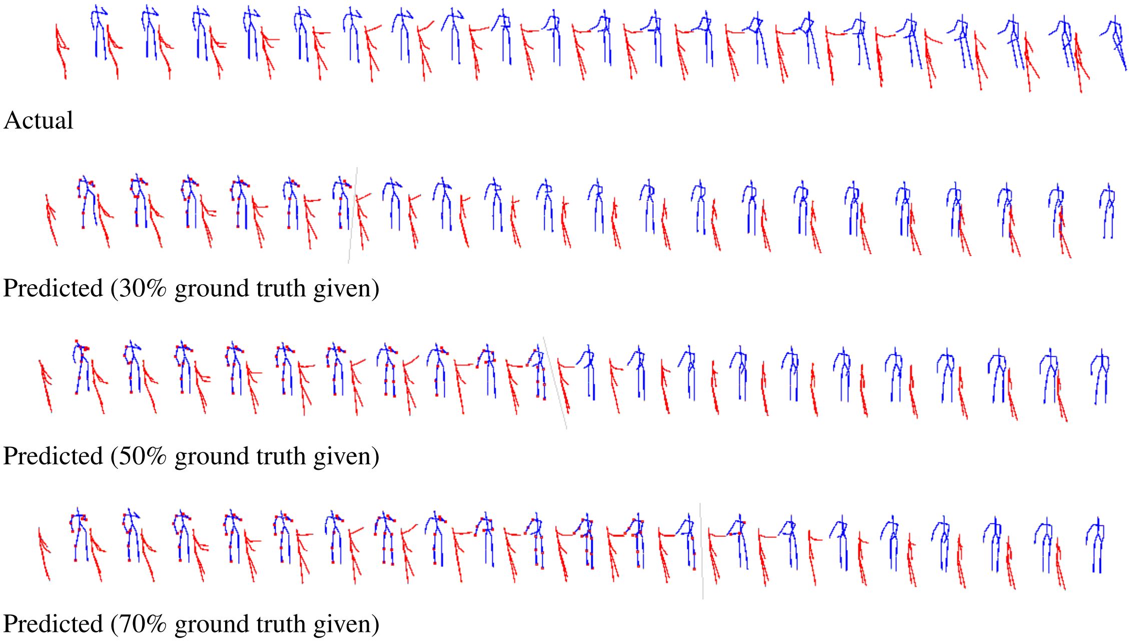
Fig. 22. shows one time step ahead prediction of the two skeletons (perception and body proprioception) for four kinds of interactions from each dataset. The prediction over space and time looks quite realistic for all the cases.

For long term predictions, the prediction improves exponentially with the percentage of data given as ground truth (see Fig. 25.). For SBU Kinect dataset, the performance of RNN (without attention), VRNN (without attention) and Model 1 (with attention) is slightly poorer than the proposed Model 2 (with attention) and Model 3 (with attention) until around 50% of the ground truth is given as the input. For ground truth $\geq 50\%$, the AFD for non-attention models and proposed Model 1 slightly improves compared to proposed Models 2 and 3. For K3HI dataset, the performance of RNN is poorer than Model 1, Model 2, Model 3, and VRNN until around 75% of the ground truth is given as the input. For ground truth $\geq 75\%$, the AFD for all the models except Model 3 are close. Thus, overall performance of attention and non-attention models are comparable.

RNN and VRNN without attention are more prone to error propagation as the predicted data is fed as the input for consecutive prediction whereas during training, the ground truth is fed as input to the model. Our model is more robust to noise as during training, for the non-salient joints, the predicted data is fed as the input for consecutive prediction. VRNNs though in general are more robust to noise, adding the proposed attention mechanism with sparsity can help in combating error propagation and improve long-term predictions. Detailed AFD for all the interactions for one time step ahead prediction are shown in Tables 15. and 16.. The AFD for all the interactions is lower than the results reported in [Huang and Kitani, 2014] for SBU Kinect dataset. This shows that our model is able to learn better representation of the underlying dynamics of interaction. For K3HI dataset, we are the first to report the AFD. Among the three variants of the proposed model, for one step ahead prediction, Model 1 performs the best. No significant difference in long-term prediction performance is observed among the three variants (see Fig. 25.). However, from a practical standpoint, Model 2 and Model 3 can be more useful as they can learn the policy and automatically determine salient regions for future time steps. So the agent can decide what action to take much earlier than the actual event occurs.

Predictions closer to the current time step are better, as observed from Figs. 27., 28.. There is continuity and the two predicted skeletons are well synchronized. The agent's predicted action or reaction at each time step also complies with the actual interaction.

It is also observed that the number of actual salient regions may change at each time step depending on the prediction error (highlighted with markers in the joints of the skeletons). This change may occur quite randomly depending on what the model has learned. Therefore, learning to predict the salient regions is a challenging task. However,



Actual

Predicted (30% ground truth given)

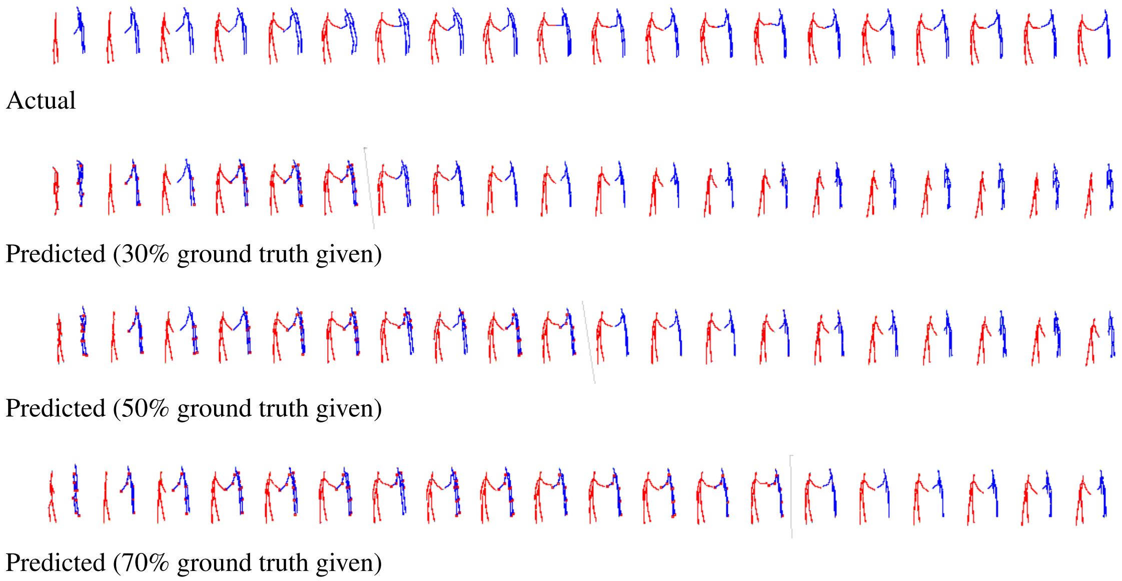
Predicted (50% ground truth given)

Predicted (70% ground truth given)

Figure 27.: The top row represents true skeletal data for the prediction at alternate time steps for SBU Kinect Intersection data for exchanging object. Each skeleton in rows 2, 3 and 4 shows one step ahead prediction until 30%, 50% and 70% of the ground truth is given (highlighted by the grey line) respectively. Beyond that, the model uses its own prediction as input for completing the patterns until the final time step is reached. The salient joints are marked red.

our model is able to predict them correctly most of the time (ref. Tables 15., 16., and Fig. 26.). Ideally, the joint trajectories that occur rarely are more difficult to learn, and hence more salient. Thus, the actual salient regions for punching, exchange objects, push, handshake and hug are mostly the hands while for action kicking its the legs. In our case, as the modeled agent can be the reacting or interacting agent, the salient region distribution is similar but not exactly the same as the ideal case.

We compute the average (over all the cases for an interaction) of the percentage of number of salient joints chosen by the model at each time step (ref. Table 17.). On average, for all the interactions, our approach considers less than 50% of the joints in a skeleton as observation to the model. For both datasets, the highest sparsity is for kicking, the lowest is for punching and shaking hands for SBU Kinect and K3HI datasets respectively. Selectively attending to fewer joints makes our model more efficient without compromising its accuracy.



Actual

Predicted (30% ground truth given)

Predicted (50% ground truth given)

Predicted (70% ground truth given)

Figure 28.: The top row represents true skeletal data for the prediction at every third instant for K3HI Intersection data for shaking hands. Each skeleton in rows 2, 3 and 4 shows one step ahead prediction until 30%, 50% and 70% of the ground truth is given (highlighted by the grey line) respectively. Beyond that, the model uses its own prediction as input for completing the patterns until the final time step is reached. The salient joints are marked red.

6.5 Conclusions

A multimodal predictive agent with perceptual and proprioceptive pathways is proposed. It completes the observed pattern for perceptual and proprioceptive modalities after each glimpse. The perceptual prediction error provides the observation for the proprioceptive modality. Experimental results using our agent for two-person interaction forecasting are comparable to non-attentional models even though our agent's observations have higher than 50% sparsity. The agent model is learned end-to-end in an unsupervised manner, without any reinforcement signal or utilities/values of states. This is the first work on an attention-based agent that actively samples its environment guided by prediction error and generates realistic 3D human skeleton interactions.

Chapter 7

An Attention-based Multimodal Predictive Agent: Exploiting Perception and Proprioception for Interaction Recognition via Generation

Abstract: We propose a general-purpose agent model consisting of perceptual and proprioceptive pathways in a multimodal setting. The agent actively samples its environment via a sequence of glimpses. The attention is driven by the agent's sensory prediction (or generation) error. At each sampling instant, the model predicts the observation class and completes the partially observed sequence. It learns where and what to sample by jointly minimizing the classification and generation errors. The model is evaluated on interaction classification and generation. It is exposed to videos of two-person interactions under two settings: (first person) one person is the modeled agent and the other person's body movements constitute its visual observation, and (third person) a spectator is the modeled agent and the two interacting persons' body movements constitute its visual observation. Three variants of the proposed model, and three ways of implementing action selection (where to attend to) for each variant are analyzed using benchmark datasets. We show that classification accuracy is comparable when sampling locations are determined from sensory prediction error or from learned weights (without involving prediction error), but the latter is less efficient in terms of model size. This is the first known attention-based agent to interact with and learn end-to-end from two-person interaction environments for recognition via generation, with high degree of accuracy and efficiency.

7.1 Introduction

An important property of human perceptual systems that fosters efficiency is that one does not tend to process a spatiotemporal observation in its entirety at once. Instead humans focus attention selectively, in space and time, on parts of the observation to acquire information when and where it is needed, and combine information from different

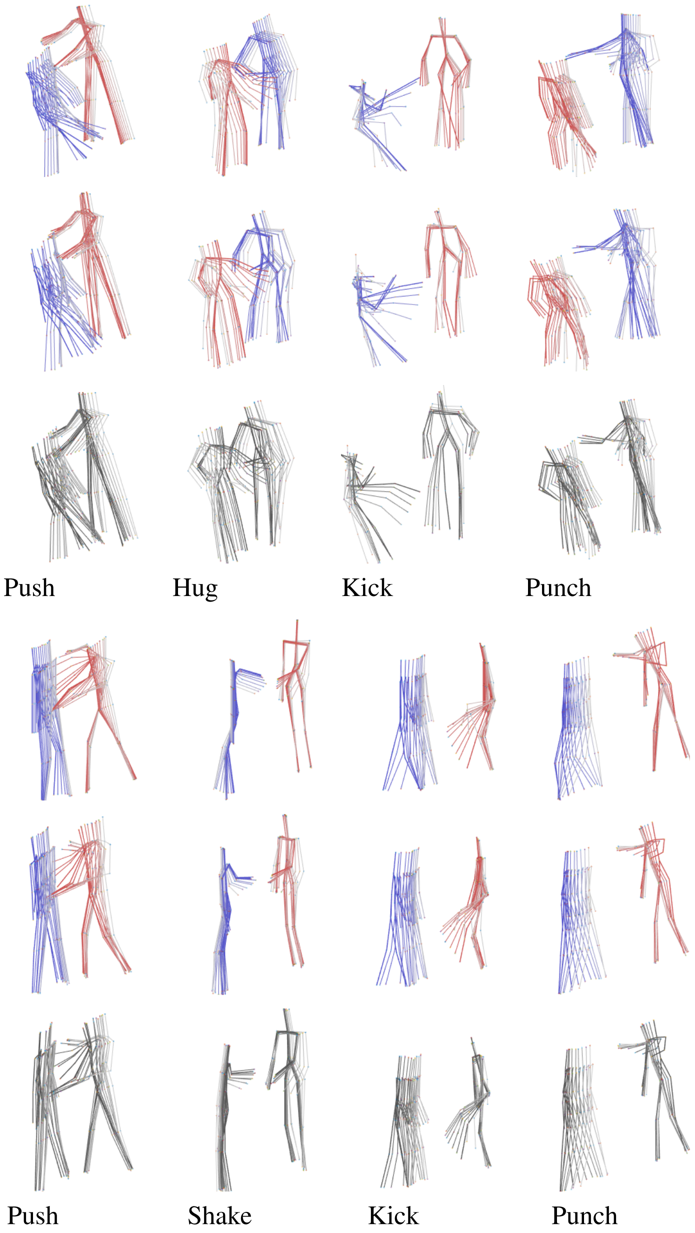


Figure 29.: First, second and third rows show the actual, generated first person and generated third person data respectively for four interactions from SBU Kinect interaction dataset. As the videos are short in length, continuous frames are shown. Fourth, fifth and sixth rows show the actual, generated first person and generated third person data respectively for four interactions from K3HI interaction dataset. As the videos are longer in length, the frames are shown in intervals. Older frames are lighter in shade than more recent frames.

fixations over time to build up an internal representation of the observation [Rensink, 2000], guiding future eye and body movements and decision making.

Inspired by this idea, we propose a predictive agent model which senses its environment as a sequence of samples. The agent is implemented in software; its actions are limited to sampling the visual environment and its own body movements. The predictive agent makes active inferences, acts and learns by minimizing sensory prediction error in a perception-action loop. Utility-based agents, that maximize reward to choose optimal actions, have dominated the field of AI for decades. Unfortunately, a reward signal is seldom present in real-world data. The proposed model does not require reinforcement or utilities/values of states which makes it more practical for real-world applications.

We apply the model for simultaneously classifying and forecasting human interactions using 3D skeletal data. Interaction forecasting is a challenging problem as the model has to learn how the behavior of one person determines the behavior of the other. Spatiotemporal relations between different skeletal joints of a person as well as the two interacting persons have to be learned for accurate prediction. The ability to recognize and model dynamics of human interaction is useful for video surveillance, human-robot interaction, assistive robotics, and robotic surveillance.

Prior work. A large volume of work has been reported on generating actions using a 3D skeleton (e.g., [Ghosh et al., 2017, Xu et al., 2019, Fragkiadaki et al., 2015, Li et al., 2018a, Chiu et al., 2019, Zhou et al., 2018, Bütepage et al., 2018, Barsoum et al., 2018, Lin and Amer, 2018, Gui et al., 2018, Chopin et al., 2021, Vinayavekhin et al., 2018, Baruah and Banerjee, 2020a]) and on generating human motion in crowded scenes (e.g., [Hoshen, 2017, Vemula et al., 2018, Varshneya and Srinivasaraghavan, 2017, Fernando et al., 2018, Adeli et al., 2020, Kothari et al., 2021]). Comparatively, much less has been reported on generating interaction of two persons using 3D skeletal data (e.g., [Huang and Kitani, 2014, Yao et al., 2018, Ng et al., 2020]). In these interaction models, the environment is viewed from one of two perspectives: first person (FP) where one of the interacting persons is the observer while the other constitutes his environment (e.g., [Huang and Kitani, 2014, Ng et al., 2020]), or third person (TP) where a person, such as an audience, is the observer and the two interacting persons constitute his environment (e.g., [Yao et al., 2018]). The models in [Huang and Kitani, 2014, Ng et al., 2020] generate the 3D pose of one of the skeletons upon observing the motions of the other. Given a sequence of 3D skeletal interaction of two persons, the model in [Yao et al., 2018] generates their 3D skeletal interaction data for future time-steps. Some of these models use attention. For example, temporal attention is used in [Vinayavekhin et al., 2018, Fernando et al., 2018], an attention mechanism that weighs different modalities is used in [Hoshen, 2017, Vemula et al., 2018], and spatiotemporal attention is used in [Varshneya and Srinivasaraghavan, 2017].

There is also a large volume of work on two-person interaction classification from videos (e.g., [Yu et al., 2020]) and skeletal data (e.g., [Li et al., 2018b, Manzi et al., 2018, Song et al., 2017, Fan et al., 2018, Le et al., 2018, Baradel et al., 2017, Qin et al., 2020, Li and Leung, 2019]). Some of these models incorporate temporal [Yu et al., 2020, Baradel et al., 2017], spatial and temporal [Song et al., 2017], or multilayer feature [Fan et al., 2018] attention mechanism.

Contributions. In this paper, we propose an attention-based agent model that learns to classify two-person interaction from 3D skeletal data by generating them. The novelty of this work is five-fold:

(1) The proposed model implements a perception-action loop as the optimization of an objective function. The action (attention) is modeled as proprioception in a multimodal setting and is guided by the perceptual prediction error, not by reinforcement.

(2) At each sampling instant, the model simultaneously predicts the interaction class and the motion of both 3D skeletons, in both FP and TP environments. Typically FP models, such as [Huang and Kitani, 2014, Ng et al., 2020], generate the motion of only one skeleton.

(3) Three variants of the model are analyzed for generation and classification accuracy, and efficiency on benchmark datasets, in both FP and TP environments. One of the variants yields higher classification accuracy than the others. In each environment, the accuracies are correlated with the number of trainable parameters.

(4) Three ways of implementing action selection (where to attend to) are analyzed for each of the above three variations. Classification accuracy is comparable when sampling locations are determined from prediction error (without any weighting) or from learned weights (without involving prediction error); however, the latter is less efficient in terms of model size.

(5) The proposed agent model is the first of its kind to interact with and learn end-to-end from two-person interaction environments in FP and TP for classification by generation, with high degree of accuracy and efficiency.

The rest of the paper is organized as follows. The proposed agent model is described in Section 7.2 and evaluated on benchmark datasets in Section 7.3. The paper ends with concluding remarks in Section 7.4.

7.2 Models and Methods

7.2.1 Preliminaries

Agent. An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators [Russell and Norvig, 2020].

Perception is the mechanism that allows an agent to interpret sensory signals from the external environment [Han et al., 2016].

Proprioception is perception where the environment is the agent's own body. Proprioception allows an agent to internally perceive the location, movement and action of parts of its body [Han et al., 2016].

Generative model. A generative model, p_{model}, maximizes the log-likelihood $\mathcal{L}(x;\theta)$ of the data, where θ is a set of parameters and x is a set of data points [Goodfellow, 2016].

Evidence lower bound (ELBO). If z is a latent continuous random variable generating the data x, computing log-likelihood requires computing the integral of the marginal likelihood, $\int p_{model}(x,z)dz$, which is intractable [Kingma and Welling, 2013]. Variational inference involves optimization of an approximation of the intractable posterior by defining an evidence lower bound (ELBO) on the log-likelihood,
$\mathcal{L}(x;\theta) \leq \log p_{model}(x;\theta)$.

Variational autoencoder (VAE) is a deep generative model that assumes the data consists of independent and identically distributed samples, and the prior, $p_\theta(z)$, is an isotropic Gaussian. VAE maximizes the ELBO given by [Kingma and Welling, 2013], $\mathcal{L}(x;\theta) \leq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\text{KL}}(q_\phi(z|x), p_\theta(z))$, where $q_\phi(z|x)$ is a recognition model, $p_\theta(x|z)$ is a generative model, $\mathbb{E}$ denotes expectation, and D_{KL} denotes Kullback-Leibler divergence.

Saliency is a property of each location in a predictive agent's environment. The attention mechanism is a function of the agent's prediction error [Spratling, 2012, Friston et al., 2009].

7.2.2 Problem Statement

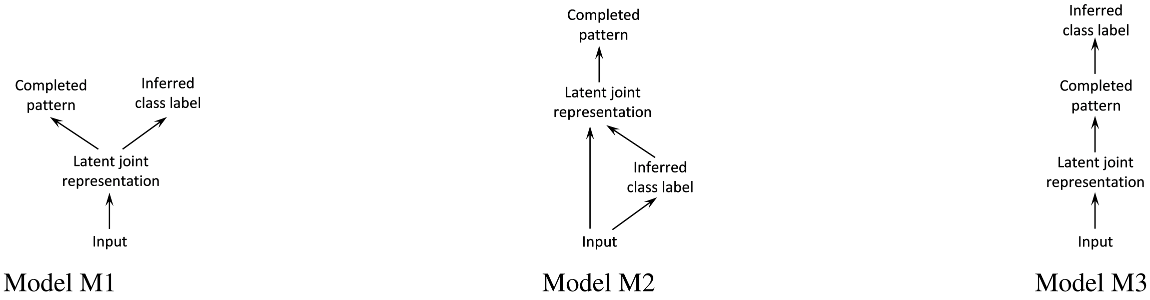
Let $\mathbf{X} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(n)}\}$ be a set of observable variables representing an environment in n modalities. The variable representing the i-th modality is a sequence: $\mathbf{X}^{(i)} = \langle X_1^{(i)}, X_2^{(i)}, \ldots, X_T^{(i)} \rangle$, where T is the sequence length. Let $\mathbf{x}_{\leq t} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\}$ be a partial observation of $\mathbf{X}$ such that $\mathbf{x}^{(i)} = \langle x_1^{(i)}, \ldots, x_t^{(i)} \rangle$, $1 \leq t \leq T$. Let $\mathbf{y}$ be a variable representing the class labels. We define the problem of *pattern completion and classification* as generating $\mathbf{X}$ and $\mathbf{y}$ as accurately as possible from the partial observation $\mathbf{x}_{\leq t}$. Given $\mathbf{x}_{\leq t}$ and a generative model p_θ with parameters θ, at any time t, the objective is to maximize the joint likelihood of $\mathbf{X}$ and $\mathbf{y}$, i.e., $\arg\max_\theta p_\theta(\mathbf{X}, \mathbf{y}|\mathbf{x}_{\leq t})$.

7.2.3 Models

We present three models for solving this problem.

Model M1. The completed pattern and class label are generated from the latent variable $z_{\leq t}$. Mathematically,

$$\begin{aligned} &\arg\max_\theta \int log(p_\theta(\mathbf{X}|\mathbf{x}_{<t}, z_{\leq t}) p_\theta(z_{\leq t}))dz \\ + &\arg\max_\theta \int log(p_\theta(\mathbf{y}|\mathbf{x}_{<t}, z_{\leq t}) p_\theta(z_{\leq t}))dz \end{aligned}$$



Model M1 Model M2 Model M3

Figure 30.: Inference pipeline of three models considered in this paper. Input, completed pattern, and inferred class label are the same in all models.

The model is trained end-to-end. See Fig. 30.a.

Model M2. The class label is inferred directly from partial observations, and then passed as an input to the generative model which generates the completed pattern. This is similar to the model in [Kingma et al., 2014]. Mathematically,

$$\arg\max_{\theta} \int log(p_\theta(\mathbf{X}|\mathbf{x}_{<t}, z_{\leq t})p_\theta(z_{\leq t}))dz \\ + \arg\max_{\phi} \log q_\phi(\mathbf{y}|\mathbf{x}_{<t})$$

where q_ϕ is a recognition model. The model is trained end-to-end. See Fig. 30.b.

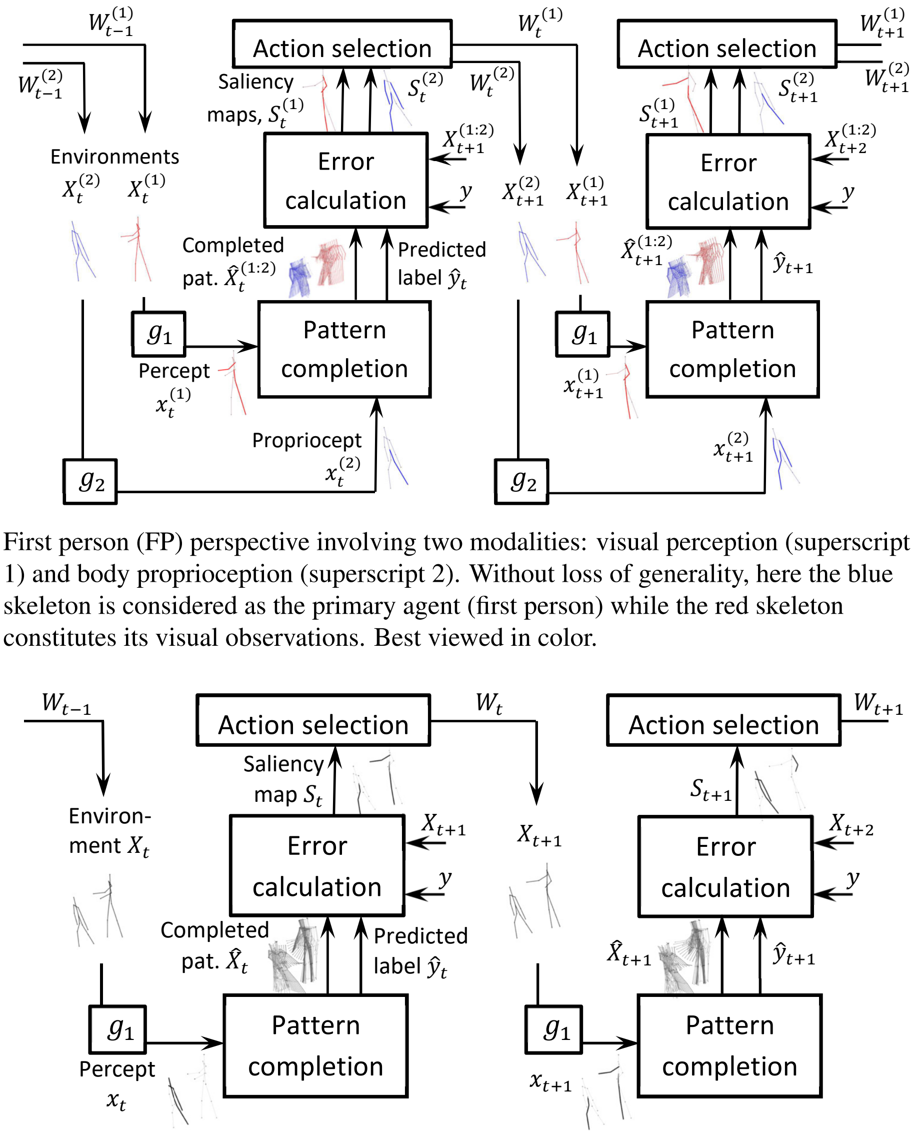
Model M3. The completed pattern is generated from the latent variable $z_{\leq t}$. The class label is inferred from the completed pattern. The pattern completion model is pretrained:

$$\arg\max_{\theta} \int log(p_\theta(\mathbf{X}|\mathbf{x}_{<t}, z_{\leq t})p_\theta(z_{\leq t}))dz$$

Then the classification model is trained:

$$\arg\max_{\pi} log(p_\pi(\mathbf{y}|\mathbf{X}_{<t})$$

Therefore, the model is not end-to-end. See Fig. 30.c.



First person (FP) perspective involving two modalities: visual perception (superscript 1) and body proprioception (superscript 2). Without loss of generality, here the blue skeleton is considered as the primary agent (first person) while the red skeleton constitutes its visual observations. Best viewed in color.

Third person (TP) perspective involving only one modality: visual perception. Hence, superscript indicating the modality is not shown.

Figure 31.: Block diagrams of the proposed attention-based agent applied to two-person interaction generation and classification. In the benchmark skeleton datasets, there is no information regarding the appearance of joints (shape, color, texture) but only their location. The appearance constitutes visual perception ('what') while location constitutes visual proprioception ('where'). There is only one visual modality in this model which we refer to as visual perception (could also be called visual proprioception), in both FP and TP cases.

7.2.4 Agent Architecture

The proposed predictive agent architecture comprises of five components: environment, observation, pattern completion and classification, action selection, and learning. See block diagrams in Fig. 31..

1. Environment. The environment is the source of sensory data. It is time-varying.

2. Observation. The agent interacts with the environment via a sequence of eye and body movements. The observations, sampled from the environment at each time instant, are in two modalities: perceptual and proprioceptive.

3. Pattern completion. A multimodal variational recurrent neural network (MVRNN) for variable length sequences is used for completing the pattern for each modality. Recognition and generation are the two processes involved in the operation of a MVRNN.

Recognition (Encoder). The recognition models, $q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t})$ for models M1 and M3, and $q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t)$ for M2, are probabilistic encoders [Kingma and Welling, 2013]. They produce a Gaussian distribution over the possible values of the code z_t from which the given observations could have been generated.

Model M1. The MVRNN consists of two recurrent neural networks (RNNs), each with one layer of long-short term memory (LSTM) units. Each RNN generates the parameters for the approximate posterior distribution and the conditional prior distribution for each modality, as in [Chung et al., 2015].

Model M2. In addition to the perceptual and proprioceptive modalities, the class label is presented as an input modality. A fully-connected layer from the class labels generates the parameters for the approximate posterior density for the class modality. The recognition model generates the class label.

Model M3. Same as M1.

The distribution parameters from all modalities are combined using product of experts (PoE), as in [Wu and Goodman, 2018], to generate the joint distribution parameters for both the conditional prior, $p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})$ for M1 and M3, or $p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)$ for M2, and the approximate posterior $q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t})$.

The recognition model, similar to that in [Chung et al., 2015], is mathematically expressed in Lines 3–9 of Algorithm 9 and Lines 5–11 of Algorithm 10. Here, ϕ^{prior} generates the mean as a linear function of its input, ϕ^{enc} generates the logarithm of standard deviation as a non-linear function of its input, ϕ^{prior} accepts the hidden state as input, and ϕ^{enc} accepts the hidden state and the current observation as input.

Generation (Decoder). **Model M1.** The generative model, $p_\theta(X_t^{(1)}, X_t^{(2)}, y_t | \mathbf{x}_{<t}, z_{\leq t})$, generates the perceptual and proprioceptive data and the class label from the latent variables, z_t, at each time step.

Model M2. The generative model, $p_\theta(X_t^{(1)}, X_t^{(2)} | \mathbf{x}_{<t}, z_{\leq t})$, generates the perceptual and proprioceptive data from the latent variables, z_t, at each time step.

Model M3. Same as M2.

Each RNN in the MVRNN generates the distribution parameters of the sensory data for a modality. The sensory data is sampled from this distribution. We assume the perceptual and proprioceptive distributions to be multivariate Gaussian as the skeletal joints are real-valued. We assume the class label distribution to be multivariate Bernoulli.

The pattern, $\mathbf{X}$, is completed at each time using an iterative method. At any time t, the model predicts $\hat{\mathbf{x}}_{t+1}$ given the observations $\mathbf{x}_{k:t}$ $(1 \leq k < t)$, then predicts $\hat{\mathbf{x}}_{t+2}$ given $\{\mathbf{x}_{k+1:t}, \hat{\mathbf{x}}_{t+1}\}$, then predicts $\hat{\mathbf{x}}_{t+3}$ given $\{\mathbf{x}_{k+2:t}, \hat{\mathbf{x}}_{t+1:t+2}\}$, and so on till $\hat{\mathbf{x}}_T$ is predicted. This method allows a fixed and finite model to predict a variable or infinite length sequence. Since only the next instant is predicted at any iteration, the model can be size efficient.

The generative model, similar to that in [Chung et al., 2015], is mathematically expressed in Lines 16–20 of Algorithm 9 and Lines 18–22 of Algorithm 10. Here, RNN_θ represents an LSTM unit, and ϕ^{dec} is the same function as ϕ^{enc}.

4. Action selection. In the proposed model, action selection is to decide the weight (attention) given to each location in the environment in order to sample the current observation. At any time t, a saliency map $S_t^{(i)}$ is computed for modality i from which the action is determined. The saliency map assigns a salience score $S_{t,l}^{(i)}$ to each location l. There are 15 locations corresponding to the 15 skeleton joints: Head (J1), Neck (J2), Torso (J3), Left Shoulder (J4), Left Elbow (J5), Left Hand (J6), Right Shoulder (J7), Right

Algorithm 8 Learning the proposed network

```
Initialize parameters of the generative model $\theta$, recognition model $\phi$, sequence length $T$.
Initialize optimizer parameters: $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\eta = 0.001$, $\epsilon = 10^{-10}$.
Initialize $W_0$ values as 1 and $x_1^{(1:2)} \leftarrow F(X_1^{(1:2)}, W_0^{(1:2)})$, where $W_0^{(1:2)}$ are the weights for the initial sampling (ref. experimental setup in Section 7.3.2) and the function $F$ generates a sample $x^{(i)}$ from the environment $X^{(i)}$ after assigning weights $W_0^{(i)}$ to modality $i$ (ref. Action selection in Section 7.2.4).
while true do
    for $\tau \leftarrow 1$ $to$ $T$ do
        Model M1:
        $\hat{X}_{1:T}^{(1:2)}, \hat{y}_{1:T} \leftarrow PatComClassModel1(x_{1:\tau}^{(1:2)})$
        Model M2:
        $\hat{X}_{1:T}^{(1:2)}, \hat{y}_{1:T} \leftarrow PatComClassModel2(x_{1:\tau}^{(1:2)})$
        Model M3:
        $\hat{X}_{1:T}^{(1:2)} \leftarrow PatComClassModel1(x_{1:\tau}^{(1:2)})$
        $\hat{y}_{1:T} \leftarrow Classifier(\hat{X}_{1:T}^{1:2})$
        Saliency Computation (Section 7.2.4 Action selection)
        $S_\tau^{(1:2)} \leftarrow g_1(X_{\tau+1}^{(1:2)}, \hat{X}_{\tau+1}^{(1:2)})$
        $W_\tau^{(1:2)} \leftarrow g_2(S_\tau^{(1:2)})$
        $x_{\tau+1}^{(1:2)} \leftarrow F(X_{\tau+1}^{(1:2)}, W_\tau)$
        Learning
        Update $\{\theta, \phi\}$ by maximizing Eq. 7.4 7.5 or 7.6.
    end for
end while
```

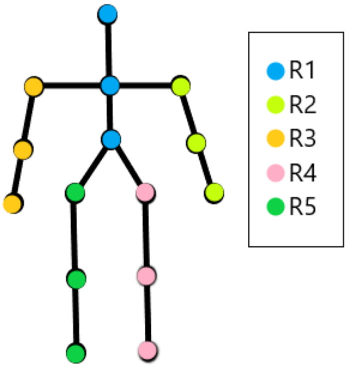


Figure 32.: Different regions in the skeleton.

Elbow (J8), Right Hand (J9), Left Hip (J10), Left knee (J11), Left foot (J12), Right Hip (J13), Right knee (J14), Right foot (J15). We compute the weights in three ways as follows.

Weights are determined by thresholding the prediction error (*pe*). The

Algorithm 9 $PatComClassModel1(x_{1:\tau}^{(1:2)})$

```
for $t \leftarrow 1$ to $T$ do
    Recognition Model
    for $i \leftarrow 1$ to $2$ do
        if $t > \tau$ then
            $x_t^{(i)} \leftarrow \hat{X}_t^{(i)}$
        end if
        $[\mu_{0,t}^{(i)} ; \sigma_{0,t}^{(i)}] \leftarrow \varphi^{prior}(h_{t-1}^{(i)})$
        $[\mu_{z,t}^{(i)} ; \sigma_{z,t}^{(i)}] \leftarrow \varphi^{enc}([x_t^{(i)}, h_{t-1}^{(i)}])$
    end for
    Product of Experts
    $z_t \sim \mathcal{N}(\mu_{0,t}, \Sigma_{0,t}), \quad$ where $\quad \Sigma_{0,t} = \Big(\sum_{i=1}^{2} {\Sigma_{0,t}^{(i)}}^{-2}\Big)^{-1}$
        and $\quad \mu_{0,t} = \Big(\sum_{i=1}^{2} \mu_{0,t}^{(i)} {\Sigma_{0,t}^{(i)}}^{-2}\Big)\Sigma_{0,t}$
    $z_t|\mathbf{x}_t \sim \mathcal{N}(\mu_{z,t}, \Sigma_{z,t}), \quad$ where $\quad \Sigma_{z,t} = \Big(\sum_{i=1}^{2} {\Sigma_{z,t}^{(i)}}^{-2}\Big)^{-1}$
        and $\quad \mu_{z,t} = \Big(\sum_{i=1}^{2} \mu_{z,t}^{(i)} {\Sigma_{z,t}^{(i)}}^{-2}\Big)\Sigma_{z,t}$
    Generative Model
    for $i = 1$ to $2$ do
        $h_t^{(i)} \leftarrow RNN_\theta(h_{t-1}^{(i)}, [z_t, x_t^{(i)}])$
        $[\mu_{x^{(i)},t}^{(i)} ; \sigma_{x^{(i)},t}^{(i)}] \leftarrow \varphi^{dec}([h_{t-1}^{(i)}, z_t])$
        $\hat{X}_t^{(i)} \leftarrow \mu_{x^{(i)},t}^{(i)}$
    end for
    Classification Model
    $h_t^{(3)} \leftarrow RNN_\theta(h_{t-1}^{(3)}, [z_t, \mathbf{x}_t, h_t^{(1)}, h_t^{(2)}])$
    $\hat{y}_t^{(i)} \leftarrow softmax([h_{t-1}^{(3)}, z_t])$
end for
```

threshold is statistically estimated on the fly and is not predetermined.

$$
\begin{gathered}
S_t^{(i)} = \|X_{t+1}^{(i)} - \hat{X}_{t+1}^{(i)}\|_1 \\
S_{t,r}^{(i)} = \tfrac{1}{|r|} \textstyle\sum_{l \in r} S_{t,l}^{(i)} \\
W_{t,l}^{(i)} = \begin{cases} 1, & \text{if } S_{t,l}^{(i)} \geq \frac{1}{n_r} \sum_{i=1}^{n_r} S_{t,r}^{(i)} \\ 0, & \text{otherwise} \end{cases} \\
x_{t+1}^{(i)} = W_t^{(i)} X_{t+1}^{(i)} + (1 - W_t^{(i)}) \hat{X}_{t+1}^{(i)}
\end{gathered}
\tag{7.1}
$$

where $X_{t+1}^{(i)}, \hat{X}_{t+1}^{(i)}$ are the true and predicted data (skeleton joint coordinates) respectively, $||.||_1$ denotes L_1 norm, $|.|$ denotes the cardinality of a set, $n_r = 5$ is the number of regions in the skeleton (J1-J3, J4-J6, J7-J9, J10-J12, J13-J15) (see Fig. 32.), and $S_{t,r}^{(i)}$ is the mean saliency over the joints in region r.

At any time, at least one region will be salient. Our experiments show that variable number of salient regions at each time step is more effective. Fixing the number of salient regions to a constant value occasionally leads to selection of regions with low saliency or overlooking regions with high saliency. In the proposed model, only the salient joints are sampled. For the non-salient joints, the observation at time $t+1$ is the predicted observation from t.

Weights are learned as coefficients of the prediction error (*lwpe*).

$$\begin{aligned} S_t^{(i)} &= W_a(X_{t+1}^{(i)} - \hat{X}_{t+1}^{(i)}) \\ W_t^{(i)} &= \sigma(S_t^{(i)}) \\ x_{t+1}^{(i)} &= W_t^{(i)} X_{t+1}^{(i)} \end{aligned} \tag{7.2}$$

where W_a is the weight matrix.

Weights are learned as coefficients of the hidden states (*lw*).

$$\begin{aligned} S_t^{(i)} &= W_a h_t^{(i)} \\ W_t^{(i)} &= \sigma(S_t^{(i)}) \\ x_{t+1}^{(i)} &= W_t^{(i)} X_{t+1}^{(i)} \end{aligned} \tag{7.3}$$

where W_a is the weight matrix.

6. Learning. The objective is to maximize Eq. 7.4, 7.5, 7.6 for models M1, M2, M3 respectively. The derivation of these equations from the objectives of multimodal VAE [Wu and Goodman, 2018], variational RNN [Chung et al., 2015], and VAE for classification [Kingma et al., 2014].

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i \log p_\theta(X_t^{(i)}|z_{\leq t},\mathbf{x}_{<t})$$
$$+\lambda_3 \log p_\theta(y|z_{\leq T},\mathbf{x}_{<T})\Big]$$
$$-\beta\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t},z_{<t}),p_\theta(z_t|\mathbf{x}_{<t},z_{<t})\big) \tag{7.4}$$

where λ_1, λ_2, λ_3, β are the weights balancing the terms.

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T},y_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i \log p_\theta(X_t^{(i)}|z_{\leq t},\mathbf{x}_{<t})\Big]+\log p_\theta(y)$$
$$-\beta\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t},z_{<t},y_t),p_\theta(z_t|\mathbf{x}_{<t},z_{<t},y_t)\big)$$
$$+\alpha\log q_\phi(y|\mathbf{x}_{\leq t}) \tag{7.5}$$

where α controls the relative weight between generative and purely discriminative learning.

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i \log p_\theta(X_t^{(i)}|z_{\leq t},\mathbf{x}_{<t})\Big]$$
$$-\beta\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}),p_\theta(z_t)\big)+\log q_\pi(y|\mathbf{X}_{1:T}) \tag{7.6}$$

where $q_\pi(y|\mathbf{X}_{1:T})$ is the classification model.

Loss function derivation: Model M1

Here we derive the objective function in Eq. 7.4. The generative and recognition models are factorized as:

$$p_\theta(\mathbf{X}_{\le T}, y_{\le T}, z_{\le T}|\mathbf{x}_{\le T}) = \prod_{t=1}^{T} p_\theta(\mathbf{X}_t, y_t|z_{\le t}, \mathbf{x}_{<t}) p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})$$

$$q_\phi(z_{\le T}|\mathbf{x}_{\le T}) = \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})$$

The variational lower bound (ELBO) on the joint log-likelihood of the generated data, $\log p_\theta(\mathbf{X}_{\le T}, y_{\le T}|\mathbf{x}_{\le T})$, is derived as:

$$\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\log p_\theta(\mathbf{X}_{\le T}, y_{\le T}|\mathbf{x}_{\le T}) \frac{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\log \frac{p_\theta(\mathbf{X}_{\le T}, y_{\le T}, z_{\le T}|\mathbf{x}_{\le T})}{p_\theta(z_{\le T}|\mathbf{x}_{\le T})} \frac{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{X}_t, y_t|z_{\le t}, \mathbf{x}_{<t}) p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})} \frac{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T} \Big[\log p_\theta(\mathbf{X}_t, y_t|z_{\le t}, \mathbf{x}_{<t}) - \log \frac{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})} + \log \frac{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})}\Big]\Big] \\
&\ge \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T} \log p_\theta(\mathbf{X}_t, y_t|z_{\le t}, \mathbf{x}_{<t})\Big] - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})\big)
\end{aligned}$$

We assume, the modalities are conditionally independent given the common latent variables [Wu and Goodman, 2018] and all observations till the current time. Therefore,

$$\log p_\theta(\mathbf{X}_t, y_t|z_{\le t}, \mathbf{x}_{\le t}) = \sum_{i=1}^{2} \log p_\theta(X_t^{(i)}|z_{\le t}, \mathbf{x}_{<t}) + \log p_\theta(y_t|z_{\le t}, \mathbf{x}_{<t})$$

Thus,

$$\begin{aligned}
&\log p_\theta(\mathbf{X}_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T}) \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t}) + \log p_\theta(y_t|z_{\leq t}, \mathbf{x}_{<t})\Big] \\
&-\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})\big) \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i\log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t}) + \lambda_3\log p_\theta(y_t|z_{\leq t}, \mathbf{x}_{<t})\Big] \\
&-\beta\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})\big)
\end{aligned}$$

where λ_1, λ_2, λ_3, β are the weights balancing the terms. Assuming the class label does not change over time, we simplify the above expression as:

$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i\log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t}) + \lambda_3\log p_\theta(y|z_{\leq T}, \mathbf{x}_{<T})\Big] - \beta\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})\big)$

Loss function derivation: Model M2

Here we derive the objective function in Eq. 7.5. The generative and recognition models are factorized as:

$$\begin{aligned}
p_\theta(\mathbf{X}_{\leq T}, y_{\leq T}, z_{\leq T}|\mathbf{x}_{\leq T}) &= \prod_{t=1}^{T} p_\theta(\mathbf{X}_t, y_t|z_{\leq t}, \mathbf{x}_{<t})p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}) \\
q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T}) &= \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t)
\end{aligned}$$

The variational lower bound (ELBO) on the joint log-likelihood of the generated data, $\log p_\theta(\mathbf{X}_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T})$, when the true label is given is derived as:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\log p_\theta(\mathbf{X}_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T})\frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\log \frac{p_\theta(\mathbf{X}_{\leq T}, z_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T})}{p_\theta(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\frac{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}\log \frac{p_\theta(\mathbf{X}_t|z_{\leq t}, \mathbf{x}_{<t})p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)p_\theta(y_t)}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)}\frac{q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t)}{q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t)}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}\Big[\log p_\theta(\mathbf{X}_t|z_{\leq t}, \mathbf{x}_{<t}) + \log p_\theta(y_t) - \log\frac{q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t)}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)} \\
&+ \log\frac{q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t)}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)}\Big]\Big] \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}\log p_\theta(\mathbf{X}_t|z_{\leq t}, \mathbf{x}_{<t}) + \log p_\theta(y_t)\Big] \\
&- \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)\big) \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t}) + \log p_\theta(y_t)\Big] \\
&- \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)\big)
\end{aligned}
$$

After adding the classification loss, the final objective function can be written as:

$$
\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T}, y_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i \log p_\theta(X_t^{(i)}|z_{\leq t}, \mathbf{x}_{<t})\Big] + \log p_\theta(y_t) \\
&\quad - \beta\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}, z_{<t}, y_t), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)\big) + \alpha\sum_{t=1}^{T}\log q_\phi(y_t|\mathbf{x}_{\leq t})
\end{aligned}
$$

where α controls the relative weight between generative and purely discriminative learning. Assuming the class label does not change over time, we simplify the above expression as:

$$\mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T}, y_{\le T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i \log p_\theta(X_t^{(i)}|z_{\le t}, \mathbf{x}_{<t})\Big] + \log p_\theta(y)$$

$$- \beta \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t}, y_t), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t}, y_t)\big) + \alpha \log q_\phi(y|\mathbf{x}_{\le T})$$

Loss function derivation: Model M3

Here we derive the objective function in Eq. 7.6. The generative and recognition models are factorized as:

$$p_\theta(\mathbf{X}_{\le T}, z_{\le T}|\mathbf{x}_{\le T}) = \prod_{t=1}^{T} p_\theta(\mathbf{X}_t|z_{\le t}, \mathbf{x}_{<t}) p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})$$

$$q_\phi(z_{\le T}|\mathbf{x}_{\le T}) = \prod_{t=1}^{T} q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})$$

The variational lower bound (ELBO) on the log-likelihood of the generated data, $\log p_\theta(\mathbf{X}_{\le T}|\mathbf{x}_{\le T})$, is derived as:

$$\begin{aligned}
&\mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\log p_\theta(\mathbf{X}_{\le T}|\mathbf{x}_{\le T}) \frac{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\log \frac{p_\theta(\mathbf{X}_{\le T}, z_{\le T}|\mathbf{x}_{\le T})}{p_\theta(z_{\le T}|\mathbf{x}_{\le T})} \frac{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{X}_t|z_{\le t}, \mathbf{x}_{<t}) p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})} \frac{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}\Big] \\
&= \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T}\Big(\log p_\theta(\mathbf{X}_t|z_{\le t}, \mathbf{x}_{<t}) - \log \frac{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})} + \log \frac{q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t})}{p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})}\Big)\Big] \\
&\ge \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T} \log p_\theta(\mathbf{X}_t|z_{\le t}, \mathbf{x}_{<t})\Big] - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})\big) \\
&\ge \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2} \log p_\theta(X_t^{(i)}|z_{\le t}, \mathbf{x}_{<t})\Big] - \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})\big) \\
&\ge \mathbb{E}_{q_\phi(z_{\le T}|\mathbf{x}_{\le T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2} \lambda_i \log p_\theta(X_t^{(i)}|z_{\le t}, \mathbf{x}_{<t})\Big] - \beta \sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\le t}, z_{<t}), p_\theta(z_t|\mathbf{x}_{<t}, z_{<t})\big)
\end{aligned}$$

where λ_1, λ_2, β are the weights balancing the terms.

After adding the classification loss, the final objective function can be written as:

$$\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T}\sum_{i=1}^{2}\lambda_i \log p_\theta(X_t^{(i)}|z_{\leq t},\mathbf{x}_{<t})\Big] - \beta\sum_{t=1}^{T} D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big) + \log q_\pi(y|\mathbf{X}_{1:T})$$

where $q_\pi(y|\mathbf{X}_{1:T})$ is the classification model.

Table 18.: Performance (AFD) averaged over all examples for each interaction in the test set and all train-test splits (mean, std. dev.) for **first person** modeling for **SBU Kinect Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection). Interactions Approach, Shake Hands, Exchange Object are abbreviated as Appr, Sh Hands, Exc Obj, respectively. Average is abbreviated as Avg.

Model	Appr	Depart	Kick	Push	Sh Hands	Hug	Exc Obj	Punch	Avg AFD
M1 (*bs*)	.031, .02	.034, .02	.072, .04	.044, .02	.032, .01	.060, .02	.037, .05	.053, .02	.045, .01
M2 (*bs*)	.026, .01	.028, .02	.064, .03	.043, .02	.031, .02	.055, .02	.032, .01	.046, .02	.041, .01
M3 (*bs*)	.020, .01	.023, .02	.050, .03	.030, .01	.021, .01	.042, .02	.024, .01	.036, .02	**.031, .01**
M1 (*pe*)	.102, .07	.125, .10	.244, .27	.129, .10	.112, .06	.171, .11	.132, .10	.170, .11	.148, .04
M2 (*pe*)	.092, .06	.100, .07	.228, .20	.131, .08	.113, .06	.170, .07	.126, .11	.159, .11	.140, .04
M3 (*pe*)	.065, .05	.085, .06	.189, .28	.093, .10	.076, .03	.129, .07	.092, .10	.126, .12	**.107, .04**
M1 (*lwpe*)	.028, .02	.033, .02	.071, .04	.043, .02	.032, .03	.059, .03	.035, .01	.052, .02	.044, .01
M2 (*lwpe*)	.029, .02	.033, .02	.077, .04	.046, .02	.033, .03	.062, .02	.036, .01	.056, .02	.047, .02
M3 (*lwpe*)	.026, .02	.030, .02	.067, .04	.040, .02	.027, .02	.052, .02	.033, .02	.047, .02	**.040, .01**
M1 (*lw*)	.032, .02	.035, .02	.072, .04	.045, .02	.032, .02	.057, .02	.036, .02	.052, .02	.045, .01
M2 (*lw*)	.061, .05	.066, .07	.146, .10	.102, .05	.076, .06	.125, .07	.082, .05	.113, .07	.096, .03
M3 (*lw*)	.020, .01	.023, .02	.052, .03	.031, .02	.022, .01	.043, .02	.025, .01	.037, .02	**.032, .01**

Table 19.: Performance (AFD) averaged over all examples for each interaction in the test set and all train-test splits (mean, std dev) for **third person** modeling for **SBU Kinect Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection); interactions Approach, Shake hands and Exchange object are abbreviated as Appr., Sh. Hands, Exc. ob. respectively; metric average AFD is abbreviated as Avg. AFD.

Model	Appr.	Depart	Kick	Push	Sh. Hands	Hug	Exc. ob.	Punch	Avg. AFD
M1 (*bs*)	.040, .03	.043, .03	.097, .05	.059, .03	.042, .03	.075, .04	.046, .01	.067, .03	.059, .02
M2 (*bs*)	.056, .04	.058, .04	.134, .08	.083, .04	.056, .05	.100, .05	.063, .02	.092, .05	.080, .03
M3 (*bs*)	.026, .02	.030, .02	.072, .04	.042, .02	.028, .02	.056, .02	.034, .01	.049, .02	**.042, .02**
M1 (*pe*)	.098, .04	.101, .04	.215, .08	.114, .07	.172, .07	.108, .04	.152, .04	.152, .04	.137, .04
M2 (*pe*)	.118, .06	.129, .06	.279, .11	.171, .08	.126, .08	.215, .06	.126, .04	.186, .04	.169, .06
M3 (*pe*)	.068, .04	.079, .04	.184, .07	.107, .04	.082, .04	.141, .06	.082, .03	.120, .03	**.108, .04**
M1 (*lwpe*)	.046, .04	.054, .05	.121, .06	.072, .03	.051, .03	.095, .04	.059, .02	.083, .03	.073, .02
M2 (*lwpe*)	.078, .06	.084, .09	.177, .10	.108, .04	.079, .04	.144, .08	.089, .04	.133, .07	.111, .04
M3 (*lwpe*)	.038, .03	.044, .03	.095, .05	.055, .02	.039, .04	.073, .03	.046, .02	.065, .02	**.057, .02**
M1 (*lw*)	.042, .03	.047, .03	.108, .07	.063, .03	.044, .04	.077, .04	.048, .01	.071, .03	.062, .02
M2 (*lw*)	.076, .09	.119, .22	.191, .18	.124, .10	.092, .08	.155, .14	.101, .10	.139, .11	.125, .04
M3 (*lw*)	.028, .02	.033, .02	.078, .04	.042, .02	.029, .02	.057, .02	.034, .01	.050, .02	**.044, .02**

7.3 Experimental Results

7.3.1 Datasets

Our model is evaluated on two datasets:

Table 20.: Performance (AFD) averaged over all examples for each interaction in the test set and all train-test splits (mean, std dev) for **first person** modeling for **K3HI Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection); interactions Approach, Shake hands and Exchange object are abbreviated as Appr., Sh. Hands, Exc. ob. respectively; metric average AFD is abbreviated as Avg. AFD.

Model	Appr.	Depart	Exc. ob.	Kick	Point	Punch	Push	Sh. Hands	Avg. AFD
M1 (*bs*)	.153, .99	.015, .01	.006, .01	.011, .01	.007, .00	.010, .01	.010, .00	.006, .00	**.027, .05**
M2 (*bs*)	.146, 1.0	.016, .01	.006, .01	.012, .01	.008, .00	.010, .01	.010, .01	.006, .00	**.027, .05**
M3 (*bs*)	.143, .85	.022, .02	.013, .01	.022, .03	.016, .02	.020, .03	.019, .02	.012, .02	.033, .04
M1 (*pe*)	.135, .74	.037, .03	.020, .01	.033, .02	.025, .02	.026, .02	.027, .01	.019, .02	**.040, .04**
M2 (*pe*)	.136, .66	.048, .03	.029, .02	.052, .03	.038, .03	.039, .02	.041, .02	.031, .02	.052, .03
M3 (*pe*)	.126, .61	.041, .03	.021, .01	.038, .03	.028, .03	.029, .03	.031, .02	.021, .02	.042, .03
M1 (*lwpe*)	.143, .87	.017, .02	.007, .01	.013, .01	.010, .02	.011, .01	.011, .01	.007, .01	**.027, .05**
M2 (*lwpe*)	.148, .91	.020, .02	.009, .01	.016, .01	.012, .01	.013, .01	.013, .01	.009, .01	.030, .05
M3 (*lwpe*)	.135, .75	.029, .03	.017, .02	.031, .05	.021, .03	.026, .04	.027, .04	.017, .03	.038, .04
M1 (*lw*)	.164, 1.1	.016, .01	.006, .01	.012, .01	.007, .00	.009, .01	.009, .01	.006, .00	**.029, .05**
M2 (*lw*)	.154, .97	.018, .02	.007, .01	.014, .01	.008, .01	.011, .01	.011, .01	.006, .01	**.029, .05**
M3 (*lw*)	.141, .85	.027, .02	.017, .02	.030, .05	.021, .03	.026, .04	.025, .04	.017, .03	.038, .04

Table 21.: Performance (AFD) averaged over all examples for each interaction in the test set and all train-test splits (mean, std dev) for **third person** modeling for **K3HI Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection); interactions Approach, Shake hands and Exchange object are abbreviated as Appr., Sh. Hands, Exc. ob. respectively; metric average AFD is abbreviated as Avg. AFD.

Model	Appr.	Depart	Exc. ob.	Kick	Point	Punch	Push	Sh. Hands	Avg. AFD
M1 (*bs*)	.155, .96	.024, .01	.013, .01	.025, .02	.018, .02	.019, .02	.020, .01	.014, .01	.036, .05
M2 (*bs*)	.155, .89	.026, .01	.016, .01	.027, .02	.023, .03	.022, .02	.023, .01	.019, .02	.039, .05
M3 (*bs*)	.154, .96	.017, .01	.007, .01	.015, .01	.010, .01	.011, .01	.012, .01	.007, .01	**.029, .05**
M1 (*pe*)	.161, .75	.044, .02	.027, .02	.054, .03	.047, .04	.040, .02	.042, .02	.031, .02	.056, .04
M2 (*pe*)	.169, .66	.047, .02	.031, .02	.062, .03	.055, .05	.046, .02	.048, .02	.035, .02	.062, .04
M3 (*pe*)	.154, .71	.038, .02	.024, .02	.048, .03	.038, .03	.037, .03	.039, .02	.026, .02	**.051, .04**
M1 (*lwpe*)	.159, .94	.024, .02	.013, .01	.026, .02	.022, .03	.019, .01	.021, .01	.014, .01	**.037, .05**
M2 (*lwpe*)	.156, .92	.029, .02	.020, .02	.036, .03	.029, .03	.029, .02	.031, .02	.020, .01	.044, .04
M3 (*lwpe*)	.151, 1.0	.033, .02	.021, .02	.041, .05	.039, .05	.033, .03	.033, .03	.023, .02	.047, .04
M1 (*lw*)	.161, 1.0	.021, .02	.010, .01	.020, .01	.015, .02	.014, .01	.015, .01	.009, .01	**.033, .05**
M2 (*lw*)	.154, .92	.024, .02	.012, .01	.024, .02	.019, .02	.018, .01	.019, .01	.012, .01	.035, .05
M3 (*lw*)	.146, .90	.031, .02	.019, .02	.036, .05	.030, .04	.030, .04	.030, .04	.020, .03	.043, .03

Table 22.: Class prediction results using **first person** modeling and **SBU Kinect Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection); classification accuracy is abbreviated as Acc.

Model	Acc.	Recall	Precision	F1 Score
M1 (*bs*)	**93.2, 4.7**	**.934, .04**	**.931, .05**	**.928, .05**
M2 (*bs*)	91.9, 5.6	.927, .04	.913, .06	.912, .05
M3 (*bs*)	82.2, 10.1	.846, .09	.817, .11	.814, .11
M1 (*pe*)	**93.1, 3.75**	**.940, .03**	**.924, .04**	**.925, .03**
M2 (*pe*)	89.3, 5.1	.895, .03	.869, .05	.886, .04
M3 (*pe*)	80.4, 8.5	.837, .08	.799, .09	.796, .09
M1 (*lwpe*)	93.1, 3.9	.939, .04	.929, .04	.929, .04
M2 (*lwpe*)	**93.8, 4.7**	**.945, .04**	**.934, .06**	**.931, .06**
M3 (*lwpe*)	81.4, 9.1	.842, .08	.809, .10	.807, .10
M1 (*lw*)	**91.5, 6.0**	**.920, .05**	**.902, .07**	**.903, .07**
M2 (*lw*)	59.8, 14.7	.655, .13	.564, .14	.627, .13
M3 (*lw*)	83.2, 8.3	.855, .07	.823, .09	.823, .09

Table 23.: Class prediction results using **third person** modeling and **SBU Kinect Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection); classification accuracy is abbreviated as Acc.

Model	Acc.	Recall	Precision	F1 Score
M1 (*bs*)	**93.7, 6.1**	**.944, .05**	**.935, .05**	**.934, .06**
M2 (*bs*)	92.1, 3.9	.923, .03	.920, .04	.914, .04
M3 (*bs*)	82.5, 8.8	.847, .08	.818, .10	.814, .10
M1 (*pe*)	**92.5,5.5**	**.930, .05**	**.927, .05**	**.922, .05**
M2 (*pe*)	90.1, 6.2	.909, .05	.879, .05	.894, .06
M3 (*pe*)	79.3, 7.8	.807 .09	.781, .09	.775, .09
M1 (*lwpe*)	91.3, 7.5	.915, .06	.907, .08	.906, .07
M2 (*lwpe*)	**91.4, 5.5**	**.919, .05**	**.908, .05**	**.905, .06**
M3 (*lwpe*)	81.7, 7.2	.842, .07	.815, .08	.811, .07
M1 (*lw*)	**92.9, 5.8**	**.951, .03**	**.921, .05**	**.924, .05**
M2 (*lw*)	71.3, 6.0	.773, .07	.694, .08	.738, .04
M3 (*lw*)	82.1, 8.5	.074, .08	.815, .09	.813, .09

Table 24.: Class prediction results using **first person** modeling and **K3HI Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection); classification accuracy is abbreviated as Acc.

Model	Acc.	Recall	Precision	F1 Score
M1 (*bs*)	**87.5, 7.1**	**.865, .08**	**.859, .08**	**.856, .08**
M2 (*bs*)	82.7, 3.1	.817, .04	.806, .04	.804, .04
M3 (*bs*)	80.1, 3.1	.796, .03	.783, .02	.777, .03
M1 (*pe*)	**85.9, 5.2**	**.854, .07**	**.838, .06**	**.839, .06**
M2 (*pe*)	84.9, 3.3	.836, .04	.835, .04	.831, .04
M3 (*pe*)	76.9, 2.6	.768, .02	.760, .02	.752, .02
M1 (*lwpe*)	**84.9, 3.5**	**.850, .05**	**.818, .03**	**.818, .03**
M2 (*lwpe*)	82.1, 6.3	.828, .07	.802, .07	.801, .06
M3 (*lwpe*)	75.6, 4.0	.759, .03	.746, .03	.739, .03
M1 (*lw*)	**86.9, 4.3**	**.865, .05**	**.852, .05**	**.853, .05**
M2 (*lw*)	83.7, 3.0	.840, .05	.824, .04	.822, .04
M3 (*lw*)	76.3, 4.7	.760, .04	.753, .04	.745, .04

Table 25.: Class prediction results using **third person** modeling and **K3HI Interaction dataset**. (*bs*), (*pe*), (*lwpe*) and (*lw*) are different action selection methods (ref. Section 4 Action selection); classification accuracy is abbreviated as Acc.

Model	Acc.	Recall	Precision	F1 Score
M1 (*bs*)	**83.0, 6.6**	**.827, .07**	**.816, .08**	**.813, .08**
M2 (*bs*)	81.1, 3.3	.796, .03	.783, .03	.780, .03
M3 (*bs*)	80.1, 3.1	.796, .03	.783, .02	.777, .03
M1 (*pe*)	**82.7, 7.3**	**.816, .08**	**.815, .08**	**.810, .08**
M2 (*pe*)	82.4, 3.9	.825, .04	.804, .04	.805, .05
M3 (*pe*)	75.0, 5.7	.762, .04	.741, .05	.738, .05
M1 (*lwpe*)	**82.1, 4.5**	**.809, .04**	**.800, .06**	**.796, .05**
M2 (*lwpe*)	80.5, 7.8	.794, .08	.790, .10	.784, .09
M3 (*lwpe*)	72.7, 8.3	.731, .07	.720, .07	.712, .07
M1 (*lw*)	**80.8, 6.3**	**.793, .07**	**.775, .08**	**.777, .08**
M2 (*lw*)	78.3, 6.3	.803, .07	.766, .07	.764, .08
M3 (*lw*)	75.0, 7.1	.758, .05	.741, .06	.736, .06

Table 26.: % salient joints (mean, std dev) sampled by different versions of our model from the ground truth using **first person** modeling shown for (*pe*); (*bs*), (*lwpe*), and (*lw*) do not have sparsity. Interactions Shake hands and Exchange object are abbreviated as Sh. Hands, Exc. obj. respectively.

Dataset	Model	Approach	Depart	Kick	Push	Sh. Hands	Exc. obj.	Punch	Hug	Avg.
SBU	M1	48.9, 4.2	48.7, 3.9	46.6, 2.8	49.3, 2.1	49.8, 2.3	48.9, 2.2	49.9, 3.2	48.3, 3.0	48.8, 1.0
	M2	48.3, 3.5	48.4, 4.3	46.7, 3.3	49.2, 2.4	49.8, 2.7	48.4, 2.5	49.3, 2.5	47.4, 2.6	48.4, 1.0
	M3	48.5, 3.7	47.8, 4.4	46.3, 2.6	49.2, 2.2	48.7, 2.4	48.0, 1.9	49.3, 3.4	48.4, 2.3	48.3, 1.0
		Approach	Depart	Exc. obj.	Kick	Point	Punch	Push	Sh. Hands	Avg.
K3HI	M1	47.9, 3.0	47.6, 2.4	47.8, 3.0	45.8, 2.6	46.8, 4.4	47.4, 2.4	47.6, 2.0	46.3, 2.9	47.2, 1.0
	M2	48.4, 2.4	48.4, 2.1	48.3, 3.9	44.5, 2.5	44.8, 4.1	47.3, 2.7	47.9, 3.2	47.7, 4.2	47.1, 1.6
	M3	48.0, 2.2	47.9, 2.2	48.2, 3.2	44.6, 2.6	45.9, 4.4	47.5, 2.7	48.0, 3.3	47.0, 3.9	47.1, 1.3

Table 27.: % salient joints (mean, std dev) sampled by different versions of our model from the ground truth using **third person** modeling shown for (*pe*); (*bs*), (*lwpe*), and (*lw*) do not have sparsity. Interactions Shake hands and Exchange object are abbreviated as Sh. Hands, Exc. obj. respectively.

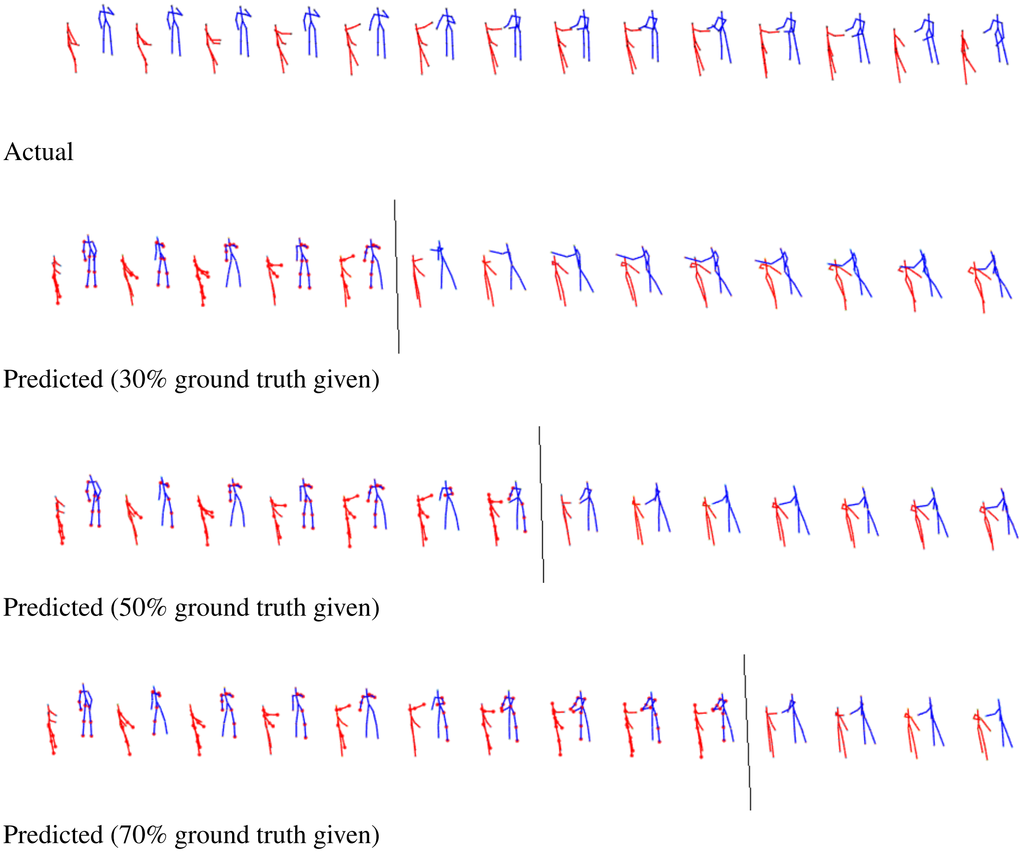
Dataset	Model	Approach	Depart	Kick	Push	Sh. Hands	Exc. obj.	Punch	Hug	Avg.
SBU	M1	47.5, 3.8	45.8, 4.6	45.1, 3.2	48.4, 2.7	47.7, 3.2	47.6, 2.8	48.7, 3.8	47.4, 2.9	47.3, 1.2
	M2	47.8, 4.5	45.6, 4.4	44.4, 3.0	48.6, 3.6	47.7, 3.8	47.5, 3.2	48.0, 3.9	47.1, 3.4	47.1, 1.4
	M3	46.7, 3.4	46.2, 4.4	44.6, 3.0	48.9, 3.1	47.9, 4.0	47.4, 2.5	47.7, 5.3	47.8, 3.4	47.1, 1.3
		Approach	Depart	Exc.	Kick	Point	Punch	Push	Sh. Hands	Avg.
K3HI	M1	47.2, .2.9	47.9, 3.0	46.9, 2.9	41.1, 3.5	39.9, 7.2	45.5, 3.1	45.8, 3.7	46.8, 5.5	45.1, 3.0
	M2	48.0, 3.6	48.6, 2.7	47.1, 2.6	41.0, 3.1	37.7, 6.4	44.6, 3.8	45.5, 3.1	45.9, 4.3	44.8, 3.7
	M3	47.2, 4.3	47.1, 3.0	45.9, 3.4	41.2, 3.7	40.4, 6.9	45.0, 2.5	44.3, 3.4	45.4, 4.4	44.6, 2.5

(1) SBU Kinect Interaction Dataset [Yun et al., 2012] is a two-person interaction dataset comprising of eight interactions: approaching, departing, pushing, kicking, punching, exchanging objects, hugging, and shaking hands. The data is recorded from 7 participants forming a total of 21 sets such that each set consists of a unique pair of participants performing all actions. The dataset has approximately 300 interactions of duration 9 to 46 frames. The dataset is divided into five distinct train test splits as in [Yun et al., 2012].

(2) K3HI: Kinect-based 3D Human Interaction Dataset [Hu et al., 2013] is a two-person interaction dataset comprising of eight interactions: approaching, departing, kicking, punching, pointing, pushing, exchanging an object, and shaking hands. The data is recorded from 15 volunteers. Each pair of participants performs all the actions. The dataset has approximately 320 interactions of duration 20 to 104 frames. The dataset is divided into four distinct train test splits as in [Hu et al., 2013].

7.3.2 Experimental setup

We use a single hidden layer, as in [Chung et al., 2015], for each modality in the MVRNN. Each modality in the MVRNN has a recurrent hidden layer of 256 units and a



Actual

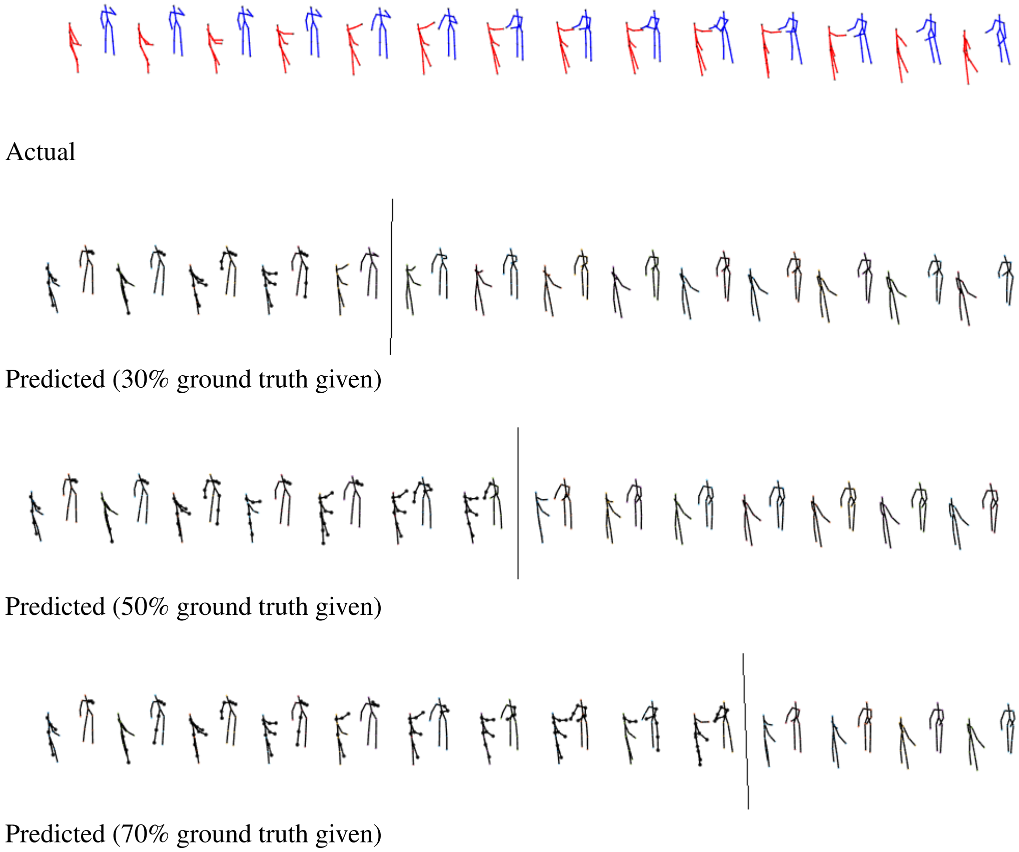
Predicted (30% ground truth given)

Predicted (50% ground truth given)

Predicted (70% ground truth given)

Figure 33.: The top row represents true skeletal data for the prediction at alternate time steps for SBU Kinect Interaction data for exchanging object for first person modeling. Each skeleton in rows 2, 3 and 4 shows one step ahead prediction until 30%, 50% and 70% of the ground truth is given (highlighted by the grey line) respectively. Beyond that, the model uses its own prediction as input for completing the patterns until the final time step is reached. The salient joints are marked red.

latent layer of 20 variables. These parameters are estimated empirically. T is variable as interaction videos are of different lengths. Stochastic gradient descent, with a minibatch size of 100, is used to train the model. Adam optimization with a learning rate of 0.001 and default hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) are used. The objective function parameters, β, λ_1 and λ_2, are fixed to 1 while λ_3 and α are fixed to 50. The models are trained until the error converges. To avoid overfitting, we use a dropout probability of 0.8 for M1, M2 and M3 at the hidden layer for generation and 0.1 for M1 and M2 at the hidden layer for classification. All hyperparameters, except the defaults, are estimated from the training set by cross-validation.


Actual

Predicted (30% ground truth given)

Predicted (50% ground truth given)

Predicted (70% ground truth given)

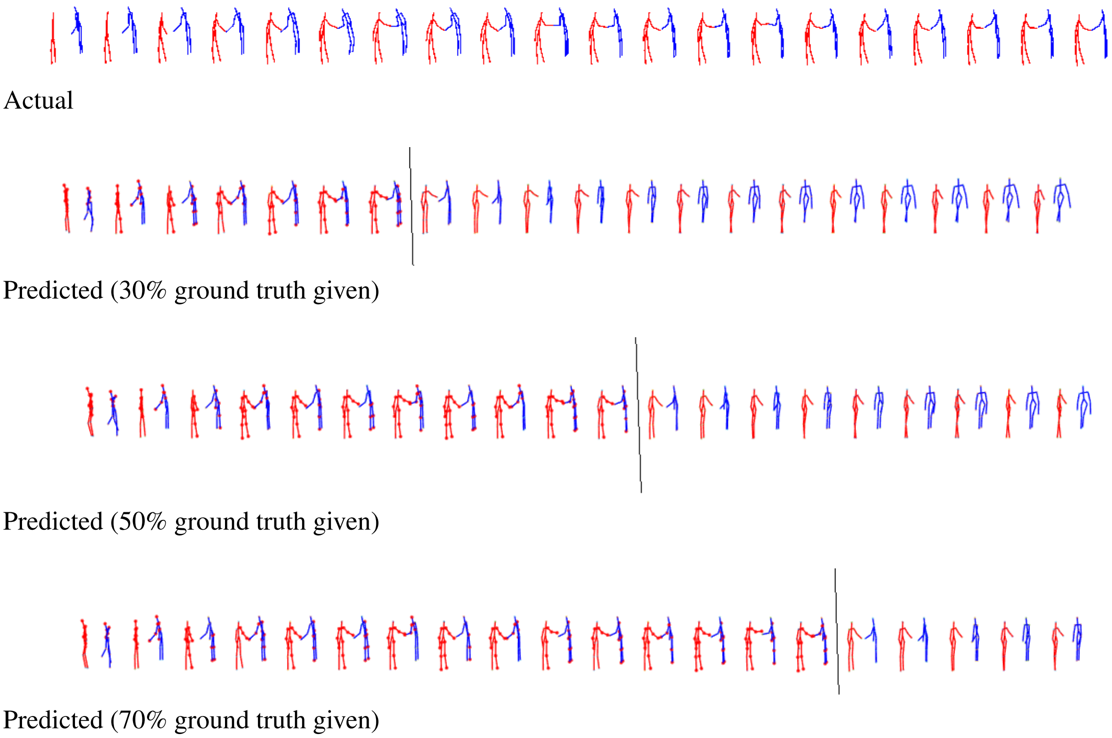
Figure 34.: The top row represents true skeletal data for the prediction at alternate time steps for SBU Kinect Interaction data for exchanging object for third person modeling. Each skeleton in rows 2, 3 and 4 shows one step ahead prediction until 30%, 50% and 70% of the ground truth is given (highlighted by the grey line) respectively. Beyond that, the model uses its own prediction as input for completing the patterns until the final time step is reached. The salient joints are marked red.

7.3.3 Evaluation

In the two benchmark datasets, each skeleton consists of 15 joints. The skeletal data in SBU is normalized. We do not apply any further preprocessing. We standardize the skeletal data in K3HI. Training models on low-level handcrafted features defeats the purpose of learning. Hence, our inclination towards operating on raw skeletal data.

Our experiments are carried out on two settings:

1. First person: Here we model the agent as the first person (one of the two skeletons). His body constitutes his internal environment while the other skeleton constitutes his external (visual) environment. Two modalities are used in our model (see Fig. 31.a): (i) visual perception which captures the other skeleton's 3D joint coordinates,



Actual

Predicted (30% ground truth given)

Predicted (50% ground truth given)

Predicted (70% ground truth given)

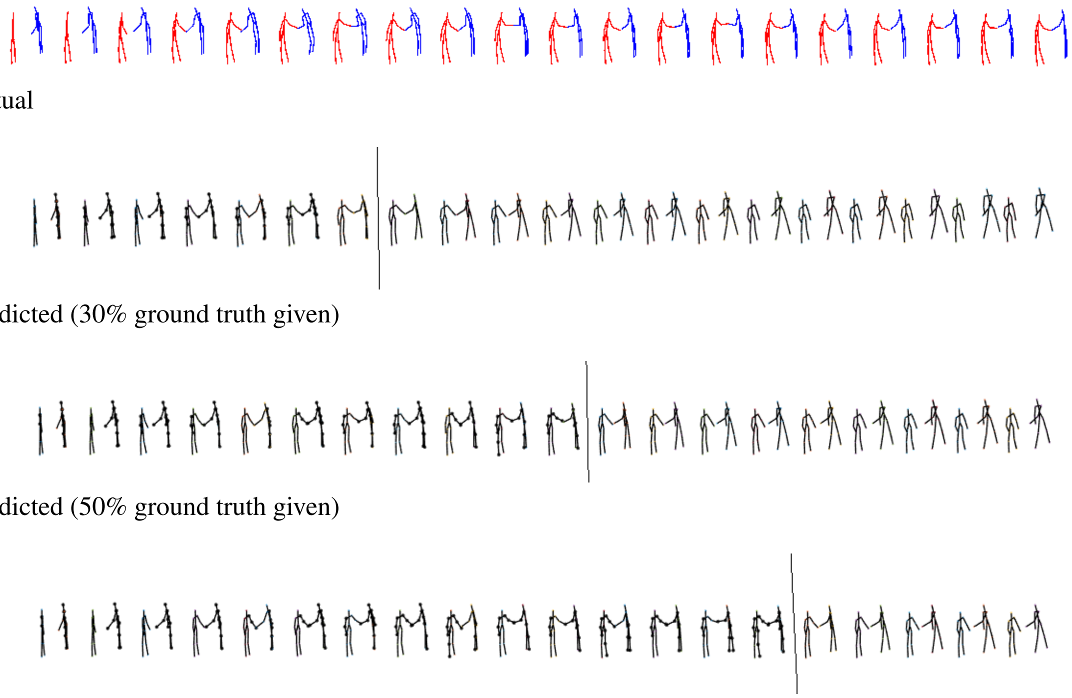
Figure 35.: The top row represents true skeletal data for the prediction at every third instant for K3HI Intersection data for shaking hands for first person modeling. Each skeleton in rows 2, 3 and 4 shows one step ahead prediction until 30%, 50% and 70% of the ground truth is given (highlighted by the grey line) respectively. Beyond that, the model uses its own prediction as input for completing the patterns until the final time step is reached. The salient joints are marked red.

and (ii) body proprioception which captures the first skeleton's 3D joint coordinates. Here, $i = 1, 2$ in the objective function (ref. Eqs. 7.4, 7.5, 7.6).

2. Third person: Here we model the agent as a third person (e.g., audience). The two interaction skeletons constitute his external (visual) environment. One modality is used in our model (see Fig. 31.b): visual perception which captures both the skeletons' 3D joint coordinates. Here, $i = 1$ in the objective function (ref. Eqs. 7.4, 7.5, 7.6).

Model variations: For each of the above two settings, we experiment with the three action selection methods (ref. "Action selection" in Section 7.2.4): *pe*, *lwpe*, and *lw*.

Ablation study - Baseline, *bs* (w/o attention): Due to lack of end-to-end models that simultaneously generate and classify two-person interactions from 3D skeletal data, our model's performance is evaluated using an ablation study which we refer to as the baseline, *bs*. The goal is to understand the utility of attention in our model. For that, we create a baseline model (*bs*) where attention (i.e. action selection, ref. Lines 17–19 in



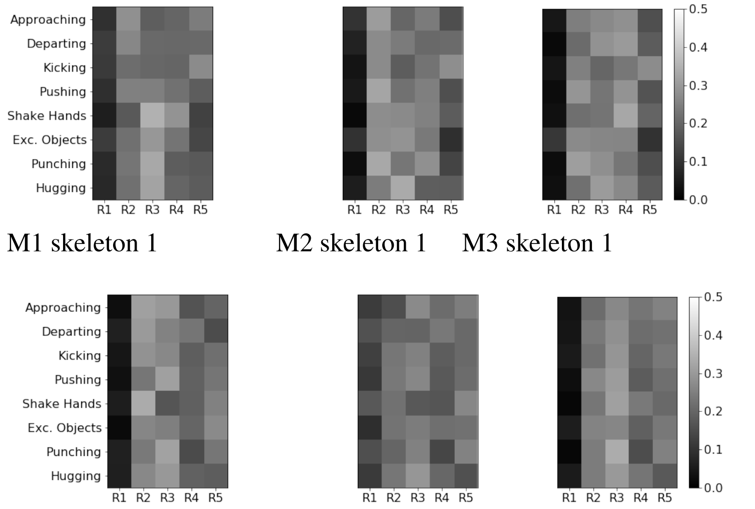
Actual

Predicted (30% ground truth given)

Predicted (50% ground truth given)

Predicted (70% ground truth given)

Figure 36.: The top row represents true skeletal data for the prediction at every third instant for K3HI Intersection data for shaking hands for third person modeling. Each skeleton in rows 2, 3 and 4 shows one step ahead prediction until 30%, 50% and 70% of the ground truth is given (highlighted by the grey line) respectively. Beyond that, the model uses its own prediction as input for completing the patterns until the final time step is reached. The salient joints are marked red.

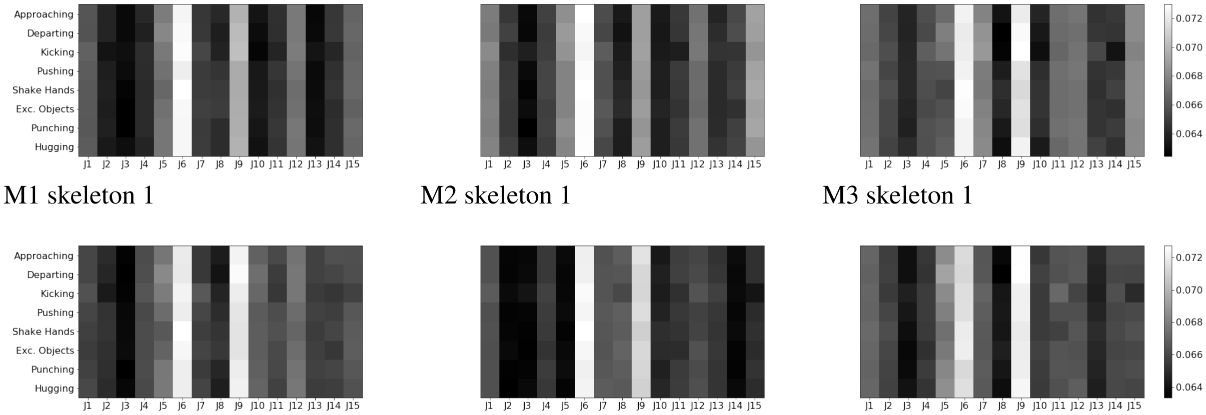


M1 skeleton 1 M2 skeleton 1 M3 skeleton 1

M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

Figure 37.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **first person, (*pe*)** modeling using SBU Kinect interaction data.

Algorithm 8) is eliminated from the proposed model. The MVRNN is modified such that the observation is sampled from all the joints (i.e., weight distribution is uniform over all



M1 skeleton 1

M2 skeleton 1

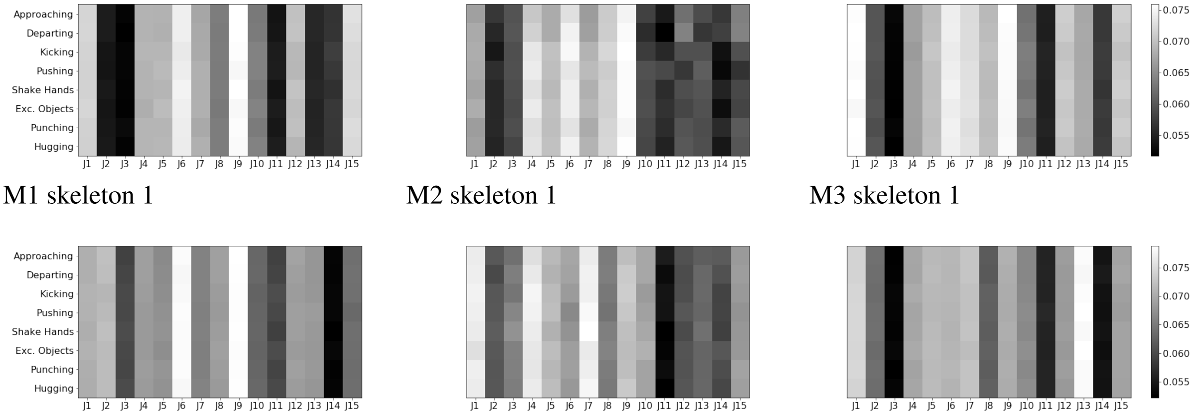
M3 skeleton 1

M1 skeleton 2

M2 skeleton 2

M3 skeleton 2

Figure 38.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **first person, (*lwpe*)** modeling using SBU Kinect interaction data.



M1 skeleton 1

M2 skeleton 1

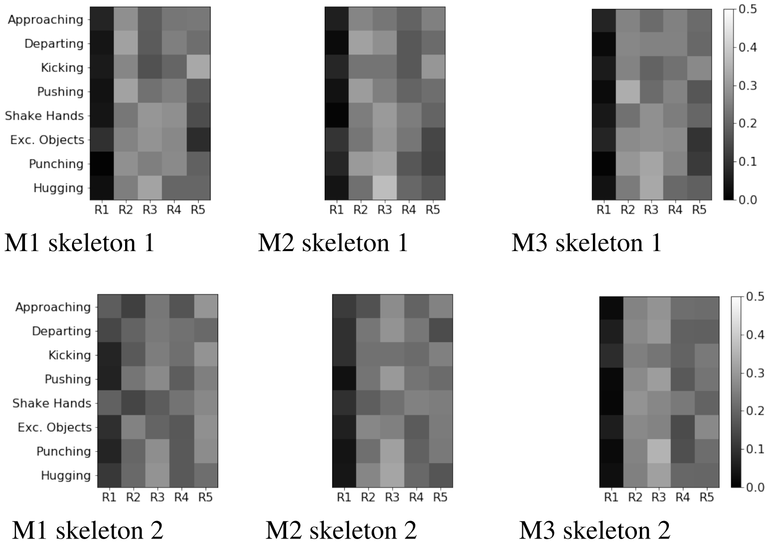
M3 skeleton 1

M1 skeleton 2

M2 skeleton 2

M3 skeleton 2

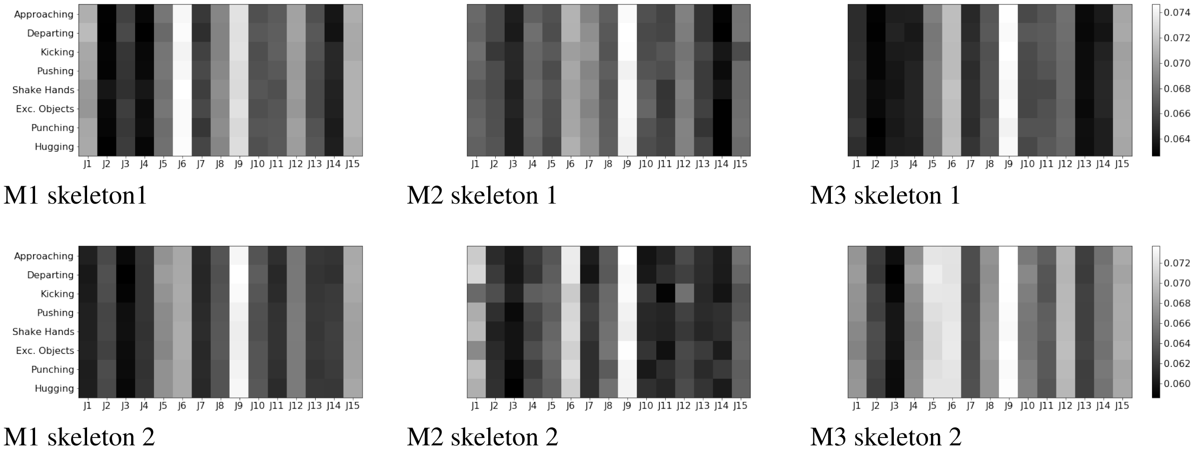
Figure 39.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **first person, (*lw*)** modeling using SBU Kinect interaction data.



M1 skeleton 1 M2 skeleton 1 M3 skeleton 1

M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

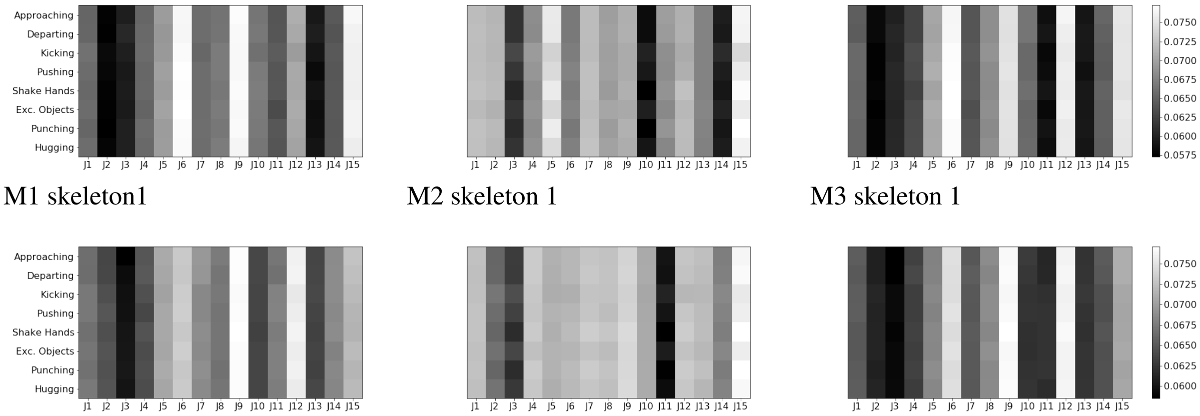
Figure 40.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **third person, (*pe*)** modeling using SBU Kinect interaction data.



M1 skeleton1 M2 skeleton 1 M3 skeleton 1

M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

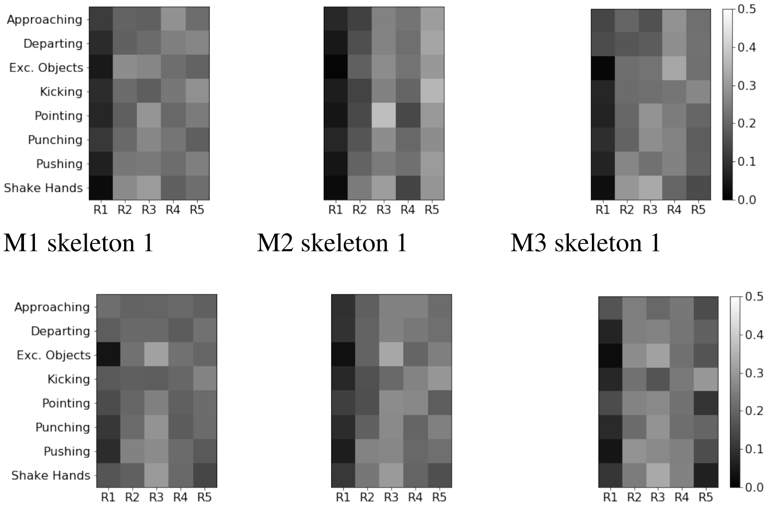
Figure 41.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **third person, (*lwpe*)** modeling using SBU Kinect interaction data.



M1 skeleton1 M2 skeleton 1 M3 skeleton 1

M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

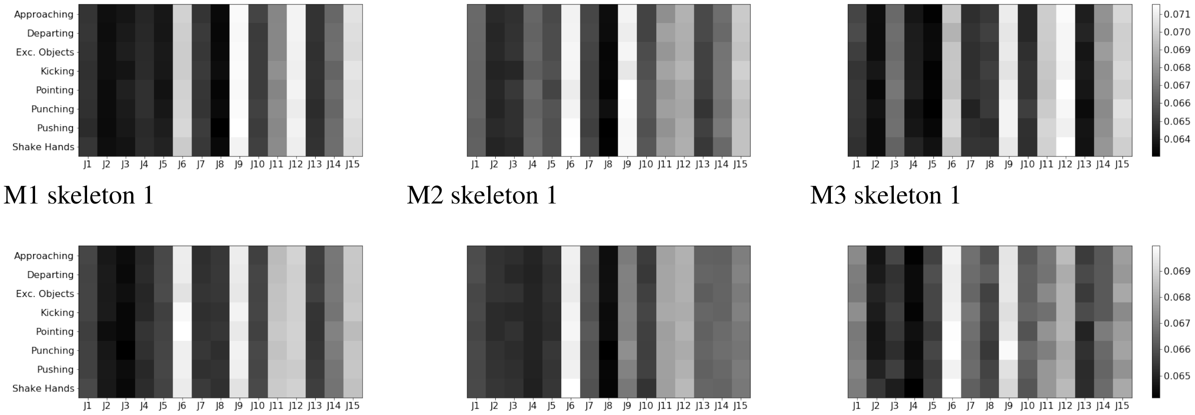
Figure 42.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **third person, (*lw*)** modeling using SBU Kinect interaction data.



M1 skeleton 1 M2 skeleton 1 M3 skeleton 1

M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

Figure 43.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **first person, (*pe*)** modeling using K3HI interaction data.



M1 skeleton 1

M2 skeleton 1

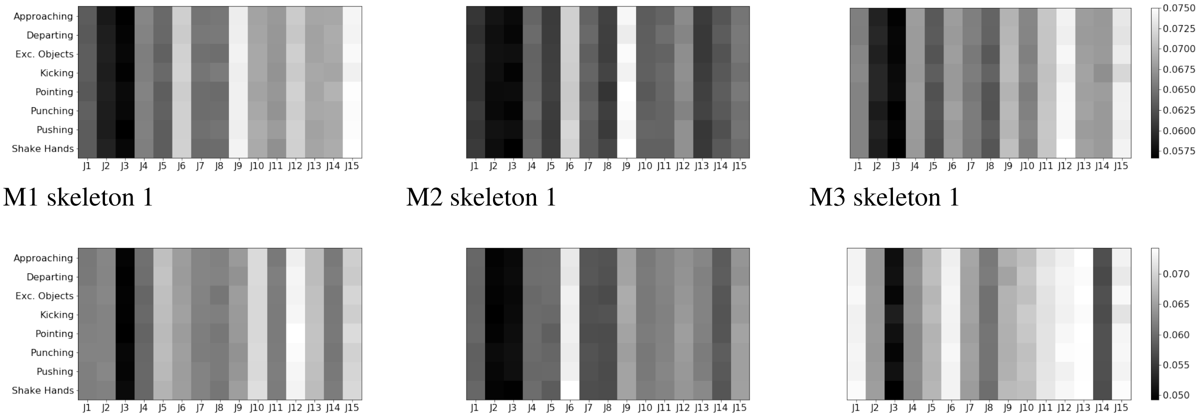
M3 skeleton 1

M1 skeleton 2

M2 skeleton 2

M3 skeleton 2

Figure 44.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **first person, (*lwpe*)** modeling using K3HI interaction data.



M1 skeleton 1

M2 skeleton 1

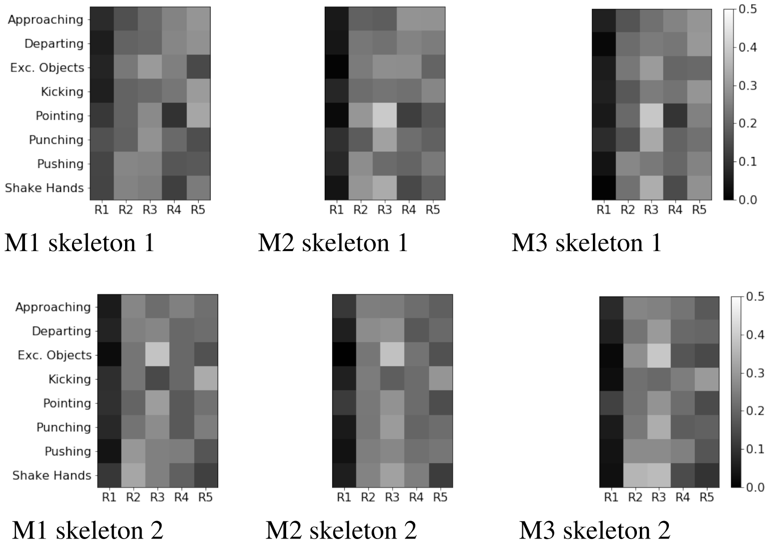
M3 skeleton 1

M1 skeleton 2

M2 skeleton 2

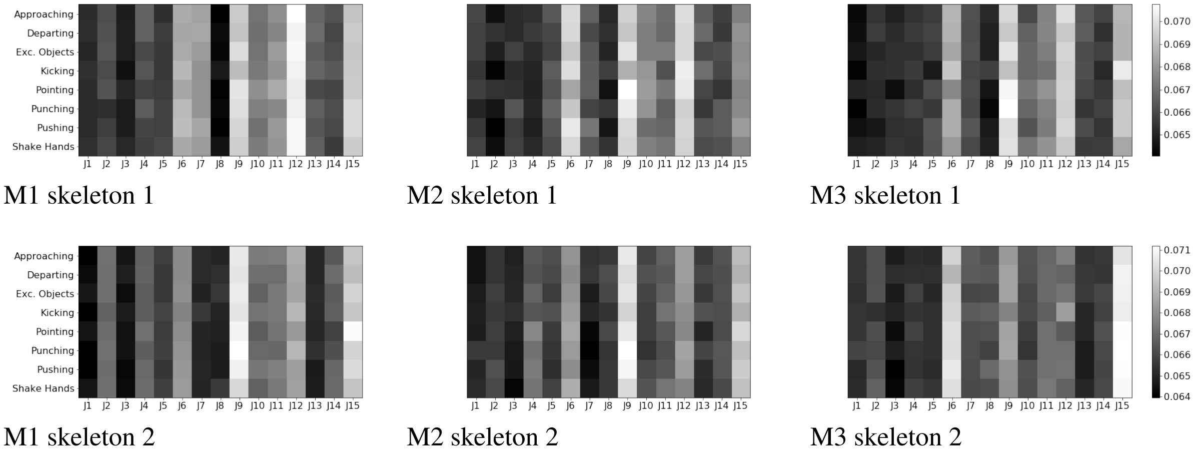
M3 skeleton 2

Figure 45.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **first person, (*lw*)** modeling using K3HI interaction data.


M1 skeleton 1 M2 skeleton 1 M3 skeleton 1

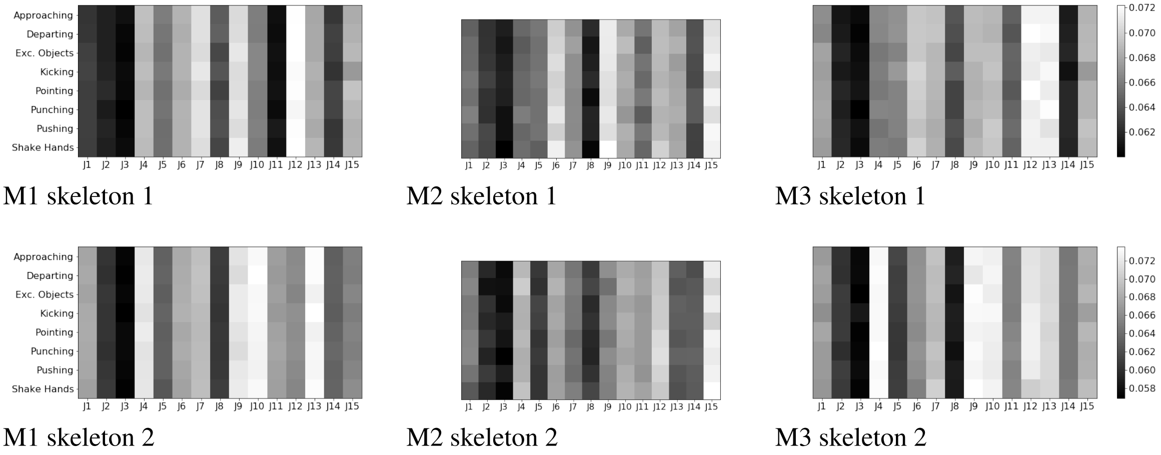
M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

Figure 46.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **third person, (*pe*)** modeling using K3HI interaction data.


M1 skeleton 1 M2 skeleton 1 M3 skeleton 1

M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

Figure 47.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **third person, (*lwpe*)** modeling using K3HI interaction data.



M1 skeleton 1 M2 skeleton 1 M3 skeleton 1

M1 skeleton 2 M2 skeleton 2 M3 skeleton 2

Figure 48.: Salient region distribution (dist.) over all interactions shown for skeleton 1 in (a–c) and the other skeleton in (d–f) for **third person, (*lw*)** modeling using K3HI interaction data.

joints) from both the skeletons at any time. Thus, the model at any time (video frame) observes the entire skeletons.

For a fair comparison, the number of layers and number of neurons in each layer are the same over all variations, including the baseline.

Evaluation metrics: We evaluate the generation using results using average frame distance (AFD), as in [Huang and Kitani, 2014]: $\frac{1}{T-1}\sum_t \|X_t^{(i)} - \hat{X}_t^{(i)}\|^2$, where $X_t^{(i)}$ and $\hat{X}_t^{(i)}$ are the true and predicted skeletal joint coordinates respectively at time t for modality i and T is the sequence length. We evaluate the classification using accuracy, recall, precision and F1-score.

7.3.4 Evaluation Results

Qualitative evaluation

Fig. 29. shows one time step ahead prediction of the two skeletons (perception and body proprioception) for four kinds of interactions from each dataset. These figures are shown for both first person and third person models, and for M1 using *pe* action selection method. The prediction over space and time looks quite realistic for all cases. Figs. 34., 36. show the visual and body proprioceptive pattern completions when 30%, 50% and 70% of the data have been observed. Our model generates realistic predictions over space and time for all the cases. As expected, short-term predictions are more accurate than long-term predictions. Even in the long-term, there is continuity and the two predicted skeletons are well synchronized. The proposed model's predicted action/reaction at each time step complies with the actual interactions.

Evaluation for generation accuracy

The AFD from first person modeling is lower than or comparable to third person modeling for most cases (ref. Tables 18.–21.). Modeling the two skeletons as distinct modalities helps in learning a better latent representation, resulting in more accurate generation. First person models have more parameters than third person models which also explains the lower AFD of the first person models.

First person: AFD is the lowest for *lwpe* and *bs* for SBU Kinect dataset and *bs* for K3HI dataset. AFD is the highest for *pe* for both datasets.

Third person: AFD is the lowest for *bs* for SBU Kinect interaction data and *lw* and *bs* for K3HI interaction data. AFD is the highest for *pe* for both datasets.

Within the same category for action selection, we do not observe much variation in AFD for the three models for both datasets (ref. Tables 18.–21.). The models do not vary much in terms of generating the skeletons, so we do not observe much variation in AFD for different models within the same category. The generation process is more dependent on different action selection methods. So we observe higher variation in AFD for different action selection methods (ref. Tables 18., 20.). The lower AFD of our model than the baseline in all the cases for both the datasets is because our model (*pe*) observes lesser than 49% of the ground truth for SBU Kinect interaction data and 48% for K3HI interaction data (ref. Tables 26., 27.).

Evaluation for classification accuracy

The classification accuracy for our first person models is higher than or comparable to our third person models in most cases. Also, the number of trainable parameters for first person models is higher than that of third person models.

First person: Among the three models, M1 yields the highest classification accuracy for almost all action selection methods for both the datasets (ref. Table 22., 24.). Among the three action selection methods, for SBU Kinect interaction data, *bs*, *lwpe*, *lw* yield the highest classification accuracy for M1, M2 and M3 respectively (ref. Table 22.), and for K3HI Interaction dataset, *bs* yields the highest classification accuracy for M1 and M3 and *pe* yields the highest classification accuracy for M2 (ref. Table 24.).

Third person: Among all the models, M1 gives the highest classification accuracy for all action selection methods for both the datasets (ref. Table 23., 25.). Among all the action selection methods, for SBU Kinect interaction data, *bs* yields the highest classification accuracy for M1 and M3, *pe* yields the highest classification accuracy for M2. Among all the action selection methods, for K3HI Interaction dataset, *pe* yields the highest classification accuracy for M2, *bs* yields the highest classification accuracy for M1 and M3 and *lwpe* yields the lowest classification accuracy for all cases.

M1 considers both the partial observations and the latent variables for predicting the class, due to which the classification accuracy is higher. M2 considers only the partial observations and not the latent variables for predicting the class. Our results show that including the latent variables to predict the class can make a great difference in the classification performance. Additionally, in M1, the classification modality shares parameters with the generation modality, whereas in M2, the classification modality does not share parameters with the generation modality, though in both cases the generation modality shares parameters with the classification modality. Thus, it is likely that the generation modality improves the classification results in M1 when compared to M2. M3 considers the generated data to predict the class, which explains its lower classification accuracy. As the generated skeletal data will deviate from the true skeletal data, the classification accuracy is low. We did not observe a consistent pattern in the performance accuracy due to different action selection methods for the same model. Thus, no action selection method is superior than the others. Results from our model, *pe* is comparable to or better than the baseline in all the cases for M1 and M2 (ref. Tables 22.–25.). Results from *lwpe* and *lw* are comparable to baseline, *bs* for M1 and M2 for K3HI dataset (ref. Table 24. and 25.).

Table 30. compares our most accurate models (for different settings and action selection methods) with relevant models reported in the literature. Results show that for SBU dataset, M2 *lwpe* and for K3HI dataset all models and action selection methods achieves higher classification accuracy than models that operate on raw skeletal data, as our model.

Analysis of action selection

We can visualize the distribution of attention weights assigned to the joints or regions as a heatmap (see Figs. 37.–48.). For each interaction class, this distribution over the joints/regions is computed from the sum of all weights W_t (ref. Eqs. 7.1–7.3) assigned to each joint/region.

The joints, whose movements have high variation over time, are more difficult to predict, and hence are more salient. Thus, the salient regions for punching, exchange objects, push, handshake, and hug are primarily the hands (e.g., punch in Figs. 37.c and 40.b; exchange object in Figs. 43.a and 43.f; push in Fig. 43.d; shake hands in Figs. 46.b and 46.f; hug in Figs. 40.a, 40.e, 37.b and 37.d) while for kicking, they are the legs (ref. Figs. 43.e and 46.f). This is not observed in some cases, such as kicking in Fig. 37.d, because the same skeleton might be the interaction initiator in some videos and the reactor in the others within the same dataset, thereby having different behaviors for the same interaction class.

We do not observe much variation in the distributions between M1, M2, M3 for the same action selection method. For any interaction, the weight distributions from *lwpe* and *lw* are similar. The attention weights are not very different for the different interactions.

Evaluation for efficiency

Efficiency of the model is evaluated by the percentage of scene observed for prediction.

Our experiments show that, during the first few sampling instants, both generation and classification accuracy improves exponentially (ref. Figs. 49.–52.). The saturation of the accuracy after that indicates our model does not need to sample a larger percentage of the data as ground truth for generation.

We compute the average (over all videos for each interaction) of the number of salient joints sampled by our model at each glimpse (ref. Table 26., 27.). We do not observe much variation in the average percentage for different models for both the datasets for both first and third person modeling. On average, for any interaction, for both datasets our model samples less than 49% and 48% of the joints in the case of FP and TP respectively. For both datasets, the highest sparsity is for kicking. The lowest sparsity is for punching (FP) and punching/pushing (TP) for SBU Kinect interaction dataset, and approaching/exchange object (FP) and approaching/departing (TP) for K3HI dataset.

Table 28.: Number of trainable parameters.

Model	First person	Third person
M1 (*bs*)	1656348	1089996
M2 (*bs*)	1134284	833676
M3 (*bs*)	1111420	827692
M1 (*pe*)	1656348	1089996
M2 (*pe*)	1134284	833676
M3 (*pe*)	1111420	827692
M1 (*lwpe*)	1657728	1092726
M2 (*lwpe*)	1135664	836406
M3 (*lwpe*)	1112800	830422
M1 (*lw*)	1657728	1092726
M2 (*lw*)	1135664	836406
M3 (*lw*)	1112800	830422

7.3.5 Design evaluation for different models

Handling missing class labels

All three models (M1, M2, M3) require true class labels to train for classification. A subset of parameters in each model is shared between the classification and generation pathways, albeit in unique ways (see Fig. 30.). In M1, the generation (completed pattern)

Table 29.: Training time (iterations) in hours.

Model	SBU		K3HI	
	First person	Third person	First person	Third person
M1 (*bs*)	1.0, 7368	.4, 4364	1.6, 5388	.7, 2452
M2 (*bs*)	1.5, 9201	.9, 8720	2.2, 5862	4.4, 17499
M3 (*bs*)	.4, 8250	.3, 8018	.7, 3459	.5, 3310
M1 (*pe*)	1.8, 7146	.5, 4166	5.2, 9154	1.8, 7199
M2 (*pe*)	2.7, 10627	1.0, 8282	3.8, 6673	5.6, 17430
M3 (*pe*)	.6, 8207	.3, 8105	1.0, 3255	.4, 2926
M1 (*lwpe*)	1.2, 5512	.5, 2844	2.7, 5421	2.5, 5832
M2 (*lwpe*)	3.4, 12169	2.6, 13030	6.5, 10350	8.7, 17499
M3 (*lwpe*)	.5, 7727	.4, 7586	.8, 2887	.6, 2685
M1 (*lw*)	1.4, 5203	1.3, 6889	4.6, 10519	2.0, 3350
M2 (*lw*)	4.0, 17999	3.4, 17999	7.0, 12352	8.2, 17499
M3 (*lw*)	.6, 8857	.5, 8541	.8, 2715	1.0, 3491

Table 30.: Comparison of classification accuracy. The other models cited in this table ([Nguyen, 2021, Li and Leung, 2016, Verma et al., 2021, Zhu et al., 2016, Liu et al., 2017, Du et al., 2015, Hu et al., 2013, Hu et al., 2019b]) are performing classification only (no generation). They take both skeletons as input, similar to our models. These works do not distinguish between first and third person environments.

Dataset	Characteristics			Models		Accuracy
	Raw skeleton	Skeletal features	Attention			
SBU		✓		Other models	[Nguyen, 2021]	96.3
		✓			[Li and Leung, 2016]	94.12
		✓			[Verma et al., 2021]	94.28
	✓				[Zhu et al., 2016]	90.41
	✓				[Liu et al., 2017]	93.3
	✓				[Du et al., 2015] as reported in [Zhu et al., 2016]	80.35
	✓			Our models (First Person)	M1 *(bs)*	93.2
	✓		✓		M1 *(pe)*	93.1
	✓		✓		M2 *(lwpe)*	93.8
	✓		✓		M1 *(lw)*	91.5
	✓			Our models (Third Person)	M1 *(bs)*	93.7
	✓		✓		M1 *(pe)*	92.5
	✓		✓		M2 *(lwpe)*	91.4
	✓		✓		M1 *(lw)*	92.9
K3HI		✓		Other models	[Hu et al., 2013]	83.33
		✓			[Hu et al., 2019b]	80.87
	✓				[Hu et al., 2013]	45.2
	✓				[Hu et al., 2019b]	48.54
	✓			Our models (First Person)	M1 *(bs)*	87.5
	✓		✓		M1 *(pe)*	85.9
	✓		✓		M2 *(lwpe)*	84.9
	✓		✓		M1 *(lw)*	86.9
	✓			Our models (Third Person)	M1 *(bs)*	83.0
	✓		✓		M1 *(pe)*	82.7
	✓		✓		M2 *(lwpe)*	82.1
	✓		✓		M1 *(lw)*	80.8

and class label are independent outputs. In M2, the class label is an input to the generative pathway, hence classification accuracy directly influences generation accuracy. In M3, the completed pattern is the input to the classification model, hence generation accuracy directly influences classification accuracy.

When class labels are missing, the generative parameters, including the shared parameters, are trained to minimize the generative loss only. All three models continue to infer irrespective of whether labels are present, noisy or missing, which makes them practical for real-world applications. A drawback of M2 is that the generation depends on the predicted class label; hence, the generation will be poor if the classification pathway is not well-trained. An advantage of M1 and M3 is that because the generation and classification pathways share parameters, even if the class labels are missing, the shared parameters will be updated by minimizing the generative error only which might improve the classification accuracy.

Number of trainable parameters

Number of trainable parameters for all the models is shown in Table 28.. Third person models has the lowest number of trainable parameters. M1 has the highest and M3 has the lowest number of trainable parameters. *lwpe* and *lw* have more trainable parameters than *pe* or *bs*.

Training time

Our models are implemented using TensorFlow 1.3 framework in Python 3.5.4. All experiments are carried out in HPC using PowerEdge R740 GPU nodes equipped with Tesla V100-PCIE-16GB.

Training time is the total time required for training the model on the trained set until the error converges. The training time for our models is shown in Table 29.. We report the average (over n-fold cross validation) convergence time in hours and the average number of iterations in Table 29.. In order to identify offline the iteration at which convergence occurs, we smooth the classification accuracy and the generation error curves by calculating the moving average with a 50-iteration window. For classification, we consider convergence is reached at the iteration when the average accuracy exceeds 90% of the highest accuracy, for M1, M2 and M3. When pretraining M3's generative model, convergence is reached at the iteration when the average error falls below 10% of the highest error.

For SBU Kinect Interaction data and both first person and third person, M3 and M2 require the least and highest amount of training time for all action selection methods. For K3HI Interaction data, M3 requires the lowest training time for all action selection methods for both first person and third person and M2 requires the highest training time for all action selection methods except M1 *(pe)* for first person.

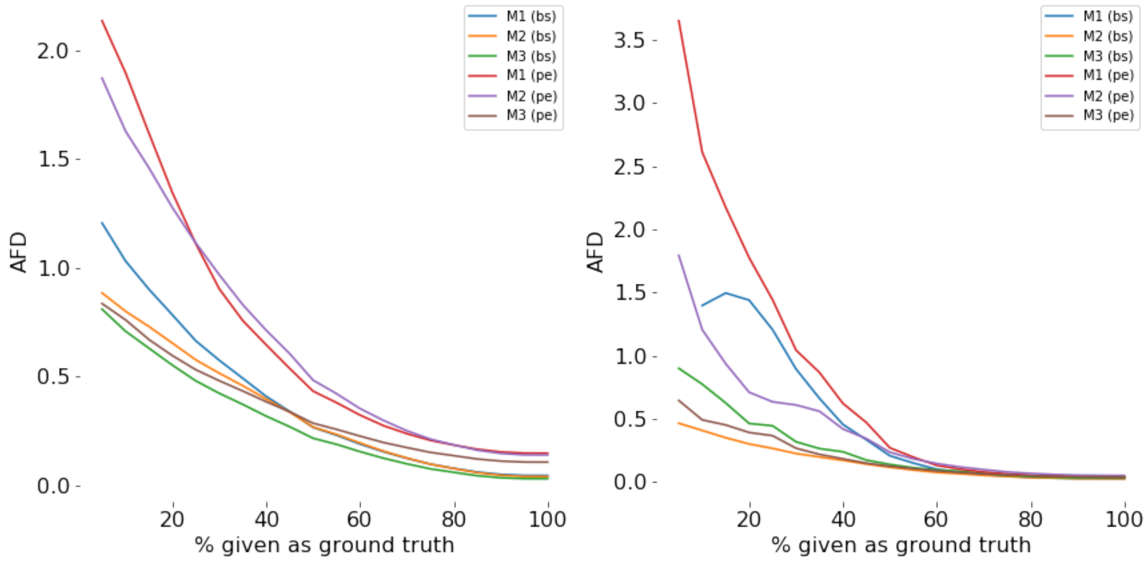
M3 is trained separately for prediction and classification tasks whereas in M1 and M2, the prediction and classification tasks are trained jointly. This shows that the model trained for a single task converges faster than when trained for multiple tasks.

End-to-end training

End-to-end training allows an entire model to be optimized for a given task(s) and dataset. However, the challenge is to search for the optimal set of parameter values in a very large space. This is often circumvented by *pretraining* selected components (layers, blocks, functions) in isolation for a number of iterations to initialize their parameters in a sub-optimal space. Then the entire model is trained end-to-end. In this paper, models M1 and M2 are trained end-to-end without any pretraining while M3 is not end-to-end.

7.4 Conclusions

A predictive agent model is proposed that sequentially samples and interacts with its environment which is constituted of 3D skeletons. At each instant, it samples a subset of skeleton joints to jointly minimize its classification and sensory prediction (or generation) errors in a greedy manner. The agent operates as a closed-loop system involving perceptual ('what') and proprioceptive ('where') pathways which are learned end-to-end. The model can be used for dynamic environments, and can scale to arbitrary

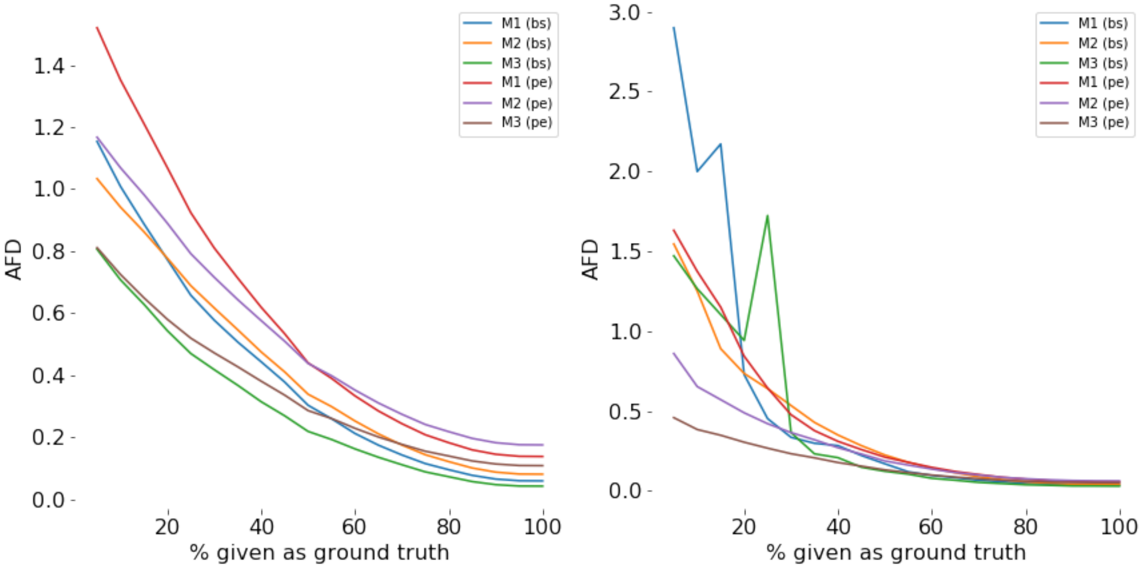


SBU Kinect Interaction Dataset

K3HI Interaction Dataset

Figure 49.: Prediction (AFD) for different percentage of ground truth given as input for **first person**. For any percentage p, $p\%$ of the actual data is given as input and the prediction is considered as input for the rest of the time steps.

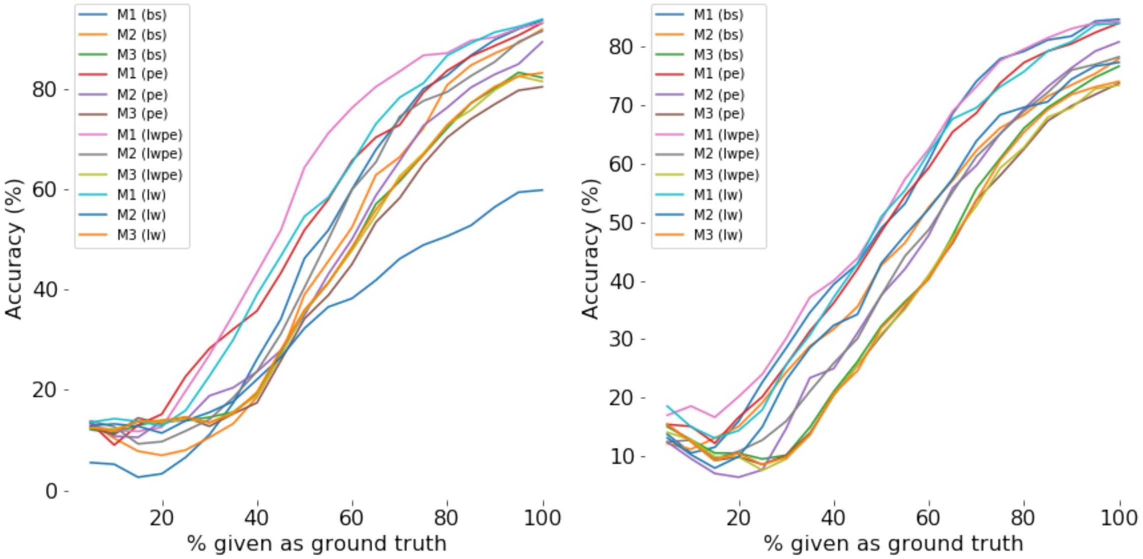
number of modalities. Experiments on interaction classification and generation on benchmark datasets reveal that the model is sample and size efficient, and yields state-of-the-art accuracy among models that operate on raw skeleton data. This is the first work to report a model's classification and generation accuracy on two-skeleton interaction videos, and in both first and third person settings.



SBU Kinect Interaction Dataset

K3HI Interaction Dataset

Figure 50.: Prediction (AFD) for different percentage of ground truth given as input for **third person**. For any percentage p, $p\%$ of the actual data is given as input and the prediction is considered as input for the rest of the time steps.



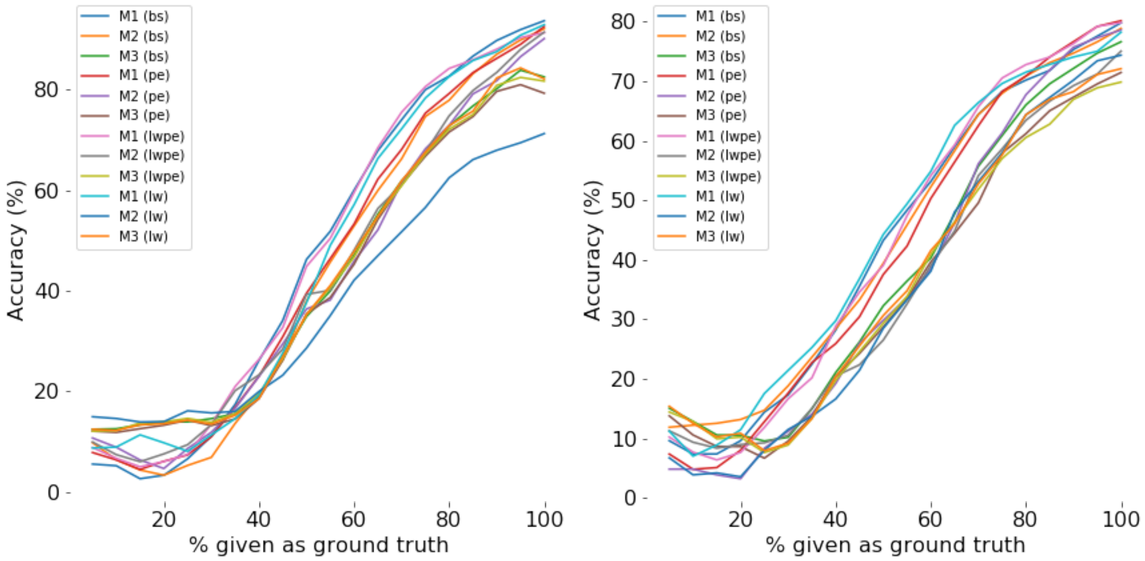
SBU Kinect Interaction Dataset

K3HI Interaction Dataset

Figure 51.: Classification accuracy for different percentage of ground truth given as input for **first person**. For any percentage p, $p\%$ of the actual data is given as input and the prediction is considered as input for the rest of the time steps.

Algorithm 10 $PatComClassModel2(x^{(1:2)}_{1:\tau})$

```
for $t \leftarrow 1$ $to$ $T$ do
    Classification Model
    $h^{cls}_t = RNN^{cls}_{\alpha}(h^{cls}_{t-1}, \mathbf{x}_{1:t})$
    $\hat{y}_t = softmax(h^{cls}_t)$
    $h^{'} = tanh(\hat{y}_t)$
    Recognition Model
    for $i \leftarrow 1$ $to$ $2$ do
        if $t > \tau$ then
            $x^{(i)}_t \leftarrow \hat{X}^{(i)}_t$
        end if
        $[\mu^{(i)}_{0,t}\,;\sigma^{(i)}_{0,t}] \leftarrow \varphi^{prior}(h^{(i)}_{t-1})$
        $[\mu^{(i)}_{z,t}\,;\sigma^{(i)}_{z,t}] \leftarrow \varphi^{enc}([x^{(i)}_t, h^{(i)}_{t-1}])$
    end for
    $[\mu^{(3)}_{0,t}\,;\sigma^{(i)}_{0,t}] \leftarrow \varphi^{prior}(h^{'})$
    $[\mu^{(3)}_{z,t}\,;\sigma^{(i)}_{z,t}] \leftarrow \varphi^{enc}([x^{(1)}_t, x^{(2)}_t, h^{'}])$
    Product of Experts
    $z_t \sim \mathcal{N}(\mu_{0,t}, \Sigma_{0,t}),$ where $\Sigma_{0,t} = \Big(\sum_{i=1}^{3} \Sigma^{(i)}_{0,t}{}^{-2}\Big)^{-1}$
        and $\mu_{0,t} = \Big(\sum_{i=1}^{3} \mu^{(i)}_{0,t}\Sigma^{(i)}_{0,t}{}^{-2}\Big)\Sigma_{0,t}$
    $z_t|\mathbf{x}_t \sim \mathcal{N}(\mu_{z,t}, \Sigma_{z,t}),$ where $\Sigma_{z,t} = \Big(\sum_{i=1}^{3} \Sigma^{(i)}_{z,t}{}^{-2}\Big)^{-1}$
        and $\mu_{z,t} = \Big(\sum_{i=1}^{3} \mu^{(i)}_{z,t}\Sigma^{(i)}_{z,t}{}^{-2}\Big)\Sigma_{z,t}$
    Generative Model
    for $i = 1$ $to$ $2$ do
        $h^{(i)}_t \leftarrow RNN_{\theta}(h^{(i)}_{t-1}, [z_t, x^{(i)}_t])$
        $[\mu^{(i)}_{x^{(i)},t}\,;\sigma^{(i)}_{x^{(i)},t}] \leftarrow \varphi^{dec}([h^{(i)}_{t-1}, z_t])$
        $\hat{X}^{(i)}_t \leftarrow \mu^{(i)}_{x^{(i)},t}$
    end for
end for
```



SBU Kinect Interaction Dataset

K3HI Interaction Dataset

Figure 52.: Classification accuracy for different percentage of ground truth given as input for **third person**. For any percentage p, $p\%$ of the actual data is given as input and the prediction is considered as input for the rest of the time steps.

Chapter 8

Speech Emotion Recognition via Generation using an Attention-based Variational Recurrent Neural Network

Abstract: The last five years have seen an exponential rise in the number of attention-based models for speech emotion recognition (SER). Most of these models use a spectrogram as the input speech representation and the CNN or RNN or convolutional RNN as the key machine learning (ML) component, and learn feature weights to implement attention. We propose an attention-based model for SER that uses MFCC as the input speech representation and a variational RNN (VRNN) as the key ML component. Since the MFCC is of lower dimension than a spectrogram, the model is size- and data-efficient. The VRNN has been used for problems in vision but rarely for SER. Our model is predictive in nature. At each instant, it infers the emotion class and generates the next observation, computes the generation error, and selectively samples (attends to) the locations of high error. Thus, attention emerges in our model, and does not require learning feature weights. This simple model provides interesting insights when evaluated for SER on the RAVDESS and IEMOCAP benchmark datasets. This model is the first to explore simultaneous generation and recognition for SER. The generation capability is a necessity for efficient recognition in the model.

8.1 Introduction

Manifestations of emotions are often involuntary. They convey one's true feelings, confidence, intentions, and expectations, which are useful for social interaction. Speech contains linguistic and paralinguistic content; both allow the conveyance of emotions albeit in different ways. The recognition of emotion from speech or *speech emotion recognition* (SER) has been an active area of research in machine learning (ML) for decades, contributing to technologies for human–machine interaction, intelligent tutoring, healthcare, and security.

This paper is concerned with SER without explicitly processing or analyzing the linguistic content in the speech. A large number of ML models have been proposed for this problem (see [El Ayadi et al., 2011, Schuller, 2018, Akçay and Oğuz, 2020] for reviews). Models incorporating attention mechanism often yield superior performance, though at the expense of additional parameters. Attentional models for SER can be largely categorized based on three aspects: speech representation used as input to the ML model, the key ML component in the model, and the implementation of attention.

The input speech is often represented as a linear- or mel-frequency spectrogram (e.g., [Mirsamadi et al., 2017, Zhang et al., 2018, Chen et al., 2018, Hsiao and Chen, 2018, Yoon et al., 2019, Xie et al., 2019, Li et al., 2019, Alex et al., 2020, Lin and Busso, 2020, Seo and Kim, 2020, Mao et al., 2020, Yu and Kim, 2020, Kwon et al., 2020b, Kwon et al., 2021]). A few studies have experimented with mel-frequency cepstral coefficients (MFCCs) (e.g., [Xu et al., 2021]), low-level descriptors (e.g., [Ramet et al., 2018]), or raw speech (e.g., [Kwon et al., 2020a]) as input. In [Neumann and Vu, 2017], the authors investigated the impact of four types of features on the performance of an attentional convolutional neural network (CNN), namely (1) 26 log mel filter banks (logMel), (2) 13 MFCCs, (3) 25 low-level descriptors (frequency- and energy-related parameters and spectral parameters) constituting the extended Geneva minimalistic acoustic parameter set (eGeMAPS), and (4) a prosody feature set consisting of PCM loudness, F0 contour, envelope of F0 contour, voicing probability, local jitter, differential jitter, and local shimmer. From their experiments with the IEMOCAP database, it was concluded that the model and training data (improvised vs. scripted speech) are more impactful than the features.

Models for SER often involve the CNN (e.g., [Neumann and Vu, 2017, Zhang et al., 2018, Seo and Kim, 2020, Mao et al., 2020, Kwon et al., 2020b, Kwon et al., 2021, Xu et al., 2021]), recurrent neural network (RNN) or long short-term memory (LSTM) (e.g., [Mirsamadi et al., 2017, Ramet et al., 2018, Hsiao and Chen, 2018, Yoon et al., 2019, Li et al., 2019, Xie et al., 2019, Yu and Kim, 2020, Alex et al., 2020, Lin and

Busso, 2020]), or convolutional RNN/LSTM (e.g., [Chen et al., 2018, Kwon et al., 2020a]). The variational autoencoder (VAE) has been rarely used (e.g., [Latif et al., 2017]). Most models utilize *feature-level attention* where the output of hidden layers are weighted. *Decision-level attention* applies to multiple-instance learning [Maron and Lozano-Pérez, 1997] where the prediction of the instances are weighted to obtain the bag-level prediction. In both types of attention, the weights are learned along with other parameters to optimize an objective. A CNN with feature-level attention significantly outperforms its decision-level counterpart on benchmark datasets [Mao et al., 2020].

Contributions. We propose an attention-based model for SER that involves the MFCC and a VRNN. The MFCC is a compressed representation of a lower dimension than a spectrogram, which allows the model to be size- and data-efficient. The unique properties of the proposed model are as follows:

(1) At each instant, the model simultaneously infers the emotion class and generates the observation (input MFCC vector corresponding to the speech window) at the next instant. Training a model by minimizing generation and classification errors in conjunction leads to more stability and less overfitting due to each error acting as a regularizer for the other [Marino et al., 2016].

(2) Attention emerges in our model due to prediction error. The model selectively samples locations in the input MFCC vector that contain unexpected observations. This saves the use of additional parameters for attention.

(3) Our SER experiments using the RAVDESS and IEMOCAP benchmark datasets show that the proposed end-to-end model yields high classification accuracy by sampling a fraction of the MFCC vector for each window, in addition to providing insights into the model design.

8.2 Models and Methods

8.2.1 Preliminaries

Generative model. Given a set of data points x, a generative model p_{model} with parameters θ maximizes the log-likelihood, $\mathcal{L}(x;\theta)$, of the data.

Evidence lower bound (ELBO). Let the data x be generated by a latent continuous random variable z. Then, computing the log-likelihood requires integrating the marginal likelihood, $\int p_{model}(x,z)dz$, which is intractable [Kingma and Welling, 2013]. In variational inference, an approximation of the intractable posterior is optimized by defining an evidence lower bound (ELBO) on the log-likelihood, $\mathcal{L}(x;\theta) \leq \log p_{model}(x;\theta)$.

Variational autoencoder (VAE) is a multilayered generative model. It assumes an isotropic Gaussian prior, $p_\theta(z)$, and i.i.d. data samples. VAE maximizes the following ELBO [Kingma and Welling, 2013]:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}[q_\phi(z|x), p_\theta(z)] \tag{8.1}$$

where $p_\theta(x|z)$ and $q_\phi(z|x)$ are generative and recognition models respectively, $\mathbb{E}$ denotes expectation, and D_{KL} denotes Kullback-Leibler divergence. The first and second terms capture accuracy and complexity respectively. The negative of this ELBO is also known as *variational free energy*, minimization of which has been hypothesized as a general principle guiding brain function [Friston, 2010].

8.2.2 Problem Statement

Let $\mathbf{X} = \langle X_1, X_2, \ldots, X_T\rangle$ be a sequence of observable variables representing an environment, where T is the sequence length. Let $\mathbf{x}_{\leq t} = \langle x_1, \ldots, x_t\rangle$ $(1 \leq t \leq T)$ be a partial observation of $\mathbf{X}$. Let $\mathbf{y} = \langle y_1, \ldots, y_T\rangle$, where y_t represents the true class label at time t. We define the *prediction* problem as generating $\mathbf{X}$ and $\mathbf{y}$ as accurately as possible from the partial observation $\mathbf{x}_{\leq t}$. At any time t, the objective is to maximize the joint likelihood of X_{t+1} and y_t, given $\mathbf{x}_{\leq t}$ and a generative model p_θ with parameters θ, i.e., $\arg\max_\theta p_\theta(X_{t+1}, y_t|\mathbf{x}_{\leq t})$.

8.2.3 Proposed Model

Our model comprises of four functions: environment, prediction, action selection, and learning. See Fig. 53..

1. Environment. The environment is the source of sensory data or observations. It is time-varying. Our model interacts with the environment by selectively sampling observations at each time instant.

2. Prediction. The model predicts using a VRNN involving two processes: recognition and generation.

Recognition (Encoder). The probabilistic encoder, $q_\phi(z_t|\mathbf{x}_{\leq t})$, produces a Gaussian distribution over the possible values of the code z_t from which the given observations could have been generated. An RNN with one layer of LSTM units constitute the recognition model. The RNN generates the parameters for the approximate posterior distribution, $q_\phi(z_t|\mathbf{x}_{\leq t})$, from the observations. The prior is sampled from a standard normal distribution, $p_\theta(z_t) \sim \mathcal{N}(0, 1)$, as in [Gregor et al., 2015].

The function of the encoder is shown in Lines 1–3 in Algorithm 12, where RNN_ϕ^{enc} represents the function of an LSTM unit, φ^{enc} is a function that returns the mean and the logarithm of the standard deviation as a linear function of the hidden state, as in [Chung et al., 2015].

Generation (Decoder). The decoder, $p_\theta(X_{t+1}, y_t|\mathbf{x}_{\leq t}, z_{\leq t})$, generates the perceptual data and the class label from the latent variables, z_t, at each time step. The generative model has two RNNs, each with one layer of hidden LSTM units.

Each RNN generates the parameters of its data distribution. The data is then sampled from this distribution. In our model, X_{t+1} is sampled from a multivariate Gaussian distribution with mean and variance generated by the corresponding decoder

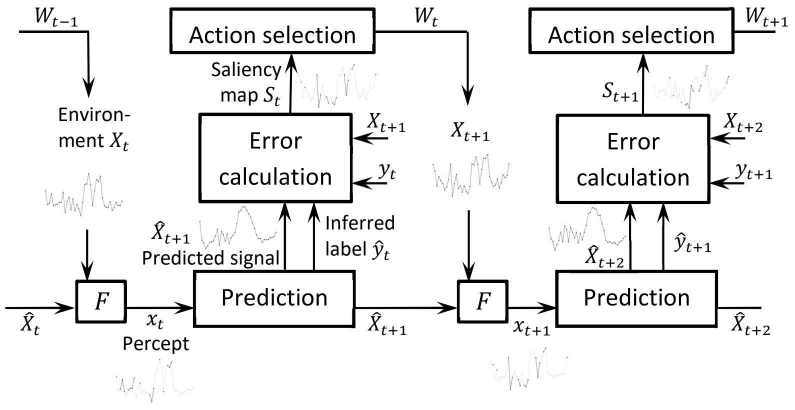


Figure 53.: Flow diagram of the proposed model.

RNN, and y_t is sampled from a multivariate Bernoulli distribution with mean generated by the corresponding decoder RNN.

The decoder equations are shown in Lines 4–8 of Algorithm 12, where functions RNN_θ^{dec} and φ^{dec} are the same as RNN_ϕ^{enc} and φ^{enc} respectively.

3. Action selection. In the proposed model, action selection amounts to deciding the binary weight (attention) given to each location of the current observation. At any time t, a saliency map S_t is computed from which the action is determined. The saliency map assigns a salience score $S_{t,l}$ to each element l of the input MFCC vector.

The weights are determined by thresholding the prediction error. The threshold is statistically estimated on the fly and is not predetermined.

$$S_t = |X_{t+1} - \hat{X}_{t+1}| \tag{8.2}$$

$$W_{t,l} = \begin{cases} 1, & \text{if } S_{t,l} \geq \frac{1}{n}\sum_{k=1}^{n} S_{t,k} \\ 0, & \text{otherwise} \end{cases} \tag{8.3}$$

$$x_{t+1} = W_t \odot X_{t+1} + (\mathbf{1} - W_t) \odot \hat{X}_{t+1} \tag{8.4}$$

where X_{t+1}, $\hat{X}_{t+1}$ are the true and predicted data (MFCC vectors) respectively, $|.|$ denotes the absolute value, $\odot$ denotes elementwise product, and n is the dimension of a MFCC vector. Eq. 8.2–8.4 are written in compact form in Line 7 of Algorithm 11.

Algorithm 11 The proposed model

```
Initialize parameters of the generative model $\theta$, recognition model $\phi$, sequence length $T$.
Initialize optimizer parameters: $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\eta = 0.001$, $\epsilon = 10^{-10}$.
Initialize $W_0 \leftarrow \mathbf{1}$ and $x_1 \leftarrow \psi(X_1, W_0)$, where $W_0$ are the weights for the initial sampling, and the function $\psi$ generates a sample $x_1$ from the environment $X_1$ after assigning weights $W_0$ (ref. Action selection in Section 8.2.3).
while true do
    for $\tau \leftarrow 1 \; to \; T$ do
        $\hat{X}_{\tau+1}, \hat{y}_\tau \leftarrow Predict(x_{1:\tau})$
        Action execution (ref. Eq. 8.2–8.4)
        $x_{\tau+1} \leftarrow F(X_{\tau+1}, \hat{X}_{\tau+1})$
        Learning
        Update $\{\theta, \phi\}$ by maximizing Eq. 8.5.
    end for
end while
```

Due to the nature of the chosen threshold function, at least one element of the MFCC vector will be salient and at least one element will be non-salient at any time. Our experiments show that variable number of salient features at each time step is more effective. Fixing the number of salient features to a constant occasionally leads to selection of features with low saliency or overlooking features with high saliency. In the proposed model, only the salient MFCC features are sampled. For the non-salient features, the observation at time $t+1$ is the predicted observation from t.

4. Learning. The objective is to maximize the expression in Eq. 8.5, which can be derived from the objectives for multimodal VAE [Wu and Goodman, 2018], VRNN [Chung et al., 2015], and VAE for classification [Kingma et al., 2014].

$$\begin{aligned}\mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \lambda_1 \log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{<t}) \\ + \lambda_2 \log p_\theta(y_t|z_{\leq t}, \mathbf{x}_{<t})\Big] - \sum_{t=1}^{T} \beta D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big) \qquad (8.5)\end{aligned}$$

where λ_1, λ_2, β are the weights balancing the terms.

Algorithm 12 $Predict(x_{1:\tau})$

```
x Recognition model (encoder)
$h_\tau^{enc} \leftarrow RNN_\phi^{enc}(x_\tau, h_{\tau-1}^{enc})$
$[\mu_\tau \,; \Sigma_\tau] \leftarrow \varphi^{enc}(h_{\tau-1}^{enc})$
$z_\tau | \mathbf{x}_\tau \sim \mathcal{N}(\mu_\tau, \Sigma_\tau)$
x Generative model (decoder)
$h_\tau^{dec_1} \leftarrow RNN_\theta^{dec}(h_{\tau-1}^{dec_1}, z_\tau)$
$[\mu_{x,\tau} \,; \Sigma_{x,\tau}] \leftarrow \varphi^{dec}(h_\tau^{dec_1})$
$\hat{X}_{\tau+1} \leftarrow \mu_{x,\tau}$
x Classification model (decoder)
$h_\tau^{dec_2} \leftarrow RNN_\theta^{dec}(h_{\tau-1}^{dec_2}, z_\tau)$
$\hat{y}_\tau \leftarrow softmax(h_{\tau-1}^{dec_2})$
```

8.3 Experimental Results

8.3.1 Datasets

We evaluate the model on two datasets: Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), and Interactive Emotional Dyadic Motion Capture (IEMOCAP). We split each dataset into 80% train and 20% test.

RAVDESS [Livingstone and Russo, 2018] is an audio-visual dataset of emotional speech and songs. We consider the audio modality and the emotional speech data, as in earlier works (e.g., [Xu et al., 2021, Yu and Kim, 2020, Kwon et al., 2021, Kwon et al., 2020a]). The considered data has 1440 audio files, vocalized by 24 professional actors (12 female, 12 male). The speech includes neutral, calm, happy, sad, angry, fearful, surprise, and disgust expressions.

The IEMOCAP is an acted, multimodal and multispeaker database, collected at SAIL lab at USC in five sessions. We consider the audio modality and the improvised speech data with four emotions (happy, sad, anger, neutral), as in [Yu and Kim, 2020]. We perform five-fold cross-validation by considering each session as one fold.

8.3.2 Experimental setup

Data preprocessing: From each audio file, we extract windows of 2 second duration, with 50% overlap. From each window, we extract 40 MFCCs using the Librosa library in Python. Thus, the input to our model at any instant t is a 40-dimensional MFCC vector. An audio file comprises of a sequence of T such vectors. Classification error is computed at the end of each audio file. We ignore windows of duration less than 2 seconds.

Training details: For RAVDESS, the recognition, generation and classification models consist of 512, 256, 512 hidden units respectively, and the latent variable dimension is 20. For IEMOCAP, the three models consist of 64 hidden units each, and the latent variable dimension is 10. These parameters are estimated experimentally. T is variable as the audio files are of different durations. We consider a minibatch size of 256. The parameters $\beta = 1$, $\lambda_1 = 1$, $\lambda_2 = 50$ are fixed. The model is learned end-to-end using backpropagation and RMSProp optimization with a learning rate of 0.001. These hyperparameters are estimated via cross-validation from the training set.

For RAVDESS, we use a dropout probability of 0.3 for recognition and generation hidden layers, and 0.1 for classification layer to prevent overfitting. For IEMOCAP, we use 0.8 for all three layers. The KL-divergence term in the objective function also acts as a regularizer [Kingma and Welling, 2013] that prevents overfitting.

Ablation study: The utility of attention in our model is analyzed using an ablation study. We create a model by eliminating attention (Line 7 in Algorithm 11) from the proposed model. The VRNN is modified such that all locations in the observation are sampled with equal weight at any time. Thus, the model always observes the entire true MFCC vector. For a fair comparison, the number of layers and neurons in each layer for this non-attentional model is kept consistent with the original model.

Evaluation metrics: To evaluate recognition accuracy, we use the common metric, weighted accuracy, as computed in [Xu et al., 2021].[1] Efficiency of the model is evaluated in terms of the proportion of MFCC vector sampled for prediction.

Table 31.: Weighted classification accuracy (%).

Data	Model (year)	Feature	WA
RAVDESS	Ours	MFCC	78.5
	Ours w/o attn.	MFCC	79.9
	[Xu et al., 2021] (2021)	MFCC	77.8
	[Zeng et al., 2019] (2019)	Spectrogram	64.5
	[Kwon et al., 2020b] (2020)	Spectrogram	79.5[1]
	[Kwon et al., 2021] (2021)	Spectrogram	80.0[1]
	[Li et al., 2020] (2020)	Low-level features	72.0
IEMOCAP	Ours	MFCC	65.5
	Ours w/o attn.	MFCC	64.4
	[Yu and Kim, 2020] (2020)	IS09 + Mel spectrogram	67.7
	[Zhang et al., 2018] (2018)	Spectrogram	70.4
	[Ramet et al., 2018] (2018)	Low-level descriptors	68.8
	[Neumann and Vu, 2017] (2017)	logMel filterbanks	62.1

Table 32.: Percentage of MFCCs (total 40) deemed salient enough to be sampled by our model from the ground truth.

Dataset	Neutral	Happy	Sad	Angry	Fearful	Disgust	Surprised	Calm	Average
RAVDESS	65.8	66.1	62.5	61.3	63.1	61.6	68.5	61.5	63.8
IEMOCAP	47.1	44.7	44.2	44.9	–	–	–	–	45.2

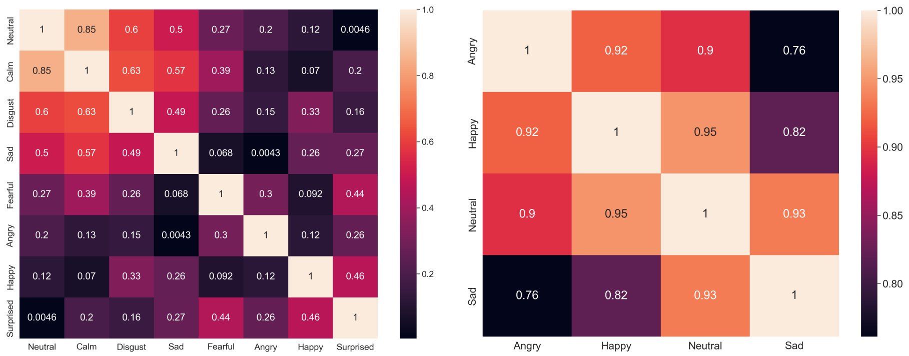
8.3.3 Evaluation results

Evaluation for accuracy: We compare the classification accuracy of our model with recent works that use similar experimental setup as ours (ref. Table 31.). Accuracy of our model is comparable to the state-of-the-art for RAVDESS dataset. Our model yields higher accuracy when compared to the state-of-the-art models using MFCC as features for RAVDESS. Accuracy of our model is comparable to some of those reported for IEMOCAP dataset. The number of parameters in our model is 22.6M and 12.78M for RAVDESS and IEMOCAP respectively. The latter appears to be much less than the model in [Zhang et al., 2018], which reported the state-of-the-art accuracy for IEMOCAP. Most works in this area did not report the number of model parameters which makes fair comparison a challenge. Increasing our model size does not increase accuracy by much.

[1] Computation of unweighted accuracy in [Kwon et al., 2020b, Kwon et al., 2021] is similar to that of weighted accuracy in [Xu et al., 2021].

Generation error guides the attention of our model, which leads to efficiency by allowing selective sampling of the input observation. Hence, generation capability is a necessity in our model. The ablation study reveals that the classification accuracy of our model with and without attention are comparable, for both datasets (ref. Table 31.). This shows that the generation and classification pathways do not interact adversely, and that the dynamic threshold used to compute attention weights (ref. Eq. 8.3) is a good choice to balance accuracy and efficiency.

Analysis of action selection: We visualize the similarity of attention weights between emotion classes in Fig. 54.. For each emotion, the mean of weight vectors assigned to the MFCC vectors is computed. This mean weight vector for an emotion corresponds to the expected attention to the different MFCC features from the proposed model. The similarity of a pair of mean weight vectors is computed as the absolute of their normalized cosine similarity. For RAVDESS, 'neutral' is most similar to 'calm' and least to 'surprised'. For IEMOCAP, 'neutral' is most similar to 'sad' and least to 'angry'.



RAVDESS dataset

IEMOCAP dataset

Figure 54.: These matrices show the similarity between the (expected) attention for each pair of emotion classes.

Evaluation for efficiency: We compute the average (over all audios for each emotion) proportion of input observation (MFCC vector) sampled by our model at each time step (ref. Table 32.). On average, for any interaction, our model samples 63.8% and 45.2% of the observation for RAVDESS and IEMOCAP respectively. The highest sparsity is for ‘angry’ emotion from RAVDESS and ‘sad’ emotion from IEMOCAP. The proposed model samples its observations efficiently without compromising accuracy.

8.4 Discussion

Attention. The attention mechanism in our model differs from most SER models from behavioral and algorithmic perspectives. Typically, end-to-end attention-based models for SER learn all parameters (including attention weights) by optimizing an objective function. In most of these models, attention is an internal mechanism that does not have a corresponding behavior. The attention parameters play a role similar to any other parameter in the model. In our model, attention is a parameterless mechanism that emerges due to prediction error, which drives action/behavior (ref. Eq. 8.2–8.4). This mechanism is interpretable as the model simply attends to its unexpected observations.

From an algorithmic perspective, SER models utilize attention weights at a higher feature level (e.g., [Mirsamadi et al., 2017, Zhang et al., 2018, Chen et al., 2018, Hsiao and Chen, 2018, Yoon et al., 2019, Xie et al., 2019, Alex et al., 2020, Lin and Busso, 2020, Kwon et al., 2021, Neumann and Vu, 2017]), or at multiple feature levels (e.g., [Ramet et al., 2018, Kwon et al., 2021]). Our model utilizes attention at the input level only.

Many SER models compute attention weights from multiple time steps (e.g., [Mirsamadi et al., 2017, Chen et al., 2018, Hsiao and Chen, 2018, Yoon et al., 2019, Alex et al., 2020, Lin and Busso, 2020, Ramet et al., 2018, Neumann and Vu, 2017]). Such models need to process the input sequence till the final time, which introduces latency. Our model computes attention from the current time only. It infers by processing the input till the current time, which allows it to be efficient.

Speech representation. Even though the authors in [Neumann and Vu, 2017] downplayed the role of speech representation, we believe the performance of our model can be improved by using a spectrogram instead of MFCCs. The spectrogram is less compact and contains more information than MFCC. Hence, using the spectrogram might increase the size and enhance the accuracy of our model. Most SER models with state-of-the-art accuracy use spectrogram.

ML component. VRNN-based models are known to perform well in computer vision applications (e.g., [Baruah and Banerjee, 2020a, Baruah and Banerjee, 2020b, Baruah et al., 2022]). However, VRNNs have rarely been explored for SER. Our model using VRNN yields encouraging results. Convolution has been shown to be an effective operation for SER. Our model's accuracy might be improved by using a convolutional VRNN.

Generation and classification are done jointly in very few end-to-end ML models. The models reported in [Wu and Goodman, 2018, Kingma et al., 2014] generate and classify handwritten numerals. Classification accuracy is not reported in [Wu and Goodman, 2018]. In [Marino et al., 2016], a sentiment analysis model was proposed to generate and classify positive and negative reviews. The model yields lower classification error by jointly minimizing classification and generation losses than that by minimizing the former only. None of these models incorporate attention. For the problem of SER, the proposed model is the first to explore simultaneous generation and recognition.

8.5 Conclusions

We propose an attention-based predictive model for SER, where the key ML component is a VRNN. For efficiency, MFCC is used as the input speech representation. This model is the first to explore simultaneous generation and classification for SER. Generation error guides the attention of the model, leading to efficiency by selective sampling. The sampled observations are used for classification. Hence, the model performs recognition via generation. Our experiments using two benchmark datasets reveal that the model yields high recognition accuracy without compromising efficiency in terms of model size and sparsity of sampled observations. Using spectrogram as speech representation and a convolutional VRNN as the ML component might improve the accuracy and reduce the efficiency of this model.

Chapter 9

Conclusions

In this chapter, we discuss the findings from each chapter and possible future work.

9.1 Conclusions

In Chapter 2, we investigated the optimal modality selection problem for time-series data in the context of late fusion. We analyzed multimodal emotion or action classification using four late fusion methods and five benchmark datasets. Our experimental analysis on product, average, Bayesian and majority voting late fusion methods show that the fusion methods perform differently based on the posterior distribution estimated by each modality. Our results show that for different fusion methods, increasing the number of modalities might not necessarily increase the classification accuracy. We analyze multiple methods for selecting a subset of modalities for late fusion and observe that information gain is an useful measure for selecting modalities which is consistent for all the datasets. The classification accuracy obtained from the selected subset of modalities is comparable to the highest accuracy in all cases.

In Chapter 3, a predictive agent model is proposed that sequentially samples and interacts with its environment. At each instant, it samples the location with maximum information gain to minimize its sensory prediction error in a greedy manner. The agent operates as a closed-loop system involving perceptual (‘what’) and proprioceptive (‘where’) pathways which are learned end-to-end, without supervision (class labels) or reinforcement. The same model can be used for static and dynamic environments. Experiments on handwriting generation reveal that the model is sample and size efficient, and yields state-of-the-art accuracy. Conceptually, this work is unique due to its modeling action/attention as proprioception, using it with perception in a multimodal setting, and experimentally validating its role in yielding state-of-the-art accuracy in an end-to-end model.

In Chapter 4, we introduced an mcAT dataset for recognizing handwritten numerals and alphabets via sequential sampling. The data is collected from 382 participants presented with images selected from benchmark datasets (MNIST, EMNIST). On average, 169.1 responses per numeral/alphabet class are recorded. The data is rigorously analyzed to reveal the efficiency of human visual recognition. The participants observed only 12.8% of an image for recognition. We proposed a baseline model to predict the location and class(es) a participant would select at the next sampling. We showed how our experimental conditions and data may be used to evaluate an attention-based reinforcement model in comparison to human performance. This mcAT dataset, with multiple benefits over eye-tracking data, fills an important gap in attention-based models research.

In Chapter 5, we extend our proposed model in Chapter 3 for classification. At each sampling instant, the agent has to complete and classify the partial sequence observed till that instant. Very few end-to-end attention-based models reported in the literature perform generation and classification of handwritten numerals/alphabets jointly. Our agent model is learned by jointly minimizing the classification and generation errors. Three variants of this model are evaluated on benchmark datasets. Their accuracies are comparable and correlate with the model size. Our experiments reveal that the proposed model is more data-efficient in handwritten numeral/alphabet recognition than human participants as well as a highly-cited attention-based reinforcement model, under the same conditions and stimuli. Qualitatively, the participants' fixation maps are more similar to our model's fixation maps than the reinforcement model's. To the best of our knowledge, this is the first attention-based end-to-end agent of its kind for recognition via generation, with high degree of accuracy and efficiency.

In Chapter 6, we extend our proposed model in Chapter 3 for human interaction generation. Experimental results using our agent for two-person interaction forecasting are comparable to non-attentional models even though our agent's observations have higher than 50% sparsity. The agent model is learned end-to-end in an unsupervised manner, without any reinforcement signal or utilities/values of states. This is the first work on an attention-based agent that actively samples its environment guided by prediction error and generates realistic 3D human skeleton interactions.

In Chapter 7, we extend our model in Chapter 5 for human interaction generation and recognition. Experiments on interaction classification and generation on benchmark datasets reveal that the model is sample and size efficient, and yields state-of-the-art accuracy among models that operate on raw skeleton data. This is the first work to report a model's classification and generation accuracy on two-skeleton interaction videos, and in both first and third person settings.

In Chapter 8, we extend our proposed model in Chapter 7 for speech emotion recognition. Our model provides interesting design insights when evaluated for SER on the RAVDESS and IEMOCAP benchmark datasets.

9.2 Application in autonomous driving

In this section, we state how we can apply our model for radar target classification in autonomous driving application.

In autonomous driving, radar sensor gives information about other vehicles in the vicinity of the vehicle loaded with the sensor. A radar processing pipeline assumes the world is made of point-like reflective objects. Features such as doppler velocity, range etc. can be extracted from the radar signal reflected from such an object [Mostajabi et al., 2020]. These objects can be classified as acceptable or unacceptable for further processing.

If X_t represent the features at time t, y_t represent the corresponding label and T is the number of samples from the object, our objective function can be written as:

$$
\begin{aligned}
&\log p_\theta(X_{\le T}, y_{\le T} | \mathbf{x}_{\le T}) \\
&\ge \mathbb{E}_{q_\phi(z_{\le T} | \mathbf{x}_{\le T})} \Big[\sum_{t=1}^{T} \lambda_1 \log p_\theta(X_t | z_{\le t}, \mathbf{x}_{<t}) \\
&\quad + \lambda_2 \log p_\theta(y_t | z_{\le t}, \mathbf{x}_{<t}) \Big] \\
&\quad - \sum_{t=1}^{T} \beta D_{KL}\big(q_\phi(z_t | \mathbf{x}_{\le t}), p_\theta(z_t)\big)
\end{aligned}
$$

where λ_1, λ_2, β are the weights balancing the terms.

9.3 Application in healthcare

In this section, we state how we can apply our model for concept normalization for medical record data in healthcare.

In this problem, the input to the model is a sequence of words from a test name. Each test name is assigned a Logical Observation Identifiers Names and Codes (LOINC) label [Laboratories, 2022].

If X_t represent a word embedding in the test name for word t, y_t represent the corresponding LOINC label, and T is the number of words in the test name, our objective function can be written as:

$$\begin{aligned}
&\log p_\theta(X_{\leq T}, y_{\leq T}|\mathbf{x}_{\leq T}) \\
&\geq \mathbb{E}_{q_\phi(z_{\leq T}|\mathbf{x}_{\leq T})}\Big[\sum_{t=1}^{T} \lambda_1 \log p_\theta(X_t|z_{\leq t}, \mathbf{x}_{<t}) \\
&\quad + \lambda_2 \log p_\theta(y_t|z_{\leq t}, \mathbf{x}_{<t})\Big] \\
&\quad - \sum_{t=1}^{T} \beta D_{KL}\big(q_\phi(z_t|\mathbf{x}_{\leq t}), p_\theta(z_t)\big)
\end{aligned}$$

where λ_1, λ_2, β are the weights balancing the terms.

REFERENCES

Adeli, V., Adeli, E., Reid, I., Niebles, J. C., and Rezatofighi, H. (2020). Socially and contextually aware human motion and pose forecasting. *IEEE Robotics and Automation Letters*, 5(4):6033–6040.

Akçay, M. B. and Oğuz, K. (2020). Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Commun.*, 116:56–76.

Alex, S. B., Mary, L., and Babu, B. P. (2020). Attention and feature selection for automatic speech emotion recognition using utterance and syllable-level prosodic features. *Circuits, Syst. Signal Process.*, 39(11):5681–5709.

Atrey, P. K., Hossain, M. A., El Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379.

Atrey, P. K., Kankanhalli, M. S., and Oommen, J. B. (2007). Goal-oriented optimal subset selection of correlated multimedia streams. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 3(1):2.

Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv:1412.7755*.

Ba, J., Salakhutdinov, R. R., Grosse, R. B., and Frey, B. J. (2015). Learning wake-sleep recurrent attention models. In *NIPS*, pages 2593–2601.

Banerjee, B. and Chandrasekaran, B. (2010a). A constraint satisfaction framework for executing perceptions and actions in diagrammatic reasoning. *J. Artif. Intell. Res.*, pages 373–427.

Banerjee, B. and Chandrasekaran, B. (2010b). A spatial search framework for executing perceptions and actions in diagrammatic reasoning. In *Diagrammatic Representation and Inference, LNAI*, volume 6170, pages 144–159. Springer, Heidelberg.

Banerjee, B. and Dutta, J. K. (2014a). SELP: A general-purpose framework for learning the norms from saliencies in spatiotemporal data. *Neurocomput.*, 138:41–60.

Banerjee, B. and Dutta, J. K. (2014b). SELP: A general-purpose framework for learning the norms from saliencies in spatiotemporal data. *Neurocomputing*, 138:41–60.

Baradel, F., Wolf, C., and Mille, J. (2017). Pose-conditioned spatio-temporal attention for human action recognition. *arXiv preprint arXiv:1703.10106*.

Barsoum, E., Kender, J., and Liu, Z. (2018). HP-GAN: Probabilistic 3D human motion prediction via GAN. In *CVPR Workshops*, pages 1418–1427.

Baruah, M. and Banerjee, B. (2020a). A multimodal predictive agent model for human interaction generation. In *CVPR Workshop*.

Baruah, M. and Banerjee, B. (2020b). The perception-action loop in a predictive agent. In *CogSci 2020: 42nd Annual Meeting of the Cognitive Science Society*, pages 1171–1177. `https://cognitivesciencesociety.org/cogsci20/papers/0215/0215.pdf`.

Baruah, M. and Banerjee, B. (2021). A dataset for handwritten numeral and alphabet recognition via sequential sampling. *arXiv preprint arXiv:TBD*.

Baruah, M., Banerjee, B., and Nagar, A. K. (2022). An attention-based predictive agent for static and dynamic environments. *IEEE Access*, 10:17310–17317.

Baumeister, R. F., Masicampo, E. J., and Vohs, K. D. (2011). Do conscious thoughts cause behavior? *Annu. Rev. Psychol.*, 62:331–361.

Busso, C., Deng, Z., et al. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proc. 6th international conference on Multimodal interfaces*, pages 205–211. ACM.

Bütepage, J., Kjellström, H., and Kragic, D. (2018). Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In *ICRA*, pages 1–9. IEEE.

Bylinskii, Z., Judd, T., Oliva, A., Torralba, A., and Durand, F. (2018). What do different evaluation metrics tell us about saliency models? *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(3):740–757.

Castellano, G., Kessous, L., and Caridakis, G. (2008). Emotion recognition through multiple modalities: face, body gesture, speech. In *Affect and emotion in human-computer interaction*, pages 92–103. Springer.

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015a). Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Transactions on Human-Machine Systems*, 45(1):51–61.

Chen, C., Jafari, R., and Kehtarnavaz, N. (2015b). Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *IEEE International Conference on Image Processing*, pages 168–172.

Chen, M., He, X., Yang, J., and Zhang, H. (2018). 3-d convolutional recurrent neural networks with attention model for speech emotion recognition. *IEEE Signal Process. Lett.*, 25(10):1440–1444.

Chiu, H., Adeli, E., Wang, B., Huang, D., and Niebles, J. (2019). Action-agnostic human pose forecasting. In *WACV*, pages 1423–1432. IEEE.

Chopin, B., Otberdout, N., Daoudi, M., and Bartolo, A. (2021). Human motion prediction using manifold-aware wasserstein gan. *arXiv preprint arXiv:2105.08715*.

Chowdhury, A. K., Tjondronegoro, D., Chandran, V., and Trost, S. G. (2018). Physical activity recognition using posterior-adapted class-based fusion of multiaccelerometer data. *IEEE journal of biomedical and health informatics*, 22(3):678–685.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In *NIPS*, pages 2980–2988.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *IJCNN*, pages 2921–2926. IEEE.

de Jong, E. D. (2016). Incremental sequence learning. *arXiv:1611.03068*.

Du, Y., Wang, W., and Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118.

Dutta, J. K. and Banerjee, B. (2014). Learning features and their transformations from natural videos. In *IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments*, pages 55–61, Orlando, FL.

Dutta, J. K. and Banerjee, B. (2015). Online detection of abnormal events using incremental coding length. In *AAAI*, pages 3755–3761.

Dutta, J. K. and Banerjee, B. (2017). Variation in classification accuracy with number of glimpses. In *IJCNN*, pages 447–453. IEEE.

Dutta, J. K., Banerjee, B., and Reddy, C. K. (2016). RODS: Rarity based outlier detection in a sparse coding framework. *IEEE Trans. Knowl. Data Eng.*, 28(2):483–495.

Egner, S. et al. (2018). Attention and information acquisition: Comparison of mouse-click with eye-movement attention tracking. *J. Eye Mov. Res.*, 11(6).

Egner, S., Itti, L., and Scheier, C. (2000). Comparing attention models with different types of behavior data. *Investig. Ophthalmol. Vis. Sci.*, 41(4):S39.

El Ayadi, M., Kamel, M. S., and Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognit.*, 44(3):572–587.

Elsayed, G., Kornblith, S., and Le, Q. V. (2019). Saccader: improving accuracy of hard attention models for vision. In *NIPS*, pages 702–714.

Eslami, S. M., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. (2016). Attend, infer, repeat: Fast scene understanding with generative models. *arXiv:1603.08575*.

Fan, Z., Zhao, X., Lin, T., and Su, H. (2018). Attention-based multiview re-observation fusion network for skeletal action recognition. *IEEE Transactions on Multimedia*, 21(2):363–374.

Fernando, T., Denman, S., Sridharan, S., and Fookes, C. (2018). Soft+ hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection. *Neural Netw.*, 108:466–478.

Fragkiadaki, K., Levine, S., Felsen, P., and Malik, J. (2015). Recurrent network models for human dynamics. In *ICCV*, pages 4346–4354.

Friston, K. (2009). The free-energy principle: A rough guide to the brain? *Trends Cogn. Sci.*, 13(7):293–301.

Friston, K. J. (2010). The free-energy principle: A unified brain theory? *Nat. Rev. Neurosci.*, 11(2):127–138.

Friston, K. J., Adams, R. A., Perrinet, L., and Breakspear, M. (2012). Perceptions as hypotheses: Saccades as experiments. *Front. Psychol.*, 3:151.

Friston, K. J., Daunizeau, J., and Kiebel, S. J. (2009). Reinforcement learning or active inference? *PloS One*, 4(7):e6421.

Fukui, H., Hirakawa, T., Yamashita, T., and Fujiyoshi, H. (2019). Attention branch network: Learning of attention mechanism for visual explanation. In *CVPR*, pages 10705–10714.

Ghosh, P., Song, J., Aksan, E., and Hilliges, O. (2017). Learning human motion models for long-term predictions. In *Intl. Conf. 3D Vision*, pages 458–466. IEEE.

Goebel, K., Yan, W., and Cheetham, W. (2002). A method to calculate classifier correlation for decision fusion. *Proceedings of decision and control*, pages 135–140.

Goodfellow, I. (2016). NIPS 2016 tutorial: Generative adversarial networks. *arXiv:1701.00160*.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. *arXiv:1502.04623*.

Gregor, K., Danihelka, I., Mnih, A., Blundell, C., and Wierstra, D. (2014). Deep autoregressive networks. In *ICML*, pages 1242–1250. PMLR.

Gui, L., Wang, Y., Liang, X., and Moura, J. (2018). Adversarial geometry-aware human motion prediction. In *ECCV*, pages 786–803.

Gunes, H. and Piccardi, M. (2005). Affect recognition from face and body: early fusion vs. late fusion. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 4, pages 3437–3443. IEEE.

Han, J., Waddington, G., Adams, R., Anson, J., and Liu, Y. (2016). Assessing proprioception: A critical review of methods. *J. Sport Health Sci.*, 5(1):80–90.

Hoshen, Y. (2017). Vain: Attentional multi-agent predictive modeling. In *NIPS*, pages 2701–2711.

Hsiao, P.-W. and Chen, C.-P. (2018). Effective attention mechanism in dynamic models for speech emotion recognition. In *ICASSP*, pages 2526–2530.

Hu, D., Wang, C., Nie, F., and Li, X. (2019a). Dense multimodal fusion for hierarchically joint representation. In *ICASSP*, pages 3941–3945.

Hu, T., Zhu, X., Guo, W., and Su, K. (2013). Efficient interaction recognition through positive action representation. *Math. Probl. Eng.*, 2013.

Hu, T., Zhu, X., Wang, S., and Duan, L. (2019b). Human interaction recognition using spatial-temporal salient feature. *Multimedia Tools and Applications*, 78(20):28715–28735.

Huang, D. and Kitani, K. (2014). Action-reaction: Forecasting the dynamics of human interaction. In *ECCV*, pages 489–504. Springer.

Isler, V. and Bajcsy, R. (2005). The sensor selection problem for bounded uncertainty sensing models. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 20. IEEE Press.

Ivanov, Y., Serre, T., and Bouvrie, J. (2005). Error weighted classifier combination for multi-modal human identification.

Jiang, M. et al. (2015). Salicon: Saliency in context. In *CVPR*, pages 1072–1080.

Kankanhalli, M. S., Wang, J., and Jain, R. (2006). Experiential sampling on multiple data streams. *IEEE transactions on multimedia*, 8(5):947–955.

Kapourchali, M. H. and Banerjee, B. (2018). Unsupervised feature learning from time-series data using linear models. *IEEE Internet of Things Journal*, 5(5):3918–3926.

Kapourchali, M. H. and Banerjee, B. (2019). State estimation via communication for monitoring. *IEEE Trans. Emerg. Topics Comput. Intell.*

Kapourchali, M. H. and Banerjee, B. (2020). EPOC: Efficient perception via optimal communication. In *AAAI*.

Kim, N. W. et al. (2017). BubbleView: An interface for crowdsourcing image importance maps and tracking visual attention. *ACM Trans. Comput. Hum. Interact.*, 24(5):1–40.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kingma, D. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv:1312.6114*.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

Kittler, J., Hatef, M., Duin, R., and Matas, J. (1998). On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239.

Koelstra, S., Muhl, C., et al. (2012). Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31.

Kothari, P., Kreiss, S., and Alahi, A. (2021). Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*.

Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., and Duin, R. P. (2000). Is independence good for combining classifiers? In *icpr*, page 2168. IEEE.

Kwon, S. et al. (2020a). Clstm: Deep feature-based speech emotion recognition using the hierarchical convlstm network. *Mathematics*, 8(12):2133.

Kwon, S. et al. (2020b). A cnn-assisted enhanced audio signal processing for speech emotion recognition. *Sensors*, 20(1):183.

Kwon, S. et al. (2021). Att-net: Enhanced emotion recognition system using lightweight self-attention module. *Applied Soft Computing*, 102:107101.

Laboratories, M. C. (2022 (accessed February 22, 2022)). Loinc. `https://www.mayocliniclabs.com/test-catalog/appendix/loinc-codes`.

Lam, K., Cheng, R., Liang, B., and Chau, J. (2004). Sensor node selection for execution of continuous probabilistic queries in wireless sensor networks. In *Proc. ACM 2nd international workshop on Video surveillance & sensor networks*, pages 63–71.

Larochelle, H. and Hinton, G. E. (2010). Learning to combine foveal glimpses with a third-order boltzmann machine. In *NIPS*, pages 1243–1251.

Latif, S., Rana, R., Qadir, J., and Epps, J. (2017). Variational autoencoders for learning latent representations of speech emotion: A preliminary study. *arXiv:1712.08708*.

Le, T. M., Inoue, N., and Shinoda, K. (2018). A fine-to-coarse convolutional neural network for 3d human action recognition. *arXiv preprint arXiv:1805.11790*.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324.

Li, C., Zhang, Z., Lee, W. S., and Lee, G. H. (2018a). Convolutional sequence to sequence model for human dynamics. In *CVPR*, pages 5226–5234.

Li, C., Zhong, Q., Xie, D., and Pu, S. (2018b). Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv preprint arXiv:1804.06055*.

Li, H., Ding, W., Wu, Z., and Liu, Z. (2020). Learning fine-grained cross modality excitement for speech emotion recognition. *arXiv preprint arXiv:2010.12733*.

Li, M. and Leung, H. (2016). Multiview skeletal interaction recognition using active joint interaction graph. *IEEE Transactions on Multimedia*, 18(11):2293–2302.

Li, M. and Leung, H. (2019). Multi-view depth-based pairwise feature learning for person-person interaction recognition. *Multimedia Tools and Applications*, 78(5):5731–5749.

Li, R., Wu, Z., Jia, J., Bu, Y., Zhao, S., and Meng, H. (2019). Towards discriminative representation learning for speech emotion recognition. In *IJCAI*, pages 5060–5066.

Lin, W.-C. and Busso, C. (2020). An efficient temporal modeling approach for speech emotion recognition by mapping varied duration sentences into fixed number of chunks. In *Interspeech*, pages 2322–2326.

Lin, X. and Amer, M. (2018). Human motion modeling using dvgans. *arXiv:1804.10652*.

Liu, J., Shahroudy, A., Xu, D., Kot, A. C., and Wang, G. (2017). Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3007–3021.

Livingstone, S. R. and Russo, F. A. (2018). The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391.

Ma, X., Zhou, C., and Hovy, E. (2019). Mae: Mutual posterior-divergence regularization for variational autoencoders. *arXiv:1901.01498*.

Maaloe, L., Fraccaro, M., Liévin, V., and Winther, O. (2019). Biva: A very deep hierarchy of latent variables for generative modeling. In *NIPS*, pages 6551–6562.

Mangai, U. G., Samanta, S., Das, S., and Chowdhury, P. R. (2010). A survey of decision fusion and feature fusion strategies for pattern classification. *IETE Technical review*, 27(4):293–307.

Manzi, A., Fiorini, L., Limosani, R., Dario, P., and Cavallo, F. (2018). Two-person activity recognition using skeleton data. *IET computer Vision*, 12(1):27–35.

Mao, S., Ching, P., Kuo, C.-C. J., and Lee, T. (2020). Advancing multiple instance learning with attention modeling for categorical speech emotion recognition. *arXiv:2008.06667*.

Marino, D. L., Amarasinghe, K., and Manic, M. (2016). Simultaneous generation-classification using LSTM. In *IEEE Symp. Ser. Comput. Intell.*, pages 1–8.

Maron, O. and Lozano-Pérez, T. (1997). A framework for multiple-instance learning. *NIPS*, 10.

Matzen, L. E., Stites, M. C., and Gastelum, Z. N. (2021). Studying visual search without an eye tracker: An assessment of artificial foveation. *Cogn. Res. Princ. Implications*, 6(1):1–22.

Mirsamadi, S., Barsoum, E., and Zhang, C. (2017). Automatic speech emotion recognition using recurrent neural networks with local attention. In *ICASSP*, pages 2227–2231.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. *NIPS*, 27:2204–2212.

Mostajabi, M., Wang, C. M., Ranjan, D., and Hsyu, G. (2020). High-resolution radar dataset for semi-supervised learning of dynamic objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 100–101.

Najnin, S. and Banerjee, B. (2015). Improved speech inversion using general regression neural network. *The Journal of the Acoustical Society of America*, 138(3):EL229–EL235.

Najnin, S. and Banerjee, B. (2017). A predictive coding framework for a developmental agent: Speech motor skill acquisition and speech production. *Speech Commun.*, 92:24–41.

Najnin, S. and Banerjee, B. (2019). Speech recognition using cepstral articulatory features. *Speech Commun.*, 107:26–37.

Navalpakkam, V. et al. (2013). Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proc. Int. Conf. WWW*, pages 953–964.

Neumann, M. and Vu, N. T. (2017). Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech. *arXiv:1706.00612*.

Ng, E., Xiang, D., Joo, H., and Grauman, K. (2020). You2me: Inferring body pose in egocentric video via first and second person interactions. In *CVPR*, pages 9890–9900.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

Nguyen, X. S. (2021). Geomnet: A neural network based on riemannian geometries of spd matrix space and cholesky space for 3d skeleton-based interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13379–13389.

Niu, G., Han, T., Yang, B.-S., and Tan, A. C. C. (2007). Multi-agent decision fusion for motor fault diagnosis. *Mechanical Systems and Signal Processing*, 21(3):1285–1299.

Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., and Bajcsy, R. (2013). Berkeley mhad: A comprehensive multimodal human action database. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 53–60. IEEE.

Oord, A. V. D., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. *arXiv:1601.06759*.

Pahalawatta, P., Pappas, T., and Katsaggelos, A. (2004). Optimal sensor selection for video-based target tracking in a wireless sensor network. In *International Conference on Image Processing*, volume 5, pages 3073–3076. IEEE.

Picard, R. W., Vyzas, E., and Healey, J. (2001). Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE transactions on pattern analysis and machine intelligence*, 23(10):1175–1191.

Poria, S., Cambria, E., Howard, N., Huang, G.-B., and Hussain, A. (2016). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, 174:50–59.

Qin, Y., Mo, L., Li, C., and Luo, J. (2020). Skeleton-based action recognition by part-aware graph convolutional networks. *The visual computer*, 36(3):621–631.

Ramet, G., Garner, P. N., Baeriswyl, M., and Lazaridis, A. (2018). Context-aware attention mechanism for speech emotion recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 126–131. IEEE.

Ranzato, M. A. (2014). On learning where to look. *arXiv:1405.5488*.

Reiss, A. and Stricker, D. (2012a). Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, page 40. ACM.

Reiss, A. and Stricker, D. (2012b). Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pages 108–109. IEEE.

Rensink, R. A. (2000). The dynamic representation of scenes. *Vis. Cogn.*, 7(1-3):17–42.

Russell, S. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 4th edition.

Sadeghi, H., Andriyash, E., Vinci, W., Buffoni, L., and Amin, M. H. (2019). Pixelvae++: Improved pixelvae with discrete prior. *arXiv:1908.09948*.

Salakhutdinov, R. and Murray, I. (2008). On the quantitative analysis of deep belief networks. In *ICML*, pages 872–879.

Sang, H. F., Chen, Z. Z., and He, D. K. (2020). Human motion prediction based on attention mechanism. *Multimed. Tools. Appl.*, 79(9):5529–5544.

Schuller, B. W. (2018). Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. *Commun. ACM*, 61(5):90–99.

Seo, M. and Kim, M. (2020). Fusing visual attention cnn and bag of visual words for cross-corpus speech emotion recognition. *Sensors*, 20(19):5559.

Sermanet, P., Frome, A., and Real, E. (2014). Attention for fine-grained categorization. *arXiv:1412.7054*.

Smallwood, J. and Schooler, J. W. (2015). The science of mind wandering: Empirically navigating the stream of consciousness. *Annu. Rev. Psychol.*, 66:487–518.

Soleymani, M., Lichtenauer, J., Pun, T., and Pantic, M. (2012). A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing*, 3(1):42–55.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*.

Spratling, M. (2012). Predictive coding as a model of the V1 saliency map hypothesis. *Neural Netw.*, 26:7–28.

Standvoss, K., Quax, S. C., and Van Gerven, M. A. (2020). Visual attention through uncertainty minimization in recurrent generative models. *BioRxiv*.

Tripathi, S., Acharya, S., Sharma, R. D., Mittal, S., and Bhattacharya, S. (2017). Using deep and convolutional neural networks for accurate emotion classification on deap dataset. In *AAAI*, pages 4746–4752.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. *arXiv:2007.03898*.

van Beers, R. J. (2007). The sources of variability in saccadic eye movements. *J. Neurosci.*, 27(33):8757–8770.

Varshneya, D. and Srinivasaraghavan, G. (2017). Human trajectory prediction using spatially aware deep attention models. *arXiv:1705.09436*.

Vemula, A., Muelling, K., and Oh, J. (2018). Social attention: Modeling attention in human crowds. In *ICRA*, pages 1–7. IEEE.

Verma, A., Meenpal, T., and Acharya, B. (2021). Multiperson interaction recognition in images: A body keypoint based feature image analysis. *Computational Intelligence*, 37(1):461–483.

Vinayavekhin, P., Chaudhury, S., Munawar, A., Agravante, D., Magistris, G., Kimura, D., and Tachibana, R. (2018). Focusing on what is relevant: Time-series learning and understanding using attention. In *ICPR*, pages 2624–2629. IEEE.

Wiem, M. B. H. and Lachiri, Z. (2016). Emotion assessing using valence-arousal evaluation based on peripheral physiological signals and support vector machine. In *Control Engineering & Information Technology (CEIT), 2016 4th International Conference on*, pages 1–5. IEEE.

Wiem, M. B. H. and Lachiri, Z. (2017). Emotion classification in arousal valence model using mahnob-hci database. *Int. J. Adv. Comput. Sci. Appl. IJACSA*, 8(3).

Wu, M. and Goodman, N. (2018). Multimodal generative models for scalable weakly-supervised learning. In *NIPS*, pages 5575–5585.

Wu, Y., Chang, E. Y., Chang, K. C.-C., and Smith, J. R. (2004). Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 572–579. ACM.

Xie, Y., Liang, R., Liang, Z., Huang, C., Zou, C., and Schuller, B. (2019). Speech emotion classification using attention-based lstm. *IEEE/ACM Trans. Audio Speech Lang. Process.*, 27(11):1675–1685.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057.

Xu, M., Zhang, F., and Zhang, W. (2021). Head fusion: Improving the accuracy and robustness of speech emotion recognition on the iemocap and ravdess dataset. *IEEE Access*, 9:74539–74549.

Xu, Y. T., Li, Y., and Meger, D. (2019). Human motion prediction via pattern completion in latent representation space. In *Conf. Computer and Robot Vision*, pages 57–64. IEEE.

Yao, T., Wang, M., Ni, B., Wei, H., and Yang, X. (2018). Multiple granularity group interaction prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2246–2254.

Yoon, S., Byun, S., Dey, S., and Jung, K. (2019). Speech emotion recognition using multi-hop attention mechanism. In *ICASSP*, pages 2822–2826.

Yu, J., Gao, H., Yang, W., Jiang, Y., Chin, W., Kubota, N., and Ju, Z. (2020). A discriminative deep model with feature fusion and temporal attention for human action recognition. *IEEE Access*, 8:43243–43255.

Yu, Y. and Kim, Y.-J. (2020). Attention-lstm-attention model for speech emotion recognition and analysis of iemocap database. *Electronics*, 9(5):713.

Yun, K., Honorio, J., Chattopadhyay, D., Berg, T., and Samaras, D. (2012). Two-person interaction detection using body-pose features and multiple instance learning. In *CVPR Workshops*, pages 28–35. IEEE.

Zeng, Y., Mao, H., Peng, D., and Yi, Z. (2019). Spectrogram based multi-task audio classification. *Multimedia Tools and Applications*, 78(3):3705–3722.

Zhang, Y., Du, J., Wang, Z., Zhang, J., and Tu, Y. (2018). Attention based fully convolutional network for speech emotion recognition. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1771–1775. IEEE.

Zhao, S., Song, J., and Ermon, S. (2017). Infovae: Information maximizing variational autoencoders. *arXiv:1706.02262*.

Zheng, Y., Zemel, R. S., Zhang, Y. J., and Larochelle, H. (2015). A neural autoregressive approach to attention-based recognition. *Int. J. Comput. Vis.*, 113(1):67–79.

Zhou, Y., Li, Z., Xiao, S., He, C., Huang, Z., and Li, H. (2018). Auto-conditioned recurrent networks for extended complex human motion synthesis. In *ICLR*.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X. (2016). Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.