Anticipatory Control on Human-Following Robots Using Online Deep Model Predictive Control

Shun Gui and Yan Luximon

***Abstract*—Mobile robots face challenges when collaborating with humans in crowded and occluded environments. To tackle this issue, we propose a solution called online deep model predictive control (Deep-MPC) and apply it to human-following robots. Deep-MPC incorporates a 3D human detector, an online learning transition model, and a data-driven MPC framework. Specifically, the 3D human detector generates the target's 3D bounding box, while the transition model predicts future states, enabling anticipatory control. By combining the 3D bounding box's intersection over union (IoU) and state anticipation, we propose a novel evaluation metric that enhances the following robustness. The data-driven MPC framework optimizes robot actions using the neural network of the transition model, and online learning occurs through autonomous interaction with the environment, eliminating the need for system modeling and controller design. To validate our method, we conducted extensive real-world human-following experiments, demonstrating its superior performance compared to some existing methods, skeleton-based methods and approaches without Deep-MPC.**

***Index Terms*—Anticipatory Control, Model Predictive Control, Human-following Robot, Online Learning**

I. Introduction

MOBILE robots have gained significant importance in various industrial and service fields, with diverse applications, including industrial mobile robots and service robots [1]. In certain scenarios, mobile robots need to collaborate with humans to transport goods, such as following human trajectories to assist with material handling in factory workshops or logistics warehouses, or following passengers to transport luggage in airports. In these scenarios, continuous human-following is essential for mobile robots [2]–[4]. Currently, there are many studies on human-following robot. Leigh et al. [5] utilized a 2D laser scanner to achieve human following both indoors and outdoors. Ferrer et al. [6] introduced a new framework for socially-aware human-following, which enables robots to walk alongside humans in urban environments. Lee et al. [2] integrated deep learning methods and variational Bayesian techniques to achieve robust human-following for home service robots. Toan et al. [7] proposed a leg detector to achieve human-following in mixed environments. However, few studies conducted in-depth investigations of scenarios with multiple people occlusions. Siva et al. [8] extracted three features, namely histogram of oriented gradients features, GIST features, and local difference binary patterns features, to detect humans for human-robot robots in scenarios where there are significant changes, such as entering-exiting a mine.

In scenarios with multiple people occlusions, human-following robots are prone to losing following of the target or misidentifying other individuals as the target. Therefore, in occlusion-rich environments, it is crucial to define the criteria for robot following failure, that is, the conditions under which the robot should determine the failure and re-identify the target. In many existing studies [2], [9] on human-following robots, following failure is commonly defined as losing the target without providing quantifiable criteria. This may be due to the fact that most research does not consider scenarios with multiple people occlusions. Furthermore, in some studies on human tracking, trackers, including Nearest-neighbor tracker [10], and multi-hypothesis tracker [11] were employed for tracking targets. The CLEAR-MOT metric [12] was commonly used to evaluate the performance of these trackers, but in human-following robots, the relative position between the followed target and the robot is crucial, which these trackers typically do not consider. Our study focuses on scenes with multiple people occlusions through formulating human-following metrics, which better reflects practical applications.

In order to enhance the robustness and reduce latency of human-robot following, some research has focused on anticipatory control [13]. For instance, Wang et al. [14] developed a neural network that consumes skeleton data to predict the target's skeleton state half a second in advance, and the robot performs the following action accordingly, leading to improved following robustness. Chen et al. [15] developed a new algorithm to predict the human's future position based on the current position and orientation using a human-walking model, to achieve smoother and faster human-following. However, these studies only predicted one future state and did not test the approach in occluded scenes. To further improve the robustness of human-following in occluded scenes, we

Manuscript received 13 October, 2023; revised 05 May, 2024; accepted 11 June, 2024. This research is partially funded by the Research Grants Council (Project No. GRF/PolyU 15606321), the Laboratory for Artificial Intelligence in Design (Project Code: RP1-3) and the Hong Kong Polytechnic University (Colour, Imaging, and Metaverse Research Centre, P0050655) of Hong Kong Special Administrative Region, China.

The authors are with the School of Design, The Hong Kong Polytechnic University, Hong Kong. shun.gui@connect.polyu.hk; yan.luximon@polyu.edu.hk. Corresponding author: Yan Luximon.

The ethical approval has been obtained from the Departmental Research Committee, the Reference Number: HSEARS20211015005.

adopted a model predictive control (MPC) framework to predict multiple future states of the target for optimizing current actions. As our method possesses the prediction capability, we proposed a novel human-following metric combining 3D IoU and state anticipation, which demonstrates better performance in experiments. Furthermore, there is a limited amount of research that specifically focuses on the occlusion challenges in the context of human-following robot tasks.

Several studies have explored the integration of deep learning and Model Predictive Control (MPC) methods. One of the most relevant categories of research to our study is learning the dynamics of a system using neural networks [16]–[18]. Hansen et al. [19] proposed the TD-MPC method, which utilizes neural networks to learn the system's dynamics and combines model-based and model-free reinforcement learning approaches to achieve optimal control strategies. [17], [18] leveraged cross entropy method [20] or reinforcement learning algorithm to optimize the actions based on the neural network-based dynamic. However, the utilization of fully online neural network dynamics with gradient backpropagation in real robots is rarely addressed in existing research. Our approach, on the other hand, is particularly well-suited for real-world robot tasks. Furthermore, Lucia et al. [21] utilized deep neural networks (DNNs) to offline approximate the optimal control policy for MPC problems, subsequently employing this approximation for online control. Similarly, Karg et al. [22] leveraged DNNs to approximate the optimal control law of MPC. In these studies, neural networks were used to replace MPC to achieve real-time control. Farshidian et al. [23] introduced a Deep Model Predictive Control (DMPC) approach, where an MPC policy was utilized as an actor to interact with the environment, collecting data for training a critic model represented by a neural network. In a similar vein, Elnour et al. [24] employed a neural network to learn the dynamics of sports facilities and subsequently applied MPC methods to address the corresponding problem. Lenz et al. [25] also proposed a variant of DeepMPC; however, it required manual data collection and offline training of the dynamics model. In contrast, our method achieves fully online learning and control without any manual intervention, rendering it well-suited for real-world applications.

With the advancement of robot learning technology, many studies have emerged that focus on improving robot behavior through interaction with the environment [26], for tasks such as robot manipulation [27], navigation [28], and quadruped locomotion [29]. Some studies [19] also employed MPC to achieve task learning, but their performance was only tested in simulation tasks. Enabling robots to learn how to accomplish tasks in the real world is a crucial aspect of robotics research. In this context, we explored how to teach human-following robots' behavior, which we believe to be a meaningful endeavor.

In this paper, we applied 3D object detection techniques for human detection in novel scenes, serving as visual input for a robotic system. Leveraging 3D object detection technology and MPC, we proposed a new metric that combines 3D IoU and state anticipation to determine human-following failure in occluded scenarios. To achieve state anticipation, we proposed a Deep Model Predictive Control (Deep-MPC) approach, which is a data-driven method that exploits online learning from data collected during robot-environment interactions to infer the system's future states and optimize the robot's current actions. Deep-MPC employs a neural network as the state transition module, taking states and actions as inputs and predicting the subsequent state. The method performs predictions to obtain predicted states for N steps, calculates the loss function by comparing these states to the target state, and finally optimizes the actions at each time step through gradient backpropagation. Deep-MPC is a fully online learning approach wherein the human-following robot interacts with the environment, using collected data to optimize its model and learn action policies. One of its advantages compared to existing human-following robot control methods is that it does not require manual construction of the robot's motion model or the design of a controller, as these operations are autonomously learned through Deep-MPC.

We applied our proposed method to real human-following robot tasks, conducted online learning, long-term following experiments, experiments in occluded and crowded scenes, and outdoor human-following experiments. Furthermore, we compared our method with some existing methods, 3D skeleton-based methods and methods without the Deep-MPC. Our method achieved higher success rates in the same occluded scenes and exhibited better robustness in more severe occlusion. Our main contributions are as follows:

- We proposed Deep-MPC, a method that enables online learning and anticipatory control, and applied it to human-following robots.
- We proposed a novel metric for evaluating failures in human-robot following that combines 3D IoU and state anticipation.
- We conducted extensive real-world human-following experiments in challenging environment to validate the advantages of our proposed method.

II. Real-time 3D Human Detection

Most existing 3D object detection methods heavily rely on 3D annotated datasets [30]; however, it is not very efficient for real-world robot applications to manually label many 3D bounding boxes due to the diversity of application scenarios for robots. To achieve handy 3D object detection on robots, we adopted a 3D detector named Recursive Cross-View (RCV) [31], which takes point clouds as input and outputs 3D bounding boxes. RCV circumvents the laborious task of manually labeling 3D annotations by transforming 3D object detection into several 2D detection through the three-view principle and recursive optimization. Compared to 3D labeling, 2D labeling is greatly simpler and easier. Furthermore, with the maturity of 2D detection technology, it is possible to obtain a reliable 2D detector through training on several hundred 2D boxes. After spending several hours annotating 2D data and training, RCV can perform 3D human detection, as shown in Figure 1. Despite limited computing power (Nvidia 1060) on a mobile robot, it attains at approximately 8fps on a live RGB-D stream. As a consequence, we applied it to a mobile robot, endowing

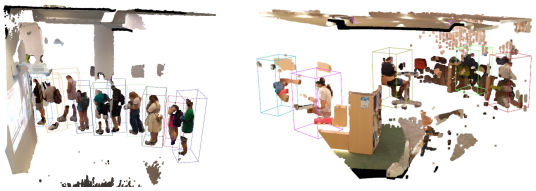

Fig. 1. The performance of RCV for 3D human detection on our own dataset.

the robot with the ability to perceive the 3D physical world. Using this capability as a foundation, we further devised the robot's anticipatory control and autonomous learning model. Please refer to [31] for more details.

III. Anticipatory Control

A. Deep Model Predictive Control

Model Predictive Control (MPC) is a type of advanced control strategy that uses a mathematical model of the system being controlled to predict future behavior and determine the optimal control actions to achieve desired objectives. Generally, MPC can be described as Eq.(1).

$$\min_{a_i} \sum_{i=t+1}^{t+H} F(s_i, \hat{s}_i) \tag{1}$$

s.t.

$$s_{i+1} = T(s_i, a_i)$$

where F is the cost function that measures the distance between s_i and $\hat{s}_i$. s_i is the state at time i, and $\hat{s}_i$ is the goal state at time i. H is the horizon. T is the transition function of the system. a_i is the action adopted by the system at time i.

MPC possesses the capability of anticipatory control, as it rollouts the future states H-steps ahead and utilizes them to optimize the current actions. To achieve that, one needs to formulate the mathematical or physical model of system transitions, which can sometimes be difficult or even infeasible. In this paper, we propose Deep-MPC, which is an online data-driven MPC that enables learning from scratch. In other words, a robot can learn how to finish a task without any prior knowledge of system transitions by using Deep-MPC. Figure 2 demonstrates the overview of Deep-MPC. Here, a neural network is leveraged to learn the transition function, that is T, see Section III-B for more details.

In Deep-MPC, a robot captures environmental data using an RGB-D sensor, which is subsequently processed by RCV, a 3D object detector. RCV yields the status of objects in relation to the robot, providing spatial information for motion planning and control. Then, the robot rollouts H steps using T. At each step, the robot samples an action (a_i), which is then fed into T along with s_i. As a consequence, an anticipatory state s_{i+1} can be generated by T. Repeatedly, an anticipatory sequence of states can be obtained, as shown in Figure 2. Compared with expected states, $\hat{s}_i$ in Figure 2, by loss function (F), we can obtain the total loss of anticipatory states, which is leveraged to optimize actions. Note that the expected states

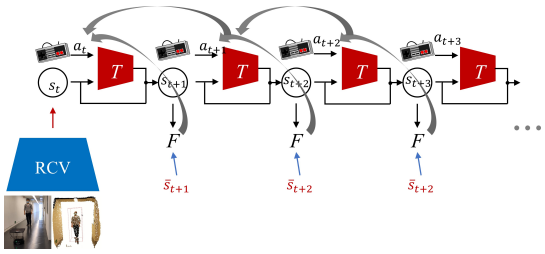

Fig. 2. Overview of Deep-MPC. The red arrow indicates state perception, the black arrows represent forward data flow, and gray curved arrows denote gradient back-propagation. T is the transition model formulated by a neural network.

are manually specified by a human operator; for instance, in our experiments, the human-specified expected coordinates (x=0, y=2.2m) relative to the robot are utilized. The state (s_i) observed by the robot is generated from the center point of the 3D bounding box detected by RCV.

Once the anticipatory sequence of states ($s_i\{i = t+1, ...t+H\}$) and actions ($a_i\{i = t+1, ...t+H\}$) is obtained, a gradient-based optimization method can be applied to optimize the actions, as the transition model (T) is implemented as a neural network. The flow of gradient is shown as gray curved arrows in Figure 2. The update law is shown in Eq. (2).

$$a_i \leftarrow a_i - \eta \frac{\partial \sum_{n=t+i+1}^{t+H} l_n}{\partial a_i} \tag{2}$$

where η is the learning rate, and $l_n = F(s_n, \hat{s}_n)$ is the loss at step n. Specifically, a_i affects only the loss generated in the subsequent time steps, that is from t+i+1 to t+H. In this manner, all actions can be planned to decrease the total loss.

In general, the control law for Deep-MPC can be summarized as follows:

$$\min_{a_i} \sum_{i=t+1}^{t+H} ||s_i - \hat{s}_i||^2 \tag{3}$$

s.t.

$$\begin{cases} a_i = P(s_i) \\ s_{i+1} = T_\theta(s_i, a_i, \delta t) + s_i \\ a_i \leftarrow a_i - \eta \frac{\partial \sum_{n=t+i+1}^{t+H} ||s_i - \hat{s}_i||^2}{\partial a_i} & \text{for 0 to H} \\ \theta \leftarrow \theta - \beta \frac{\partial ||s_{i+1} - (T_\theta(s_i, a_i, \delta t) + s_i)||^2}{\partial \theta} & \text{for every } k \text{ steps} \end{cases}$$

where $s_i \in \mathbb{R}^{1\times 2}$ represents the position (x, y) of the followed target in the robot coordinate system. $a_i \in \mathbb{R}^{1\times 2}$ represents the forward velocity and rotation velocity of the mobile robot, and $\delta \in \mathbb{R}^{1\times 1}$ represents the gap time between s_i and s_{i+1}. The output is the next state s_{i+1}. Next, we introduce each components of Deep-MPC in detail.

B. Transition Model

In MPC, the transition model (T_θ) is employed to anticipate the succeeding state. In some previous studies, researchers formulated a neural network that directly predicts next state based on the current state and action, as shown in the left-hand side section in Figure 3 and Eq.(4).

$$s_{i+1} = T_\theta(s_i, a_i) \tag{4}$$

where θ represents the parameters of a neural network. By contrast, we propose a new structure, see the right-hand side section in Figure 3, which utilizes a neural network to predict the state's rate of change, given the current state, action, and interval. This is inspired by the state equation of a system, see Eq. (5).

$$\begin{cases} \dot{s} = As + Ba \\ y = Cs + Da \end{cases} \tag{5}$$

where A, B, C, and D are coefficient matrices. s, a, and y refer to the state, action and output of the system, respectively. Specifically, the differential equation express the state variables as time derivatives, describing how the system's state changes with respect to time. The response of the system, given initial states and inputs, can be derived. Inspired by this, we employ a neural network, specifically a MLP with ReLU as the activation function, which does not include any recurrent units. This neural network is used to simulate the differential equation and predict the next state, as depicted in Figure 3 and described by Eq. (6).

$$s_{i+1} = T_\theta(s_i, a_i, \delta t) + s_i \tag{6}$$

where δt is the interval between s_i and s_{i+1}. In experiments, we observe several advantages of this architecture, including faster convergence rates, and reduced prediction errors resulting from data scarcity in early robot task learning. Similarly, this architecture can be viewed as a form of skip connection [32], but we only implement skip connections for a subset of the input.

We adopt an online mode to train the transition model, as illustrated in Figure 4. The implementation details can be accessed in the Appendix. The update law is shown in Eq. (7).

$$\theta \leftarrow \theta - \alpha \frac{\partial ||s_{i+1} - (T_\theta(s_i, a_i, \delta t) + s_i)||^2}{\partial \theta} \tag{7}$$

The robot executes actions in the real-world environment and collects state transition data $(s_i, a_i, \delta_i, s_{i+1})$, which is then used to train the transition model. At each step, the robot captures environmental data using an RGB-D sensor, which is subsequently processed by RCV, which yields the status of objects in relation to the robot. For further details on the training process, please refer to the next section.

C. Anticipatory Control

In this section, we developed a complete anticipatory control algorithm, see **Algorithm 1**, which can be applied to mobile robots, based on the Deep-MPC and the transition model. A mobile robot is outfitted with a 3D human detector (RCV) that has been trained for the purpose of detecting humans in a 3D space. Additionally, the robot is equipped with sensors that can capture RGB and point cloud data from the environment in real time. The robot performs the following operations:

1) **Executing Deep-MPC** based on current state perceived by RCV. In each rollout, a proportional controller (P) is leveraged to compute a coarse action. After anticipating H steps ahead, a gradient descent algorithm is employed to optimize all actions with the aim of minimizing the distance between the anticipated states and the target states.
2) **Executing the first action**, followed by perception of the environment to obtain the next state. Note that we utilize receding MPC, which involves using the transition model to predict future states and optimize actions over a finite time horizon; however, only the first action is performed.
3) **Collecting transition data** of the robot, then updating the transition model.

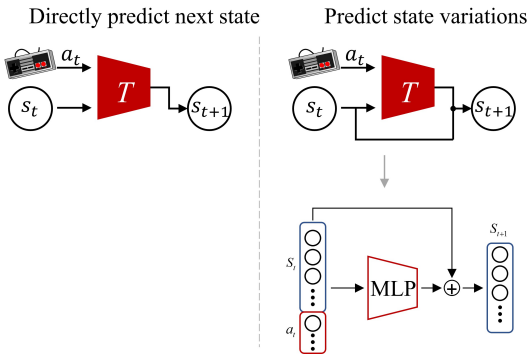

Fig. 3. Transition models. The left-hand side represents a direct transition model that predicts the next state based on the current state and action, while the right-hand side represents a transition model that predicts state variations. A multi-layer perceptron (MLP) with ReLU as activation function is applied.

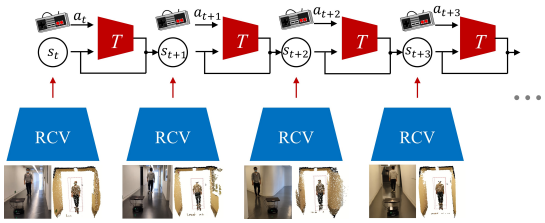

Fig. 4. Online training of the transition model.

In the first operation, we employed a simple proportional controller to generate initial robot actions, which accelerates the convergence rate of action optimization compared to random sampling or training a policy network. This approach leads to improved efficiency of robot learning by allowing the optimization process to converge more quickly to an effective solution. Note that this simple proportional controller does not utilize any knowledge of the system dynamics, which means that the robot is still able to learn independently how to perform the task. All operations are executed online, allowing the robot to collect data and optimize its model simultaneously, and to learn how to complete tasks quickly by interacting with the real environment as much as possible. This approach is of significant importance for robot applications, as it reduces the need for complex modeling processes and enhances the level of intelligence of the robot.

Algorithm 1 Anticipatory Control on Mobile Robots

Require: Create an empty buffer D, a proportional controller P, anticipatory horizon H, expected states $\{s_0^{H-1}\}$.
Require: Initialize neural network T_θ with parameters θ, and a mobile robot with a RGB-D sensor.
Require: Trained RCV model.
  while not ended **do**
    Capture environmental data using an RGB-D sensor and note as *env*.
    $s_0 \leftarrow$ RCV(*env*)
    for $i \leftarrow 0$ to L **do**
      // Deep-MPC
      for $h \leftarrow 0$ to $H-1$ **do**
        $a_{i+h} \leftarrow P(s_{i+h})$ {// P controller as a coarse Actor.}
        $s_{i+h+1} \leftarrow T_\theta(s_{i+h}, a_{i+h}, \delta t) + s_{i+h}$
        $l_{i+h+1} \leftarrow ||\hat{s}_h - s_{i+h+1}||^2$
      end for
      for $step \leftarrow 0$ to n **do**
        Update $\{a_i^{i+H-1}\}$ using Eq.(2)
      end for
      // Environment Interaction
      Perform a_i on the mobile robot.
      Capture environmental data (*env*) using an RGB-D sensor.
      $s_{i+1} \leftarrow$ RCV(*env*)
      Add experience to buffer $D \leftarrow D \bigcup (s_i, a_i, \delta t, s_{i+1})$
      // Transition Learning
      for every k steps **do**
        Perform a gradient descent step on $||s_{t+1} - (T_\theta(s_t, a_t) + s_t)||^2$ with respect to θ.
      end for
    end for
  end while

D. Following Metric

Initially, we established the criteria for robot following failure, as many prior studies lacked a quantitative definition of when robot following is considered unsuccessful. In occluded or crowded environments, other individuals may be present around the target being followed. Hence, we determined that the robot's following has failed if either **(1) the distance between the target detected in two consecutive successful detections exceeds a predefined threshold** or **(2) the robot loses the target for a prolonged period or the target moves out of range.** To elaborate on the first criterion, we suggested that the robot should continually perceive the target's position over time. However, in multi-person occlusion scenarios, this continuity may be disrupted when the target is temporarily blocked from view. If the disruption surpasses a specific threshold, the robot may not accurately determine the target's position, necessitating re-identification of the target. It is pertinent to note that the re-identification process is not within the scope of this study and will be addressed in future research.

For the first criterion, some metrics can be applied, such as Euclidean distance. Furthermore, our method benefits from

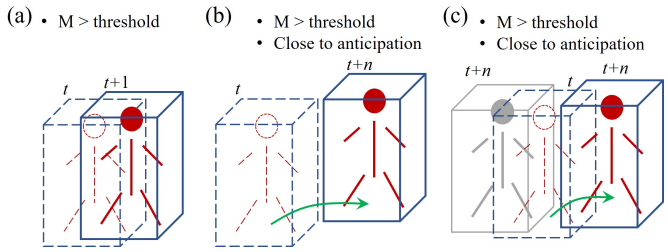

Fig. 5. A novel metric that combines 3D IoU with state anticipation. The red person is the target that the robot is following, while the grey person is considered as a disturbance.

3D object detection technology, which enables us to achieve continuous following of a particular individual, based on 3D intersection over union (IoU). The continuity of human motion over both space and time leads to the intersection of 3D bounding boxes detected at two closely spaced moments, see Figure 5(a), which can be exploited to ensure that a robot can continuously follow a target. However, the 3D IoU metric fails to work when the target is occluded for an extended period of time, as illustrated in Figure 5(b). For example, when the target is occluded and then re-detected by the robot after a period, the 3D bounding box detected at this moment does not intersect with the one detected in the previous moment, making it impossible to utilize 3D IoU to continue following the target. Additionally, in crowded scenes with occlusions, as shown in Figure 5(c), it is highly possible to detect multiple 3D boxes that intersect with the previous target box, rendering the 3D IoU-based method ineffective.

To alleviate this issue, we proposed a novel metric that combines 3D IoU with state anticipation, represented by Eq. (8).

$$\mathrm{M} = (1-\alpha)U(B_t, B_{t+n}) + \alpha \max(0, 1 - F(s_{t+n}, \bar{s}_{t+n})) \quad (8)$$

where U is the function that computes 3D IoU between B_t and B_{t+n}, F is the function that measures the distance between two states. s_{t+n} is the state detected at $t+n$, $\bar{s}_{t+n}$ is the state anticipated by our method at $t+n$, α is a factor that balances these two components, and $\alpha = 0.1(n-1)$. n refers to the detection interval between two successful detections. Specifically, $n=1$ indicates that the target has been detected in two consecutive detections. $n>1$ implies that the target is not detected for $n-1$ times between two successful detections. We set a threshold for the metric M, such that when M falls below this threshold, the detection is considered a failure, indicating that the algorithm failed to detect the target. The algorithm then proceeds to the next detection. Figure 5 demonstrates the metric. In Figure 5(a), M is equal to the 3D IoU between the two boxes, as both detections are successful. In Figure 5(b) and (c), due to reasons such as occlusion, the robot fails to detect the target for $n-1$ times between two successful detections, but detected the target at $t+n$ with a state close to the anticipated state (indicated by the green arrow). Since M exceeds the threshold, the detection is judged as successful. Note that when n>9 (criteria (2)) and the target is still not detected, we regarded the task as failed. The experimental

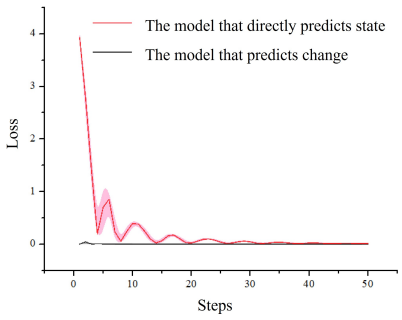

Fig. 6. Convergence speed and error values of two transition models.

results demonstrate that the proposed metric can improve the following stability of the robot in occluded scenarios.

To ensure a fair comparison of the characteristics of these three metrics, namely Euclidean distance, 3D IoU, and M, we established thresholds for each of them. Specifically, the Euclidean distance threshold was set at 0.4m, the 3D IoU threshold was set at 0.2, and the M threshold was set at 0.2 as well. It should be noted that, in our experiments, we calculated the average Euclidean distance corresponding to 3D IoU values close to 0.2, which was approximately 0.4m. As M is on the same scale as the 3D IoU values, we used the same value for its threshold. In our comparative experiments, we evaluated the impact of these three metrics on the experimental results. Our newly proposed metric, M, demonstrated superior performance compared to the other two metrics.

IV. Experiments

A. Platform

We assembled a physical mobile robot as an experimental platform for our study. The equipment used in the platform consisted of a TurtleBot2 mobile robot, an RGB-D sensor (Azure Kinect DK), and a laptop computer with an Intel Core i5 8^{th} CPU and a Nvidia 1060 GPU. Our method can achieve a control rate of around 6Hz.

B. Comparison of Two Transition Models

In Figure 3, we mentioned two transition models: one that directly predicts the next state, and another that predicts the rate of change of the state. Here, we compared the performance of these two models. As our method employs online learning, where the robot updates the model while interacting with the environment, we aim for the transition model to converge as quickly as possible with minimal early training error. The training data for the two models were obtained from our previous study, where we collected 2,000 state transition data sets for training. It's worth noting that all hyperparameters were configured identically for both models. Figure 6 shows the convergence speed and training error of the two models, and it is evident that our proposed model exhibits exceptional performance, making it better suited for robot online learning scenarios.

C. Control Variables

The most important control variables in Equation (3) are the learning rate η of the action (a_i), the system state $\hat{s}_i$, and the horizon (H). We conducted multiple experiments to determine the optimal values of these variables. Initially, based on our experience, we set $\hat{s}_i$ to (0, 2.2m). Then, we explored the impact of different action learning rates on the human-following experiments. We evaluated the following performance of the system using the system's following state (s_i), and the experimental results are shown in Table I. Our findings indicated that the system achieved the best following performance when the action learning rate was set to 0.75. It is important to note that we conducted independent experiments five times for each parameter. Furthermore, to ensure real-time performance of the system, we set the hyperparameter "Epoch for Deep-MPC" to 5. With the action learning rate set to 0.75, we varied the values of $\hat{s}_i$ and conducted independent experiments five times for each value. The experimental results, presented in Table II, demonstrate that our robot maintained good following performance across different $\hat{s}_i$ values. Regarding the horizon H in Deep-MPC, we conducted experiments to explore the system's state error and optimization time for different values while keeping the system state and action learning rate at their optimal values. The experimental results are presented in Table III. To strike a balance between error values and optimization time, we set the H value to 5. It is important to note that for each H value, we conducted experiments using 100 sets of data.

Finally, Table IV presents the values of these variables, along with other hyperparameters, which were determined through experimental evaluation.

TABLE I
COMPARATIVE EXPERIMENTS OF η. 'ERROR' IS THE DISTANCE BETWEEN s_i AND $\hat{s}_i$.

η	0.10	0.45	0.75	1.5
Error	(-0.21, 0.45)	(-0.16, 0.21)	(-0.09, 0.07)	(-0.10, 0.08)

TABLE II
COMPARATIVE EXPERIMENTS OF $\hat{s}_i$. 'ERROR' IS THE DISTANCE BETWEEN s_i AND $\hat{s}_i$.

$\hat{s}_i$/m	(0, 1.8)	(0, 2.2)	(0, 2.6)	(0, 3.0)
Error	(-0.10, 0.06)	(-0.09, 0.07)	(0.08, 0.08)	(-0.08, 0.07)

TABLE III
COMPARATIVE EXPERIMENTS OF H.

H	3	5	8	10
loss (m)	0.057	0.055	0.052	0.051
t (s)	0.072	0.085	0.120	0.137

D. Online Learning

In this section, we performed experiments on human-following task using online learning. The mobile robot opti-

TABLE IV
CONTROL VARIABLES OF DEEP-MPC.

$\hat{s}_i$	η	H	Epoch for Deep_MPC	β	k
(0,2.2)	0.75	5	5	0.001	20

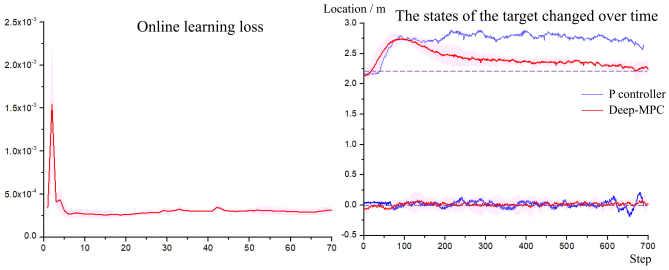

Fig. 7. Online learning loss and system state variations.

mizes its transition model and actions based on date collected through interaction with the physical world. The algorithm used in this experiment is presented in Algorithm 1. Initially, the robot has no interaction data and could only attempt some actions resulting from unreliable anticipatory control to interact with the environment. As more interaction data is collected, the robot gradually became capable of executing effective anticipatory control and subsequently learns how to complete the human-following task.

To demonstrate the learning process and performance of the proposed method, we designed a online human-following scenario where the human was required to maintain a similar walking speed and trajectory. A P controller without Deep-MPC optimization (i.e., the action generated by the P controller in Algorithm 1 is not optimized by Deep-MPC) was also applied in this task as a comparative experiment. The experiment was conducted with a target state of (x,y) = (0, 2.2m), requiring the robot to maintain the target directly in front of it at a distance of 2.2 meters. The results demonstrate that, as shown in the right-hand side of Figure 7, Deep-MPC can quickly learn how to follow the target within one minute and exhibits significantly better performance than the P controller without Deep-MPC optimization. The left-hand side of Figure 7 shows the online learning loss of the transition model, which is updated every 10 interactions between the robot and the environment. Initially, the loss is high due to the limited number of training. As the number of training increases, the loss value decreases, but the distance between the system state and the target state keeps increasing due to overfitting of the model with insufficient data. The model cannot effectively generalize to new states due to the overfitting. However, as more data is collected, the predictive power of the transition model is enhanced, enabling the robot to perform effective anticipatory control and gradually approach the target state. Note that we conducted three experiments using online learning with Deep-MPC.

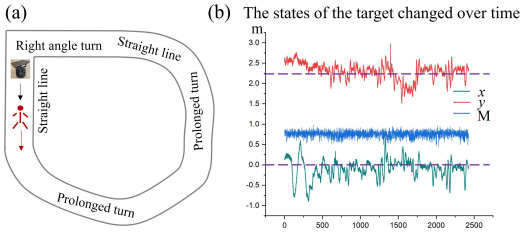

Fig. 8. Long-term following experiments. (a) The experimental site. (b) The states of the target.

E. Long-term Following

To validate the stability of the proposed method, we applied the model learned through online learning in the previous section to long-term human-following experiments. In this experiment, the robot is required to follow the target for an extended duration of approximately 500 seconds. We selected a circular corridor within a building, as illustrated in Figure 8(a), which includes straight paths, prolonged turn paths, and an right angle turn. Figure 8(b) presents the variation in the system state and M during one experiment. Due to the walking speed and sudden changes in direction, such as when turning, the system state exhibits fluctuations. However, the states generally remain close to the target states, indicating the fundamental effectiveness of our proposed method. A total of 10 experimental trials were conducted, and the robot achieved a success rate of 100%. A complete experimental video is available in here.

F. Occluded and Crowded Following

To verify the robustness of our method, we conducted robot following experiments in environments with occlusion and congestion. The metric M was utilized to determine whether the experiment failed. Two scenarios were considered: a cluttered and narrow office space, and an entrance area of a building. Figure 9 shows the office, where the background is cluttered and the space is confined. Nevertheless, the robot was able to stably follow the target even when other people occlude the target or when multiple people are present within the field of view, as shown in Figure 9. The full experiment video can be found in here. In addition, we deployed the robot at the entrance of a building to evaluate the sustained object following capability of the proposed method, as shown in Figure 9. We performed four experiments, each lasting approximately seven minutes. During the experiments, there was at least one people and up to six people within the robot's field of view. Table I shows the success rate of the method in following the target for different numbers of people within the field of view. The results demonstrate that our method achieves a very high success rate, with several failures occurring when there are many people (more than 4 people) within the field of view, see the last case in Figure 9, causing the robot to lose the target. Figure 9 illustrates some experimental scenes, including the failed case. Overall, our proposed method achieves a very high success rate in completely unstructured and randomly interfered scenarios, demonstrating the effectiveness of our method. The full experiment video can be found in here.

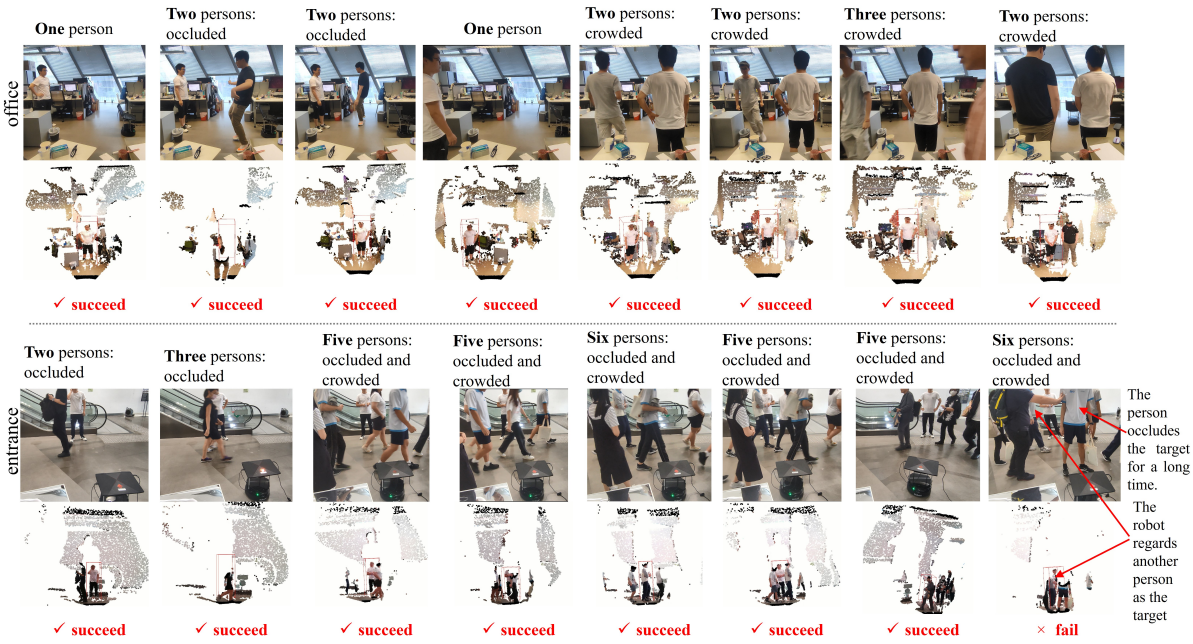

Fig. 9. Occluded and crowded following experiments, from both a third-party perspective (up) and the robot's perspective (down). More experiments can be found in here.

Our method exhibits high robustness, mainly attributed to Deep-MPC's ability to predict future states. The robot can anticipate the location of the target when the target is occluded. When the occlusion disappears, the robot can match the anticipated state with the detected state to recapture the target and continue following. Our method's ability to reliably anticipate the target's state is evidenced by the statistical analysis of 1200 consecutive sets of predicted and actual target locations, which exhibit an average error of approximately 0.045m.

G. Comparative Experiments

To demonstrate the performance of our proposed method, we conducted comparative experiments in the entrance area of a building. Specifically, we carried out the method in [14], STPOTR [33], the 3D skeleton-based method, the method without the Deep-MPC module, and the method utilizing the Deep-MPC module in occluded and crowded scenes. The 3D skeleton-based method calculates the average position of all skeleton points as the target position, and uses Euclidean distance metric in Section III-D to determine the failure of following. The method without the Deep-MPC module employs a PI controller to control the robot's behavior, and uses 3D IoU metric in Section III-D as the criterion to determine the failure of following. The results of the experiments, as shown in Table V, indicate that our proposed method achieves a higher success rate than the other methods, even in the presence of more severe occlusion.

TABLE V
RESULTS OF COMPARATIVE EXPERIMENTS. OCC. MEANS THE NUMBER OF PEOPLE OCCLUDED WITHIN THE ROBOT'S FIELD OF VIEW. S AND F DENOTE SUCCESS AND FAILURE RESPECTIVELY.

Method	Metric	Occ.	S	F	Rate
3D skeleton + Prediction + PID [14]	/	1	7	5	54.55%
		2	5	5	50.00%
		3	1	2	33.33%
Overall					52.00%
STPOTR [33]	/	1	5	5	50.00%
		2	3	6	33.33%
		3	0	3	0
Overall					36.36%
3D skeleton	Euclidean distance	1	9	7	56.25%
		2	5	3	62.50%
		3	0	2	0%
Overall					53.85%
3D box + 3D IoU +PI controller	3D IoU	1	23	3	88.46%
		2	5	1	83.33%
		3	3	1	75.00%
Overall					86.11%
3D box + 3D IoU + Deep-MPC	M	1	29	0	**100%**
		2	23	0	**100%**
		3	9	1	**90.00%**
		4	7	2	**88.89%**
		5	3	1	**75.00%**
Overall					**94.67%**

V. DISCUSSION

A. Detection Interval

Our method achieves an average detection interval of approximately 0.167 seconds (6Hz), which means that the time interval between s_t and s_{t+1} in Figure 3 is 0.167s. However, in the presence of occlusion, the robot may lose track of the target, resulting in an elongated interval between s_t and s_{t+1}. In order to alleviate the decline in the performance of the transitional model caused by the difference in detection intervals due to occlusion, we also used the detection interval as one of the inputs to the transitional model.

B. Robot Movement

In the human-following experiments, our objective is to maintain the relative position between the human and the robot, rather than ensuring that the robot precisely follows the trajectory of the human's movements. For instance, when the human walks around the robot, the robot only needs to adjust its orientation without necessarily tracing the exact trajectory of the human's movements.

C. Onset of Tracking

The RCV model currently lacks the capability to recognize different individuals. In our experiments, we position the target to be followed appropriately relative to the robot and make efforts to ensure there are no other people around the target. However, this setup is a compromise as we currently lack a better method to accurately identify the target in crowded scenes at the beginning of the following task. For instance, using facial recognition is challenging to guarantee capturing the target's face by the robot initially, and relying on clothing or other features is difficult to ensure uniqueness. The current focus of our work primarily revolves around the robot's following control algorithm and the following metric. Accurately identifying the target from multi-person scenes at the start of the following task will be the subject of our future research.

D. Transition Model

In our research, we also considered using Recurrent Neural Networks (RNNs) to implement the transition model. However, during our literature review, we found that RNNs are challenging to use in fully online tasks due to their computational expense and training difficulties [34], [35]. Our proposed Deep-MPC approach is designed for fully online learning, starting from scratch, which is particularly difficult to achieve with RNNs. On the other hand, the recursive pattern we currently use is also applied in some simulated robot research [17].

E. The Role of Occlusion

Occlusions play a critical role in the target following task as they hinder the robot's ability to maintain visual contact with the target. When the target is occluded, the robot faces the challenge of losing visibility and struggles to accurately determine the target's position and movement. To address this issue, our proposed approach leverages the robot's perception, anticipation, and control capabilities. By utilizing a 3D human detection method, we can detect a 3D bounding box even when the target is partially occluded. Then, we propose a following metric which combines 3D IoU and anticipation. When the target is occluded during the following task and cannot be detected, the robot intensifies its reliance on trajectory anticipation. The longer the occlusion persists, the stronger the reliance on trajectory anticipation becomes, until the point where the following is deemed to have failed.

F. The Setting of Transition Model

Deep-MPC is a fully online approach that requires simultaneous data collection for transition model learning and optimization of robot actions. This necessitates a high demand for real-time performance in the algorithm. While our settings may appear uncommon, they are driven by the requirements of our fully online application scenario. For the model, we employ a simple structure (Figure 3) to ensure rapid convergence. Consequently, we cannot set many epochs, as it would prolong online training time. Furthermore, a moderate batch size is preferred to avoid excessive computational demands on the system. Through multiple real robot experiments, we determined that 5 epochs and a batch size of 128 strike a balance between rapid model convergence and real-time performance within the available computing resources. Additionally, we observed that due to the simplicity of the MLP structure we employed, adjusting these parameters to different values yielded similar training results.

VI. Appendix

TABLE VI
THE IMPLEMENTATION DETAILS OF ONLINE TRAINING OF TRANSITION MODEL.

Training loss	validation loss	structure	epoch
0.032	0.040	[5,256,1024, 512,128,2]	5
lr	batch size	samples	training time
0.001	128	2500	about 20s

References

[1] G. Sawadwuthikul, T. Tothong, T. Lodkaew, P. Soisudarat, S. Nutanong, P. Manoonpong, and N. Dilokthanakul, "Visual goal human-robot communication framework with few-shot learning: a case study in robot waiter system," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1883–1891, 2021.

[2] B.-J. Lee, J. Choi, C. Baek, and B.-T. Zhang, "Robust human following by deep bayesian trajectory prediction for home service robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, DOI 10.1109/ICRA.2018.8462969, pp. 7189–7195, 2018.

[3] D. Han and Y. Peng, "Human-following of mobile robots based on object tracking and depth vision," in *2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA)*, DOI 10.1109/ICMRA51221.2020.9398366, pp. 105–109, 2020.

[4] M. Zhang, X. Liu, D. Xu, Z. Cao, and J. Yu, "Vision-based target-following guider for mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9360–9371, 2019.

[5] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2d laser scanners," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, DOI 10.1109/ICRA.2015.7139259, pp. 726–733, 2015.

[6] G. Ferrer, A. G. Zulueta, F. H. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous robots*, vol. 41, no. 4, pp. 775–793, 2017.

[7] N. Van Toan, M. Do Hoang, P. B. Khoi, and S.-Y. Yi, "The human-following strategy for mobile robots in mixed environments," *Robotics and Autonomous Systems*, vol. 160, p. 104317, 2023.

[8] S. Siva and H. Zhang, "Robot perceptual adaptation to environment changes for long-term human teammate following," *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 706–720, 2022.

[9] D. Han and Y. Peng, "Human-following of mobile robots based on object tracking and depth vision," in *2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA)*, pp. 105–109. IEEE, 2020.

[10] C. Dondrup, N. Bellotto, F. Jovan, M. Hanheide *et al.*, "Real-time multisensor people tracking for human-robot spatial interaction," 2015.
[11] K. O. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1710–1715. IEEE, 2008.
[12] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.
[13] J.-S. Hu, J.-J. Wang, and D. M. Ho, "Design of sensing system and anticipative behavior for human following of mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 61, DOI 10.1109/TIE.2013.2262758, no. 4, pp. 1916–1927, 2014.
[14] A. Wang, Y. Makino, and H. Shinoda, "Machine learning-based human-following system: Following the predicted position of a walking human," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4502–4508. IEEE, 2021.
[15] J. Chen and W.-j. Kim, "A human-following mobile robot providing natural and universal interfaces for control with wireless electronic devices," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2377–2385, 2019.
[16] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.
[17] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," in *International Conference on Learning Representations*, 2019.
[18] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.
[19] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," in *International Conference on Machine Learning, PMLR*, 2022.
[20] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89–112, 1997.
[21] S. Lucia, D. Navarro, B. Karg, H. Sarnago, and O. Lucia, "Deep learning-based model predictive control for resonant power converters," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 409–420, 2020.
[22] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3866–3878, 2020.
[23] D. Hoeller, F. Farshidian, and M. Hutter, "Deep value model predictive control," in *Conference on Robot Learning*, pp. 990–1004. PMLR, 2020.
[24] M. Elnour, Y. Himeur, F. Fadli, H. Mohammedsherif, N. Meskin, A. M. Ahmad, I. Petri, Y. Rezgui, and A. Hodorog, "Neural network-based model predictive control system for optimizing building automation and management systems of sports facilities," *Applied Energy*, vol. 318, p. 119153, 2022.
[25] I. Lenz, R. A. Knepper, and A. Saxena, "Deepmpc: Learning deep latent features for model predictive control." in *Robotics: Science and Systems*, vol. 10, p. 25. Rome, Italy, 2015.
[26] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.
[27] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.
[28] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *Conference on Robot Learning*, pp. 420–429. PMLR, 2020.
[29] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on Robot Learning*, pp. 2226–2240. PMLR, 2023.
[30] Q. Meng, W. Wang, T. Zhou, J. Shen, Y. Jia, and L. Van Gool, "Towards a weakly supervised framework for 3d point cloud object detection and annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4454–4468, 2021.
[31] S. Gui and Y. Luximon, "Recursive cross-view: Use only 2d detectors to achieve 3d object detection without 3d annotations," *IEEE Robotics and Automation Letters*, 2023.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
[33] M. Mahdavian, P. Nikdel, M. TaherAhmadi, and M. Chen, "Stpotr: Simultaneous human trajectory and pose prediction using a non-autoregressive transformer for robot follow-ahead," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9959–9965. IEEE, 2023.
[34] S. Nath, V. Liu, A. Chan, X. Li, A. White, and M. White, "Training recurrent neural networks online by learning explicit state variables," in *International conference on learning representations*, 2019.
[35] M. N. Fekri, H. Patel, K. Grolinger, and V. Sharma, "Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network," *Applied Energy*, vol. 282, p. 116177, 2021.

Shun Gui is currently a PhD candidate at Hong Kong Polytechnic University, specializing in robot vision, human-robot interaction, and robot learning techniques. He received his bachelor's degree from Beijing University of Posts and Telecommunications in 2013 and his master's degree from Nanyang Technological University in 2021.

His research interests focus on developing machine learning algorithms and computer vision techniques to enhance robot perception and interaction in real-world environments

Yan Luximon obtained the PhD degree from the Hong Kong University of Science and Technology.

She is an Associate Professor in School of Design at The Hong Kong Polytechnic University. She also serves as Chair of School Research Committee, Lab Leader for Asian Ergonomics Design Lab and Deputy Discipline Leader for BA Product Design. She is also President of Hong Kong Ergonomics Society (HKES). She has published over 100 peer-reviewed journal papers, book chapters, patents and international conference papers. Her research interests include ergonomic design, anthropometry and its application in design, 3D digital human modeling and CAD, AI design tool and visualization, head and face related products, human computer interaction, cultural difference, statistical and mathematical models.