Integrating Eye Gaze in Human Motion Prediction

Hongyi Chen
hongyic@andrew.cmu.edu
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

Human-robot collaborations have been recognized as an essential component for future factories. It remains challenging to properly design the behavior of those co-robots. The first step of those robots is predicting the humans movement in order to safely plan their own motion trajectories and efficiently collaborate with humans. Recent papers predict future trajectories based on past trajectories using learning-based methods, however, they ignore that people's natural, nonverbal behavior like eye gaze data can provide additional insight into their goals and concerns about the task. This paper adopts human eye gaze based intention recognition approach and integrate this estimated intended target into semi-adaptable neural networks to predict human movement in longer-timescale. The experiments validate the effectiveness of human eye gaze in long-timescale movement prediction, achieving 12.1% accuracy improvement compared to the baseline model without eye gaze.

1 INTRODUCTION

Smooth interactions among intelligent entities depend on a clear understanding of what the others would do in various circumstances. For example, soccer players predict the motions of their teammates for better cooperation; pedestrians have a notion of where others are going so as to avoid collisions. In modern factories, human workers and robots are two major workforces, automotive manufacturers such as Volkswagen and BMW introduced human-robot cooperation in final assembly lines in 2013 [25]. These robots that interact in proximity with humans are required to know what the human is going to do in the near future. The benefits are that, based on the predictions, robots can plan collision-free trajectories to assure human's safety and schedule their actions in advance to improve task efficiency. However, It remains challenging to recognize human intention and predict human movement due to the nonlinearity and stochasticity in the human behavior [19]. In addition, individual differences are also prominent. Prediction models that work for one person may not be applicable to another.

Nowadays, advancements in deep learning have enabled the creation of data-driven neural network models that can learn complex features given sufficient data. These have enabled various applications ranging from temporal forecasting to motion prediction [24]. Researchers also adopted neural networks to predict human behavior and intention in human-robot interaction tasks and got promising results [8][2][15]. However, human's natural, nonverbal behaviors, which can provide insightful information about the concerns in tasks, have remained largely unadopted in these models.

In fact, people found that human's visual behaviour is intrinsically linked to how we plan and execute actions. Humans' eyes are constantly scanning and interrogating the environment around us in a continuous cycle of observation and prediction [7]. As a consequence, by monitoring the eye movement behaviour of others, we are able to infer their future movement. An understanding of this behaviour can provide us with opportunities in both cooperative and noncooperative settings. For example, a poker player observing which cards their opponents have been gazing upon to potentially determine which hand they are trying to compile; or an assistant handing a surgeon a tool that they will potentially use next. Collaborative robots may be able to improve their interactions with humans by anticipating the human's intentions by combining the human eye gaze and ontic behaviours.

Based on these discoveries, We think it's likely that incorporating visual behaviour to form *a prior* probability of these intentions can improve prediction accuracy of human motion. Thus, to confirm my hypothesise, We adopt an intention recognition approach that estimate intended target from human eye gaze and demonstrate how it substantially improves the human motion prediction performance. The main contribution through this work is that by introducing eye gaze information into the prediction model, the accuracy of the long-timescale prediction model is significantly improved compared to the baseline model without eye gaze information.

The remainder of the paper is organized as follows. Section 2 formulates the human motion prediction problem. Section 3 briefly discusses the previous works in the human motion prediction problem and human eye gaze pattern. Section 4 shows the methodology of the semi-adaptable neural networks We used in this paper. Section 5 demonstrates how we analyze eye gaze and recognize human intention. Section 6 represents the experiment results. Section 7 points out the challenges and problems We am facing currently and proposes my next step plan. Section 8 concludes the paper.

2 PROBLEM FORMULATION

Predicting human motion is important for smooth human robot interaction, because first, if the robot knows what the human is going to do, it can adapt its actions to collaborate with humans in an efficient way, and second, plan collisionfree trajectories to guarantee humans safety [15]. This paper concerns the prediction of one human joint (e.g., wrist), which is reasonable because when a human works in close proximity to a robot, special attention should be paid to the movement of human's hand. Moreover, one joint motion prediction is extendable to that of multiple joints, which can be found in paper [5].

The transition model of human joint motion is formulated as

$$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), i) + w_k \tag{1}$$

where $\boldsymbol{x}(k+1) \in \mathbb{R}^{3M}$ denotes human's M-step positions of the joint at future time steps k+1, k+2, ..., k+M in a Cartesian coordinate system. $M \in \mathbb{N}$ is the prediction horizon. Denoting the Cartesian

position of the joint at time step k by $p(k) \in \mathbb{R}^3$, $\boldsymbol{x}(k+1)$ is obtained by stacking $p(k+1), p(k+2), ..., p(k+M)$. $\boldsymbol{x}(k) \in \mathbb{R}^{3N}$ denotes human's past N-step positions of the joint. It is also constructed by stacking the position vectors $p(k), p(k-1), ..., p(k-N+1)$. $i \in \mathbb{R}^3$ is the position of the next possible intended target, and this position is obtained by the eye gaze intention recognition module which we will discuss in section 5. $w_k \in \mathbb{R}^{3M}$ is a zero-mean white Gaussian noise. The function $f(x(k), i) : \mathbb{R}^{3N} \times \mathbb{R}^3 \rightarrow \mathrm{R}^{3M}$ represents the transition of the human motion, which takes historical trajectory and current action label as inputs, and outputs the future positions of the joint.

Since human behavior differs greatly across individuals and is highly time-varying, function f may not be a time invariant function. Though f takes the continuous parameter intention i as one of the inputs to accommodate some of the variances, an adaptable model of f is still desired to account for continuous changes online in order to provide accurate prediction.

3 RELATED WORK

3.1 Human motion prediction

Early attempts have been made to predict human motion using kalman filter and particle filter [5] [4], where the problem is posed as a tracking problem. Another category of approaches assumes that human is rational with respect to certain cost functions. Human trajectories can then be predicted by optimizing the cost function [13]. The difficulty of this method is that the cost functions of human are hard to obtain due to stochasticity and complexity in human intention. Another domain of work prominently focuses on latent variable based probabilistic models. Wu et al. [26] use hidden markov models (HMMs) combined with multilayer perceptrons to model the evolution patterns of motion trajectory.

Similar to HMMs, recurrent neural networks (RNNs) have distributed hidden states to store information about the past, and many works on RNNs have obtained big success on human motion prediction [8][2], but they still suffer from several problems. First problem is that RNNs are hard to train. Heroic efforts of many years still fail to accelerate the training speed of RNNs. Second problem is that predictions from RNNs are deterministic, which is not satisfactory in human robot interaction, since the robot needs the uncertainty level of human's future motion for safe motion planning. The last serious problem is that the RNN models are fixed and they cannot adapt to time-varying human behaviors.

Besides, previous papers aim to solve these problems by proposing a semi-adaptable neural network[15]. To be specific, a neural network is trained offline to represent the human motion transition model, and then recursive least square parameter adaptation algorithm (RLS-PAA) is adopted for online parameter adaptation of the last layer in the neural network and for uncertainty estimation. The proposed method advantages human motion prediction in three aspects. First, it is computationally more efficient to use a feedforward neural network than to use a RNN for approximation of the human transition model. In the meanwhile, the mechanism for adaptable feedforward neural networks is equally applicable to adaptable RNNs. Second, it only updates the top levels of neural networks which accelerate the process training, which is crucial for online learning. This only helps to finetune the networks for each individual. Third, it computes the uncertainty level of the predictions, which is important for safe motion planning of robots. Also the experiments results prove that semi-adaptable neural network outperform previous methods like identifier-based algorithm. That's why We will use this network as my baseline model and integrate eye gaze into it.

3.2 Eye gaze for intention recognition

Since eye gaze behavior is so closely tied to people's goals, it has been widely studied as a modality for understanding people's mental state in a variety of applications. Matsuzaka et al. [17] showed that people's gaze predicts their intended grasp object and strategy (one- or two-handed) in a VR manipulation task. Huang and Mutlu [11] uses hand-crafted features to predict a person's food order from a manipulator robot in a human-robot interaction study. Admoni's paper [1] gave a more concrete review and analysis of eye gaze in human-robot interaction, in the end, they asked how can eye gaze be integrated with other social behaviors. For this question, Singh et al. proposed using a Bayesian model of human intention recognition that combines gaze and model-based approaches for online human intention recognition [23]. Gaze data is used to build probability distributions over a set of possible intentions, which are then used as priors in a model-based intention recognition algorithm. Duarte et al. [6] shows that people can follow gaze cues when seeing other people perform an object manipulation task, and their understanding persists even when a robot is giving gaze cues. You can see Reuben's paper [22] for a full more relevant papers about eye gaze pattern during manipulation.

4 SEMI-DAPTABLE NEURAL NETWORKS

Since human's motion is not only time-varying but also highly nonlinear, we propose to use a neural network to construct the model f, bacause neural networks have good model capacity. To make it adaptable, notice that if we remove the last layer, the pretrained neural network becomes an effective feature extractor [3], the features from which are better than handcrafted ones [21]. Therefore we only adapt the weights of output layer of the neural network online, which fixes the weights of the remaining layers, hence fixing the extracted features.

The proposed procedure is that

1) We first design a neural network architecture;

2) We train the model f offline;

3) During online execution, we adapt the parameters of the last layer of the neural network using efficient adaptation algorithm;

4) We then compute the uncertainty level of predictions given the adaptation result.

The algorithm is shown in Algorithm 1.

4.1 Training the Neural Network

To train the transition model f, we choose an n-layer neural network with ReLU activation function which takes the positive part of the input to a neuron.

$$f(\boldsymbol{x}(k), i) = W^T max(0, g(U, s_k)) + \epsilon(s_k) \tag{2}$$

where $s_k = [\boldsymbol{x}(k)^T, i, 1]^T \in \mathbb{R}^{3N+2}$ is the input vector, g denotes $(n-1)$ - layer neural network, whose weights are packed in U.

Algorithm 1: Semi-adaptable neural network for human motion prediction

Input : Offline trained neural network (2) with g, U and W
Output : future trajectory $\mathbf{x}(k+1)$
Variables : Adaptation gain F, *a priori* mean squared estimation error of states $X_{\tilde{x}\tilde{x}}$, mean squared estimation error of the parameters $X_{\tilde{\theta}\tilde{\theta}}$, neural network last layer parameters θ, estimated rate of change $\delta\theta$ (approximation of $\Delta\theta$), variance of zero-mean white Gaussian noise $Var(w_k)$
Initialization: $F = 1000\mathcal{I}$, θ = column stack of W,$X_{\tilde{x}\tilde{x}} = \mathbf{0}$, $X_{\tilde{\theta}\tilde{\theta}} = \mathbf{0}$, $\lambda_1 = 0.998$, $\lambda_2 = 1$

while *True* **do**
  Wait for a new joint position p captured by Kinect and current action label a from action recognition module;
  Construct $s_k = [p(k), p(k-1), ..., p(k-N+1), a, 1]^T$;
  Obtain $\Phi(k)$ by diagonal concatenation of $max(0, g(U, s_k))$;
  Update F by (7);
  Adapt the parameters θ in last layer of neural network by (6);
  Calculate future joint trajectory $x(k+1)$ by (3);
  Update $\delta\theta$ and calculate $X_{\tilde{x}\tilde{x}}$ and $X_{\tilde{\theta}\tilde{\theta}}$ by (8) and (11);
  send $x(k+1)$ and $X_{\tilde{x}\tilde{x}}$ to robot control.
end

$\epsilon(s_k) \in \mathbb{R}^{3M}$ is the function reconstruction error, which goes to zero when the neural network is fully trained. $W \in \mathbb{R}^{n_h \times 3M}$ is the weights of the last layer, where $n_h \in \mathbb{N}$ is the number of neurons in the hidden layer of the neural network [20].

4.2 Parameter Adaptation Algorithm

To accommodate both the time varying behavior of human and individual differences among different people, it is important to update the parameters online. In this paper, we applied the recursive least square parameter adaptation algorithm (RLS-PAA) with forgetting factor [9] to asymptotically adapt the parameters in the neural network.

By stacking all the column vectors of W, we get a time varying vector $\in \mathbb{R}^{3Mn_h}$ to represent the weights of last layer. θ_k denotes its value at time step k. To represent the extracted features, we define a new data matrix $\phi_k \in \mathbb{R}^{3M \times 3Mn_h}$ as a diagonal concatenation of M pieces of $max(0, g(U, s_k))$. Using ϕ_k and θ_k, (1) and (2) can be written as

$$\boldsymbol{x}(k+1) = \phi_k \theta_k + w_k \tag{3}$$

Let $\hat{\theta}_k$ denotes the parameter estimate at time step k, and let $\widetilde{\theta}_k = \theta_k - \hat{\theta}_k$ be the parameter estimation error. We define the *a priori* estimate of the state and the estimation error as:

$$\hat{\boldsymbol{x}}(k+1|k) = \phi_k \hat{\theta}_k \tag{4}$$

$$\widetilde{\boldsymbol{x}}(k+1|k) = \phi_k \widetilde{\theta}_k + w_k \tag{5}$$

The core idea of RLS-PAA is to iteratively update the parameter estimation $\hat{\theta}_k$ and predict $\boldsymbol{x}(k+1)$ when new measurements become available. The parameter update rule of RLS-PAA can be summarized as:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + F_k {\phi_k}^T \widetilde{\boldsymbol{x}}(k+1|k) \tag{6}$$

where F_k is the learning gain updated by:

$$F_{k+1} = \frac{1}{\lambda_1(k)} \left[F_k - \lambda_2(k) \frac{F_k \phi_k {\phi_k}^T F_k}{\lambda_1(k) + \lambda_2(k) {\phi_k}^T F_k \phi_k} \right] \tag{7}$$

where $0 < \lambda_1(k) \leq 1$ and $0 < \lambda_2(k) \leq 2$. Typical choices for $\lambda_1(k)$ and $\lambda_2(k)$ are:

- $\lambda_1(k) = 1$ and $\lambda_2(k) = 1$ for standard typical least squares gain.
- $0 < \lambda_1(k) < 1$ and $\lambda_2(k) = 1$ for least squares gain with forgetting factor.
- $\lambda_1(k) = 1$ and $\lambda_2(k) = 0$ for constant adaptation gain.

4.3 Mean Squared Estimation Error Propagation

To guarantee safety, the uncertainty of the prediction is also quantified during online adaptation [16].

4.3.1 State estimation. : Note that $\hat{\theta}_k$ only contains information up to the $(k-1)$th time step, and $\widetilde{\theta}_k$ is independent of w_k. Thus the *a priori* mean squared estimation error (MSEE) $\boldsymbol{X}_{\widetilde{x}\widetilde{x}}(k+1|k) = \mathbb{E}\left[\widetilde{\boldsymbol{x}}(k+1|k)\widetilde{\boldsymbol{x}}(k+1|k)^T\right]$ is

$$\boldsymbol{X}_{\widetilde{x}\widetilde{x}}(k+1|k) = \phi_k \boldsymbol{X}_{\widetilde{\theta}\widetilde{\theta}}(k) {\phi_k}^T + Var(w_k) \tag{8}$$

where $\boldsymbol{X}_{\widetilde{\theta}\widetilde{\theta}}(k) = \mathbb{E}\left[\widetilde{\theta}_k \widetilde{\theta}_k^T\right]$ is the mean squared error of the parameter estimate and $Var(w_k)$ is the variance of zero mean white Gaussian noise.

4.3.2 Parameter estimation. : Since the system is time varying, $\triangle\theta_k = \theta_{k+1} - \theta_k \neq 0$. According to parameter estimation algorithm in (6), the parameter estimation error is

$$\widetilde{\theta}_{k+1} = \widetilde{\theta}_k - F_k {\phi_k}^T \widetilde{\boldsymbol{x}}(k+1|k) + \triangle\theta_k \tag{9}$$

The estimated parameter is biased and the expectation of the error can be expressed as

$$\mathbb{E}(\widetilde{\theta}_{k+1}) = \left[We - F_k {\phi_k}^T \phi_k \right] \mathbb{E}(\widetilde{\theta}_k) + \triangle\theta_k \tag{10}$$

The mean squared error of parameter estimate follows from (9) and (10):

$$\begin{aligned} \boldsymbol{X}_{\widetilde{\theta}\widetilde{\theta}}(k+1) =& F_k {\phi_k}^T \boldsymbol{X}_{\widetilde{x}\widetilde{x}}(k+1|k) \phi_k F_k - \boldsymbol{X}_{\widetilde{\theta}\widetilde{\theta}}(k) {\phi_k}^T \phi_k F_k \\ & - F_k {\phi_k}^T \phi_k \boldsymbol{X}_{\widetilde{\theta}\widetilde{\theta}}(k) + \mathbb{E}\left[\widetilde{\theta}_{k+1}\right] \triangle {\theta_k}^T \\ & + \triangle\theta_k \mathbb{E}\left[\widetilde{\theta}_{k+1}\right]^T - \triangle\theta_k \triangle {\theta_k}^T + \boldsymbol{X}_{\widetilde{\theta}\widetilde{\theta}}(k) \end{aligned} \tag{11}$$

Since $\triangle\theta_k$ is unknown in (10) and (11), we define $d\theta_k = \hat{\theta}_k - \hat{\theta}_{k-1}$, and approximate $\triangle\theta_k$ as $\delta\theta_k$ which is the average of $d\theta_i, i = k - n_w + 1, k - n_w, ..., k$, where $n_w \in \mathbb{N}$ is the window size.

At step k, the predicted trajectory $\boldsymbol{x}(k \hat{+} 1|k)$ together with the uncertainty matrix $\boldsymbol{X}_{\widetilde{xx}}(k+1|k)$ is then sent to robot control to generate the safety constraint.

5 EYE GAZE INTENTION ANALYSIS

We adopt the eye gaze intention analysis from Aronson's thesis [22]. In his methodology, We realize that we can take advantage of some of the physiological characteristics of human gaze to simplify the process of eye gaze data. Rather than having free control over eye gaze direction, people follow consistent eye gaze patterns, which consist primarily of fixations (500 - 2000ms stationary periods) separated by saccades (rapid 100 - 500 ms ballistic trajectories between fixation locations). This kind of pattern makes the raw eye gaze signal being unstable and noisy, thus we need to process the raw eye gaze data first before we integrate it into semi-adaptable neural networks.

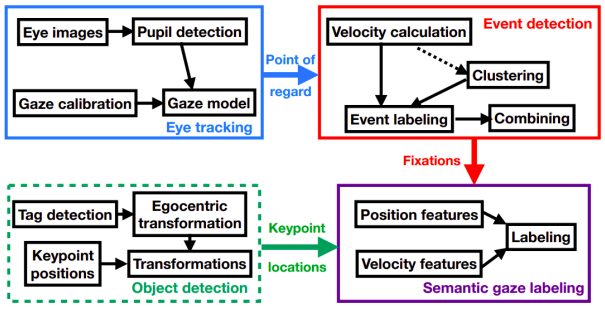

Figure 1: Flow chart of the gaze analysis pipeline. Raw eye gaze is subdivided into fixations and then labeled with a keypoint supplied by an (external) object tracking system.

The standard eye gaze processing pipeline proceeds through several steps (see Figure 1). First, the raw eye gaze data is collected using an off-the-shelf eye tracker. Next, the eye gaze data is segmented into fixations, corresponding to physiological principles of eye gaze and easing the classification problem. Then, each individual fixation is labeled with the most likely object in the workspace that it corresponds to. Finally, the timed sequences of foveated objects are analyzed according to the needs of our assistance procedure. Since the activity recognition during manipulation is heavily dependent on which objects the user looks at. Therefore, mapping this raw gaze signal to semantic object labels can provide better results [19, 20]. In my current work, We did event detection to extract the fixation sequence from the raw eye gaze sequence, and plan to do the semantic gaze labeling in the next step.

There are two traditional algorithms for event detection: dispersion thresholding (I-DT) and velocity thresholding (I-VT) [77]. Both depend on the fact that the point-to-point velocity during saccades are generally much larger than during fixations. In I-DT, a measure of dispersion (e.g. variance) is calculated over windows of the eye gaze signal. Windows less than a manually-chosen value are determined to be fixations, while windows above the value are labeled saccades. In I-VT, the point-to-point velocity of the signal (the numerical derivative) is computed, and each point is labeled a fixation or saccade if it is below or above a custom threshold. Then, successive fixation labels that exceed a minimum fixation time are fused together and determined to be a single fixation.

For this work, We adopt a variant of I-VT, known as I-BMM [78, 79]. This method similarly calculates the velocity of the eye gaze signal (by angle), but learns a dynamic threshold by fitting a 2-component Gaussian mixture model to a sample of eye gaze data. Then, adjacent fixation labels are fused, and fusions that exceed a specified minimum time are labeled as fixations. And the position of fixation is taken as the position of next possible intended target position. You can find the a python implementation that works both offline and online written by Aronson in github link https://github.com/HARPLab/ibmmpy.

6 RESULTS

6.1 DATASET

In order to verify the proposed human motion prediction approach, We first did experiment in HARMONIC dataset [18], a large multimodal dataset of human interactions in a shared autonomy setting. This dataset is collected from a robot assisitve eating task. In this task, participants teleoperate a robot to pick up one of three morsels on a plate in front of them (see Figure 2). The dataset provides human, robot, and environment data streams from twenty-four people engaged in an assistive eating task with a 6 degree-of-freedom (DOF) robot arm. From each participant, it recorded video of both eyes in real-world position, joystick commands, electromyography from the participant's forearm used to operate the joystick, third person stereo video and the joint positions of the 6 DOF robot arm. We only use the eye gazes positions and wrist joint positions in real-world. Even though this dataset contains all the data We need for experiment, it is not quiet suitable for our proposed algorithms as We will discuss in detail in section 7.

6.2 Network Training and Online Adaptation

We use a 3-layer neural network with 40 nodes in the hidden layer. The number of nodes in the input layer and the output layer is varied from 15 to 150 in different models. The loss function is set to be mean squared error loss. The optimizer is adam optimizer and the number of epochs is 1000. We train the models in 80% data samples of the entire dataset and test the models in remaining 20% data samples.

After obtaining the neural network model for motion transition, We use RLS-PAA to adapt the weights of the last layer. Assume the number of nodes in the last layer is k, and each node has 41 parameters, of which 40 correspond to the outputs from the hidden layer, and the remaining one parameter is a bias term. In total, there are $41k$ parameters to be adapted online.

6.3 Models Results

In previous semi-adaptable neural networks predictions, they only predict for next three to nine data points, which is less than 0.1 second. To validate the effectiveness of eye gaze in different timescales,

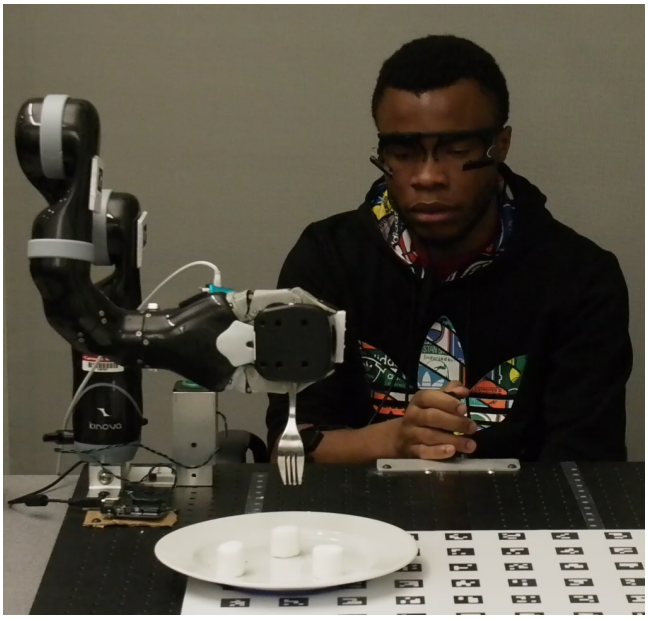

Figure 2: Robot assistive eating task.

We tested on four different models described as follows: (the time interval between each data point is close to 0.01s)

- short predict model: predict next 5 data points based on past 5 data points. Thus the number of nodes in the input layer and output layer is $3 \times 5 = 15$.
- median predict model: predict next 10 data points based on past 10 data points. Thus the number of nodes in the input layer and output layer is $3 \times 10 = 30$.
- long predict model: predict next 20 data points based on past 20 data points. Thus the number of nodes in the input layer and output layer is $3 \times 20 = 60$.
- super long predict model: predict next 50 data points based on past 20 data points. Thus the number of nodes in the input layer is $3 \times 20 = 60$ and the number of nodes in the output layer is $3 \times 50 = 150$.

The testing results of different models with and without eye gaze is shown in Table 1, the metric of testing results is mean squared estimation error (MSEE cm^2)

We can easily see that the effectiveness of eye gaze in prediction increases as we extend the timescale of model prediction. We think this can be explained by the previous psychology experiments about eye gaze behavior. People will look at relevant locations before interacting with them, the advance in time could be varied in different tasks and situation, but the normal time advance is ranged from few hundreds of milliseconds to a few seconds. As a consequence, eye gaze gradually becomes crucial as the timescales of models grow from few milliseconds to 150 milliseconds.

On the other hand, a few milliseconds would be too short for human to spot an object, thus it's likely that eye gaze signal during that time period is a rapid and noisy saccades signal. Also, since We fused adjacent eye gaze signal into fixation labels, it's possible that this fused fixation signal is misleading in short time period. That's why we see a huge negative effect of eye gaze in short model prediction.

7 REFLECTION

In this section, We want to discuss the problems We discover when testing in HAROMIC dataset and why that's a problem. To better test the effectiveness of eye gaze in prediction, what should we do next.

7.1 Problem in dataset

We notice that in video people spend a lot of time looking at the end-effector of the robot. Unlike in by-hand manipulation, people look at the end-effector of the robot throughout the trial. Specifically, 68.1 ± 2.1% of the fixations during each trial were at the end-effector or tool [22]. Presumably, this gaze difference is due to people needing visual feedback to determine the location of the robot end-effector, whereas during by-hand manipulation, people can use their own proprioception to determine their hand position. Besides, it's pretty common that people held the robot stationary and alternated their focus between the end-effector of the robot and their goal object, and monitoring glances, in which people moved the robot while looking back and forth between it and their goal position. These patterns can also give confusing fixation sequence and mislead the predictions. At last, in situations when people encounter problems during teleoperation, eye gaze pattern will change and provide useless signal. For example, people often moved their heads significantly more than usual in order to get a better view when robot blocks occlude their view of the target morsel; people will look at the joint that is causing them an issue when the robot goes into a problematic kinematic configuration. All these reasons extend the time period between we look at the intended object and we teleoperate the robot to pick it up and thus make the normal eye gaze based prediction method become less helpful.

In contrast, in previous study designed by Johansson [12] where users were instructed to grasp an object, manipulate it around an obstacle, and place it down elsewhere, people followed consistent eye gaze patterns. They looked at relevant locations before interacting with them: people look at the object until just before they grasp it, the obstacle until just before navigating around it, and the placement location just before placing the object. In addition, the paper found that people rarely looked at their own hands and that their gaze was almost entirely directed towards task-relevant locations. Similar results were found with people performing natural tasks like making tea [14] and making a sandwich [10]. My proposed approach is more suitable for the human performed natural tasks, where eye gaze can provide insights about human concerns in short future,instead of human teleoperate robot tasks.

7.2 Future work

Because of the problems We mentioned above, We decided to collect my own dataset for testing. One ideal testing scenario is the human-robot collaborative desktop assembly task as illustrated in Fig 3. Humans have two plans in mind: inserting the RAMs in the motherboard first and then assembling the disk to the desktop case, or assembling the disk to the desktop case first and then inserting

Table 1: Experiments of eye gaze in different time-scale models (results metric MSEE cm^2)

Models	Data input (w/wo eye gaze)	Data output	Results wo eye gaze	Results w eye gaze	Improvements
short model	past 0.05s movement	next 0.05s movement	49.92	64.34	-28.9%
median model	past 0.1s movement	next 0.1s movement	187.10	184.49	1.2%
long model	past 0.2s movement	next 0.2s movement	782.67	761.83	2.6%
super long mode	past 0.2s movement	next 0.5s movementl	1242.38	1091.68	12.1%

the RAM to the motherboard. The robot may collaborate with the human by handing the other RAM to the human if the human is assembling a RAM, or bringing the screwdriver to the human if the human is assembling the disk [15]. In my plan, the robot only needs to recognize the intention of humans and predict their arm trajectory for now. If we can get pretty accurate prediction results, then the robot could help hand tools to humans in later experiments. For specific hardware equipment, We plan to use one Kinect sensor to monitor the dynamic environment and two Ensenso cameras to capture the static components placed in the workspace. For simplicity, the desktop case and the helmet are attached to markers so that Kinect can directly retrieve their location in real time.

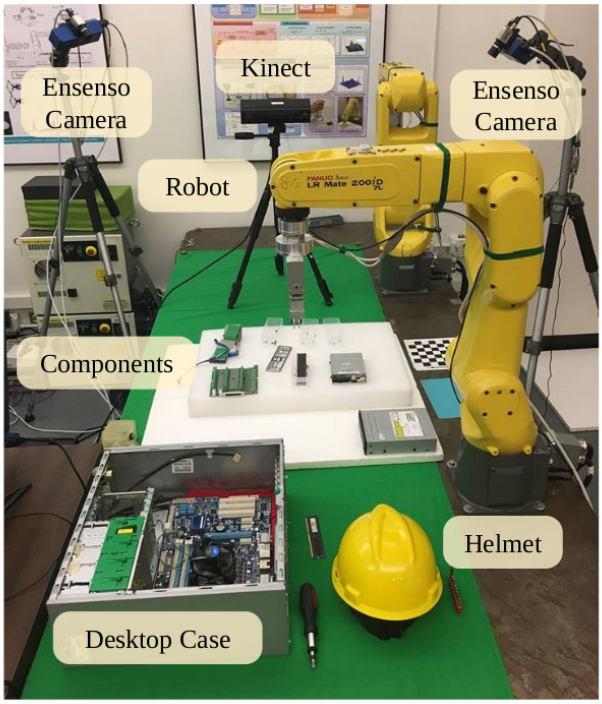

Figure 3: Experiment setting.

With decreasing cost and increasing robustness, eye trackers are entering the consumer market, is it most appropriate to use the Pupil Core mobile eye tracker (see Figure 4) for our application. This system consists of a glasses-like frame worn by the user, on which a number of cameras are mounted. One or two IR cameras are mounted above the users' eyes (corresponding to a monocular or binocular setup) and record high-frequency video of the eyes themselves. In addition, a forward-mounted ("egocentric" or "world") camera captures the scene from the point of view of the user. We can use the Pupil Labs Pupil tracker with their built-in pupil detection and then map the pupil center points to the gaze point in the world camera.

Figure 4: Pupil Core binocular eye tracker.

After setting the experiment platform and collecting the needed data, we can test again to see the effectiveness of eye gaze in prediction.

8 CONCLUSION

This project integrates human eye gaze into semi-adaptable neural networks to predict human motion, which not only improve the prediction accuracy but also enable the model to do longer-timescale prediction. We built the semi-adaptable neural networks, which can accommodate individual differences and time-varying behaviors, for predictions. Then, We fuse the raw eye gaze sequence to get the fixation sequence, which helps to get rid of the noise and saccades, and integrate it into the networks. The experiments results in HARMONIC dataset demonstrate the effectiveness of eye gaze in prediction, achieving 12.1% accuracy improvement in long-timescale movement prediction.

REFERENCES

[1] Henny Admoni and Brian Scassellati. Social eye gaze in human-robot interaction: A review, May 2017.
[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces, 06 2016.
[3] Ben Athiwaratkun and Keegan Kang. Feature representation in convolutional neural networks, 07 2015.
[4] Frank Broz and Geoffrey Gordon. Better motion prediction for people-tracking, 03 2004.

[5] Yujiao Cheng, Weiye Zhao, Changliu Liu, and Masayoshi Tomizuka. Human motion prediction using semi-adaptable neural networks. pages 4884–4890, 07 2019.

[6] Nuno Duarte, Jovica Tasevski, Moreno Coco, Mirko Raković, Aude Billard, and José Santos-Victor. Action Anticipation: Reading the Intentions of Humans and Robots. *arXiv e-prints*, page arXiv:1802.02788, February 2018.

[7] Tom Foulsham. Eye movements and their functions in everyday tasks, 11 2014.

[8] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions, 10 2017.

[9] Graham Goodwin and Kwai Sin. Adaptive filtering prediction and control, 01 1984.

[10] Mary Hayhoe, Anurag Shrivastava, Ryan Mruczek, and Jeff Pelz. Visual memory and motor planning in a natural task [abstract], 02 2003.

[11] Chien-Ming Huang and Bilge Mutlu. Anticipatory robot control for efficient human-robot collaboration. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, HRI '16, page 83–90. IEEE Press, 2016.

[12] Roland Johansson, Göran Westling, Anders Bäckström, and John Flanagan. Eye–hand coordination in object manipulation, 10 2001.

[13] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning, 05 2011.

[14] M. Land, Neil Mennie, and Jennifer Rusted. The roles of vision and eye movements in the control of activities of daily life, 02 1999.

[15] Changliu Liu, Te Tang, Hsien-Chung Lin, Yujiao Cheng, and Masayoshi Tomizuka. Serocs: Safe and efficient robot collaborative systems for next generation intelligent industrial co-robots, 09 2018.

[16] Changliu Liu and Masayoshi Tomizuka. Safe exploration: Addressing various uncertainty levels in human robot interactions, 07 2015.

[17] A. Matsuzaka, L. Yang, C. Guo, T. Shirato, and A. Namiki. Assistance for master-slave system for objects of various shapes by eye gaze tracking and motion prediction. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1953–1958, 2018.

[18] Benjamin A. Newman, Reuben M. Aronson, Siddhartha S. Srinivasa, Kris Kitani, and Henny Admoni. HARMONIC: A multimodal dataset of assistive human-robot collaboration. *CoRR*, abs/1807.11154, 2018.

[19] Zhen Peng, Tim Genewein, and Daniel Braun. Assessing randomness and complexity in human motion trajectories through analysis of symbolic sequences. *Frontiers in human neuroscience*, 8:168, 03 2014.

[20] Harish Ravichandar and Ashwin Dani. Human intention inference using expectation-maximization algorithm with online model learning, 09 2016.

[21] Ali Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition, 03 2014.

[22] Aronson Reuben. Eye gaze for assistive manipulation, 2020.

[23] Ronal Singh, Tim Miller, Joshua Newn, Liz Sonenberg, Eduardo Velloso, and Frank Vetere. Combining planning with gaze for online human intention recognition, 07 2018.

[24] Michael Suguitan, Randy Gomez, and Guy Hoffman. Demonstrating moveae: Modifying affective robot movements using classifying variational autoencoders. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '20, page 78, New York, NY, USA, 2020. Association for Computing Machinery.

[25] Christian Weckenborg, Karsten Kieckhäfer, Christoph Müller, Martin Grunewald, and Thomas S. Spengler. Balancing of assembly lines with collaborative robots. *Business Research*, 13(1):93–132, April 2020.

[26] Di Wu and Ling Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition, 06 2014.