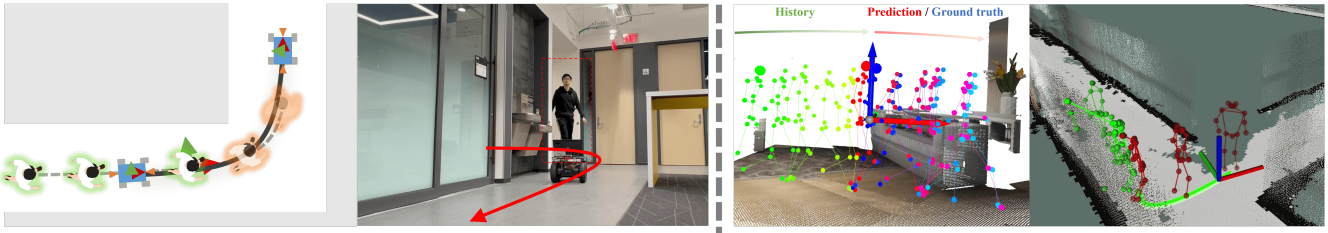
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Map-Aware Human Pose Prediction for Robot Follow-Ahead

Qingyuan Jiang, Burak Susam, Jun-Jee Chao and Volkan Isler
University of Minnesota
Shepherd Laboratories, 100 Union St SE
{jian0345, susam001, chao0107, isler}@umn.edu



(a) Robot Follow-Ahead: illustration (left) and real-robot demonstration (right)

(b) Human Pose Forecasting: Synthetic (left) and realistic (right) dataset

Fig. 1: **(a) The Robot Follow-Ahead Task:** A mobile robot maintains the sight of a human actor while driving in front of them in an indoor environment **(b) Map-aware human pose prediction.** To achieve the follow-ahead task, given the pose histories (shown in green), we predict long-term human poses (shown in red with ground truth in blue) by incorporating the local map information and generating input for a predictive robot controller.

***Abstract*— In the robot follow-ahead task, a mobile robot is tasked to maintain its relative position in front of a moving human actor while keeping the actor in sight. To accomplish this task, it is important that the robot understand the full 3D pose of the human (since the head orientation can be different than the torso) and predict future human poses so as to plan accordingly. This prediction task is especially tricky in a complex environment with junctions and multiple corridors. In this work, we address the problem of forecasting the full 3D trajectory of a human in such environments. Our main insight is to show that one can first predict the 2D trajectory and then estimate the full 3D trajectory by conditioning the estimator on the predicted 2D trajectory. With this approach, we achieve results comparable or better than the state-of-the-art methods three times faster. As part of our contribution, we present a new dataset where, in contrast to existing datasets, the human motion is in a much larger area than a single room. We also present a complete robot system that integrates our human pose forecasting network on the mobile robot to enable real-time robot follow-ahead and present results from real-world experiments in multiple buildings on campus. Our project page, including supplementary material and videos, can be found at: `https://qingyuan-jiang.github.io/iros2024_poseForecasting/`**

I. INTRODUCTION

Imagine a robot working as a photographer and recording videos in front of a moving actor. To keep sight of the front of the actor's body and facial expressions, the robot needs to drive in front of the actor, catch their pace, and actively predict their motion while avoiding obstacles in the environment. This task of controlling a robot to maintain the visibility of a human actor while driving in front of them, is called "robot follow-ahead" [1], [2]. A key component of existing robot follow-ahead approaches is to predict future human poses [3] to help the robot stay in front of the actor without losing their sight.

While arbitrary human trajectories are feasible in open spaces, obstacles in indoor environments constrain the set of possible motions and, therefore, reduce the space of available trajectories. To provide sufficient information for human pose forecasting, researchers have been exploring the advantage of using environmental information [4], [5], [6], [7]. Suppose a human is walking in a hallway facing a left turn ahead. With high probability, the actor would take a left turn after a few steps. Such information may help with human pose prediction over a long time horizon. Aiming for the follow-ahead task in a real-robot setting, we show how environment information can be used to predict long-horizon human poses and propose a real-time human pose prediction method that runs three times faster than the state-of-the-art methods and performs better or comparable.

Moreover, existing human pose forecasting datasets that contain environmental information are either limited to a single room region such as PROX [8] or gathered from synthetic data [9]. To overcome these issues, we built a robot (Fig. 2) with two cameras that can localize the robot and simultaneously record human motion in a building-scale space. We collect a realistic dataset (Real-IM: Real Indoor Motion Dataset) containing human motion with complete environmental information and multiple human motion styles. We train our human pose prediction model on both synthetic data [9] and on our new Real-IM dataset. We show in Sec. VII that our method outperforms the state-of-the-art methods for predicting both the trajectory and pose of the human. With the predicted human pose, we demonstrate that the proposed real-time algorithm enables the robot to follow a human in the front in real-world experiments.

From the robot follow-ahead perspective, the closest work is [3], which uses human pose prediction results to perform robot follow-ahead in an open space. However, they require

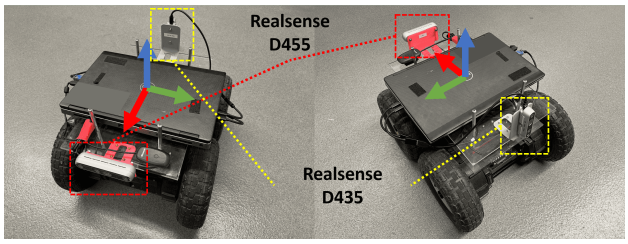


Fig. 2: **Robot**. Our mobile robot is assembled with two Realsense RGB-D cameras based on a Rover robot. We use the front camera to build the map for localization while navigating. The rear camera detects and tracks the human actor for 3D skeleton poses. The robot coordinate frame is also shown.

an extra third-person-view camera to localize the actor and are therefore constrained in a single-room region. Ours is the first work to demonstrate robot follow-ahead in a building-scale environment without relying on other off-board cameras and utilizing full 3D pose information.

Our contributions can be summarized as follows:

1) We present a new real-time method for human pose forecasting that considers the surrounding environment represented as occupancy maps.
2) We present a new building-scale real-world dataset for human pose prediction by building a robot system with dual cameras for localization and navigation.
3) We conduct experiments on synthetic and the proposed realistic datasets. Our method outperforms the baselines and state-of-the-art methods on both trajectory prediction and long-term human pose forecasting. Moreover, we show that our method is three times faster than existing methods, which allows us to perform robot follow-ahead in a real-world setup with a simple controller.

II. RELATED WORK

We summarize related work in two directions: robot follow-ahead and the human pose prediction.

A. Robot Follow-Ahead

The human follow-ahead problem was studied earlier in [1]. Kalman Filter [1] or Extended Kalman Filter [10], [11] are used to estimate the human trajectory. The Deep Reinforcement Learning method in [12] implicitly predicts the human trajectory by setting rewards. None of these works predicted full human poses. Recently, the work of [3] jointly predicts the human trajectory and human poses for the robot to follow ahead in an open space. However, an offboard third-person-view camera is still needed. In addition, human tractories are predicted in [13], [14] in Bird-Eye view (BEV). One recent work [15] predicts the human trajectory using the human skeleton poses. Our work focuses on human pose prediction for the robot's follow-ahead problem and proposes to plan the robot's path by forecasting the human pose given the environmental information.

B. Human Pose Prediction

Prediction of the human pose attracted increasing attention in the past few years [16], [17], [18], [19]. Researchers have managed to predict the human poses in an open space using architectures such as Multi-Layer Perception (MLP) [20], Generative Adversarial Networks (GANs) [21], [18], diffusion-based method [17], Graph Convolutional Networks (GCN) [22], or Transformers [23], [24], [25], [26].

Recent work started to incorporate environmental information with the human motion prediction task [9], [27], [4] and the human pose synthesis task [28], [6], [8], [29]. This body of literature relies on data generated with a simulation-based large-scale dataset [9] and in PROX [8] with a single-room realistic dataset. In order to deal with the human-environment interaction, some work explicitly predicts the goal and the path (or contact points) [9], [4], [27]. Methods include using conditional variational autoencoders (cVAE) [9], [4] or GANs [6]. Computationally expensive calculations such as voxelization [4] or contact map [27] are involved. Some work extract the environment features implicitly using PointNet [28], [29] or ResNet18 [6], [5]. Meanwhile, diffusion-based models have been proposed to generate human poses [7].

We build our work on [27]. We use a similar Gated Recurrent Unit (GRU) based network structure, but we focus on different scales of the human pose forecasting problem. By considering the interaction between the full 3D environment point cloud and the actor's joints, CA [27] addresses better on the complex human-environment interaction in a small space, such as sitting on a sofa or lying in bed. In contrast, we aim to forecast human poses for robot follow-ahead tasks. We focus on larger indoor scenes while the human is walking with less contact with objects. We improve and speed up the long-term trajectory prediction by conditioning it on a 2D occupancy map. In Sec. VII-B, we show that considering only 2D maps is faster than considering the full 3D geometry for this setup and can achieve a better forecasting performance in a long time horizon.

III. PROBLEM FORMULATION

We formulate the human pose prediction problem in a known environment as follows. We define a human pose with J joints as $\mathbf{x} \in \mathbb{R}^{J\times 3}$. The past N-step human motion is represented as $\mathbf{X}_{1:N} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{N\times J\times 3}$. All the human joints are represented in the latest human pose's coordinate frame. We use the human actor's torso (hip) keypoint's position $\mathbf{x}^a$ to represent the human trajectory in 2D. We denote the human's surrounding environment as $\mathbf{S}$. Given the human pose history $\mathbf{X}_{1:N}$ and the surrounding environment $\mathbf{S}$, we would like to forecast the future human poses $\mathbf{X}_{N+1:N+T}$ with our network $\mathcal{F}$ within time horizon T. That is:

$$\hat{\mathbf{X}}_{N+1:N+T} = \mathcal{F}(\mathbf{X}_{1:N}, \mathbf{S}) \tag{1}$$

Similar to existing works, we assume only one person is observed by the robot. We also assume the environment is entirely static and known to the robot. In Sec. VII, we investigate the performance of human pose prediction given a

fully known map, a limited field of view, and a fully unknown map.

IV. APPROACH

Given human history poses and surrounding environment, our method predicts the future poses in two steps similar to [9], [4]: first predict a human trajectory in the 2D environment, then complete the full-body pose based on the predicted trajectory. Note that the human heading direction can differ from the torso and the 2D trajectory. The motion planning module needs a full-body pose as input to calculate the viewing quality. Therefore, we predict full-body poses instead of a 2D trajectory for human motion. The following subsections introduce our representation of the environment and human poses. Then, we describe each component in our network architecture in detail.

A. Representation

Occupancy Map. We want to use environmental information to help with human motion prediction. We represent the environment with the local occupancy map $\mathbf{S}_t$ around the human pose $\mathbf{X}_t$. We clip the local occupancy map by distance d_x and d_y along x-axis and y-axis of the actor frame, respectively. We denote the resolution of the occupancy map as r.

We encode the human pose $\mathbf{X}_{1:N}$ with two more representations. A trajectory map and a local pose.

Trajectory and trajectory map. First, we extract the trajectory of the human based on the torso, i.e., $\mathbf{x}_1^a, \mathbf{x}_2^a, \ldots, \mathbf{x}_N^a$. We denote the 2D human trajectory path as $\mathbf{P}_{1:N} = (\mathbf{x}_1^a, \mathbf{x}_2^a, \ldots, \mathbf{x}_N^a) \in \mathbb{R}^{N\times 2}$. Note that all $\mathbf{x}_i^a$ are in the human pose's coordinate frame. Then, we encode them into a trajectory map. At each timestamp $\mathbf{x}_t^a = (x, y, z)$, we project $\mathbf{x}_t^a$ into the 2D map with the same resolution and center as the local occupancy map. We use a binary map as the ground truth for training. $I(u, v) = 1$ if $(u, v) = ((x, y) - (d_x, d_y))/r$. u, v is the pixel coordinate. We use a Gaussian distribution heatmap as the observation. The Gaussian has a center at the projected position with a predefined covariance σ. $I(u, v) = \mathcal{N}((u, v), \sigma)$. Thus, we represent the human torso trajectory with the trajectory map $\mathbf{M_{1:N}}$ by shape $H \times W \times N$, as shown in Fig. 3.

Local pose. For a human pose $\mathbf{x}$, we subtract the human pose by the torso position $\mathbf{x}^a$ to describe the human pose regardless of the position. We denote it as $\mathbf{x}^R = \mathbf{x} - \mathbf{x}^a$. And $\mathbf{X}_{1:N}^R = [\mathbf{x}_1^R, \mathbf{x}_2^R, \ldots, \mathbf{x}_N^R]$.

B. PathNet

We predict the human trajectory with the occupancy map and the human trajectory history based on a U-net [30] architecture similar to [13]. We concatenate the occupancy map of the last frame $\mathbf{S}_N$ and the human past trajectory map $\mathbf{M}_{1:N}$, and input it to an encoder $\mathbf{\Phi}$. We extract the bottleneck vector as the latent feature vector $\mathbf{z}$ and concatenate it with the past trajectory $\mathbf{P}_{1:N}$. We decode the feature vector with the decoder ψ to a probability trajectory map $\hat{\mathbf{M}}_{N+1:N+T}$ with shape $H \times W \times T$. We use soft-argmax [31], [32] to calculate the actor's position for each time stamp in the future.

$$\mathbf{z}_{traj} = \mathbf{\Phi}(\mathbf{S}_N, \mathbf{M}_{1:N}) \tag{2}$$

$$\hat{\mathbf{M}}_{N+1:N+T} = \mathbf{\Psi}(\mathbf{z}_{traj}, \mathbf{P}_{1:N}) \tag{3}$$

$$\hat{\mathbf{P}}_{N+1:N+T} = \text{soft}\arg\max(\hat{\mathbf{M}}_{N+1:N+T}) \tag{4}$$

We use multiple loss functions as training objectives to ensure the planning trajectory is accurate in the long term and collision-free. We define a trajectory loss $\mathcal{L}_{traj}$, a final position loss $\mathcal{L}_{final}$, and a trajectory map loss $\mathcal{L}_{map}$ and a collision loss $\mathcal{L}_{col}$.

$$\begin{aligned}
\mathcal{L}_{traj} &= \frac{1}{T}\sum_{t=N+1}^{N+T} \|\mathbf{P}_t - \hat{\mathbf{P}}_t\|_2 \\
\mathcal{L}_{final} &= \|\mathbf{P}_{N+T} - \hat{\mathbf{P}}_{N+T}\|_2 \\
\mathcal{L}_{map} &= \frac{1}{T}\sum_{t=N+1}^{N+T} \text{BCE}(\mathbf{M}_t, \hat{\mathbf{M}}_t) \\
\mathcal{L}_{col} &= \frac{1}{T}\sum_{t=N+1}^{N+T} (\hat{\mathbf{M}}_t \cdot |\mathbf{S}|)
\end{aligned} \tag{5}$$

BCE represents the Binary Cross-Entropy loss with weight w. The overall loss term is given by the sum of the loss terms with weights λ. In the experiment section, we empirically select $w = 40$, $\lambda_3 = 2$ and $\lambda_1 = \lambda_2 = \lambda_4 = 1.0$.

$$\mathcal{L} = \lambda_1\mathcal{L}_{traj} + \lambda_2\mathcal{L}_{final} + \lambda_3\mathcal{L}_{map} + \lambda_4\mathcal{L}_{col} \tag{6}$$

C. PoseNet

Given the trajectory prediction $\hat{\mathbf{P}}_{N+1:N+T}$ and human local pose history $\mathbf{X}_{1:N}^R$, we predict the human local pose $\hat{\mathbf{X}}_{N+1:N+T}^R$ with a Gated Recurrent Unit (GRU) network first. Then, we transform the local pose to the predicted position by each time stamp. We first encode the past local pose $\mathbf{X}_{1:N}^R$ and predicted trajectory $\hat{\mathbf{P}}_{N+1:N+T}$ separately with Multi-layer Perceptrons (MLPs). We concatenate and feed the latent feature $\mathbf{z}_{pose}$ into the GRU network $\mathbf{\Theta}$. We then decode the local pose recurrently by a GRU cell module $\mathbf{\Gamma}$ using the latent feature $\mathbf{z}_{pose}$ at each time stamp.

$$\begin{aligned}
&\mathbf{z}_{pose,N} = \mathbf{\Theta}(\mathbf{X}_{1:N}^R, \hat{\mathbf{P}}_{N+1:N+T}) \\
&\mathbf{z}_{pose,(t+1)}, \quad \hat{\mathbf{X}}_{t+1}^R = \mathbf{\Gamma}(\mathbf{z}_{pose,t}, \hat{\mathbf{P}}_t) \\
&\hat{\mathbf{X}}_t = \hat{\mathbf{X}}_t^R + \hat{\mathbf{P}}_t
\end{aligned} \tag{7}$$

The loss function $\mathcal{L}_{pose}$ is given in Eq. 8. We use the ground truth trajectory as the input during the training time and trajectory prediction from the PoseNet at inference time.

$$\mathcal{L}_{pose} = \frac{1}{TJ}\sum_{t=N+1}^{N+T}\sum_{J} \|\mathbf{X}_t^j - \hat{\mathbf{X}}_t^j\|_2 \tag{8}$$

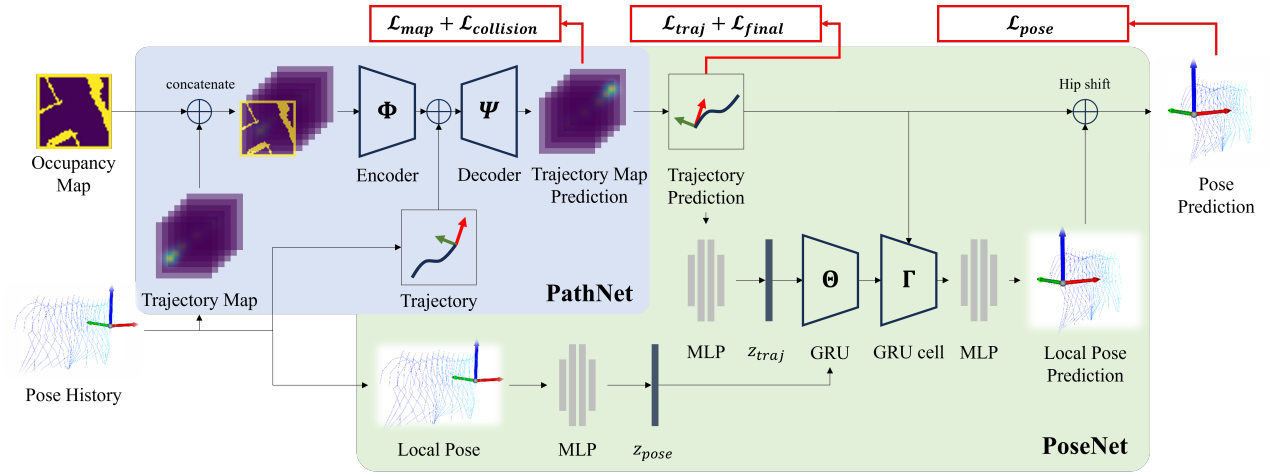


Fig. 3: **Network**. Our network has two parts. A PathNet to predict human trajectory, and a PoseNet to predict human future poses. The PathNet takes input from the occupancy map as well as the human trajectory and predicts the human future trajectory. The PoseNet uses the prediction results and local pose as input and predicts the future poses with a Gated Recurrent Unit (GRU) based network.

V. SYSTEM DESIGN

We first present the mobile robot system in Sec. V-A and then introduce the software architecture in Sec. V-B.

A. Robot

We build our robot system based on the Rover mobile robot [33]. As shown in Fig. 2, we install two Realsense cameras to perform navigation in the indoor environment and simultaneously track human pose. More specifically, we use a Realsense D455 as our front camera and use a Realsense D435 as the rear camera. We map and localize the robot with the front camera by combining visual odometry and IMU sensor reading. Our onboard computer is equipped with an Intel I7 processor, 32G RAM, and a P3200 Nvidia GPU.

B. Modules

Our mobile robot system consists of four modules built on the Robot Operation System (ROS) [34]. We briefly describe our system architecture in this section; additional details can be found in the supplementary material.

Mapping and localization. To create 2D and 3D maps, we use the RGB-D image and IMU data from the front camera. We apply the *rtabmap* package [35] to create visual odometry from color images and merge it with the IMU reading by using the *robot_localization* package [36]. The merged odometry is used for mapping the environment and for localization during navigation.

Human Pose Detection. We use *Yolo-v8* [37] to extract the human skeleton pose from the color images and set the detection frequency to 15Hz on our onboard computer. By combining the pose estimation result with the depth image and the localization result, we obtain the 3D skeleton pose and transform it into the global frame.

Human Pose Forecast. We run our human pose forecasting algorithm in real-time as in Sec. IV. We collect consecutive 3D poses during inference and transform them into the latest pose frame. In practice, we execute the forecast in 15-fps and set our predicting horizon as 3 seconds.

Navigation. We formulate the path planning task as a finite-horizon optimal control problem. We define our objective function at each time step based on the Pixels-Per-Area (PPA) metric [38]. PPA measures the viewing quality by considering the viewing distance and viewing angle. Given the predicted human motion, we calculate a sequence of robot controller inputs such that the total PPA cost overtime is maximized. We solve this optimal control problem using the Dynamic Programming (DP) [39] method and plan the robot's trajectory. A sequence of target robot poses is calculated and fed as input to the *move_base* package [40]. A local path is planned to avoid collision with the environment and is executed by sending velocity commands to the robot. We provide more details on this formulation in the supplementary material attached to our project website.

VI. DATASET

In addition to the standard synthetic dataset: GTA-IM [9], we are also interested in evaluating our method on a large-scale real-world dataset. However, existing realistic datasets, such as PROX [8], are limited to single-room areas. Therefore, we collect and present the Real Indoor Motion (Real-IM) dataset for the large-scale human follow-ahead. We simultaneously capture the entire human body and environment in building-scale spaces using the robot described in Sec. V-A. Compared to the existing synthetic dataset (GTA-IM), our dataset contains more movement patterns, including walking, crab moving, and varying moving speeds.

We collect 12 sequences from 5 different building halls. Each sequence contains an approximately four-minute movement. The dataset includes sequences with different lighting conditions recorded at different times of day. We invited multiple actors with different walking styles and genders. We provide the raw ROS bags and pre-processed data for direct use. In the supplementary material from our webpage, we present a few sample images from our dataset visualized in Rviz [34].

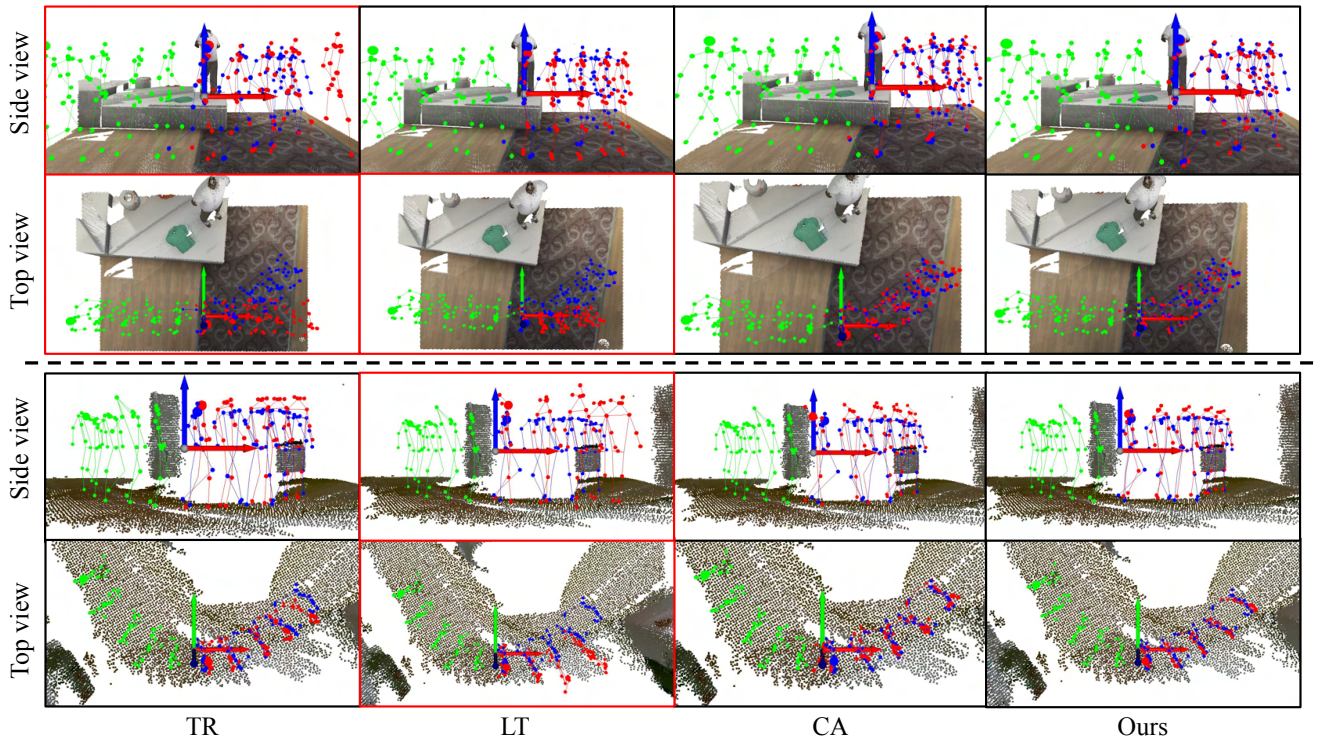


Fig. 4: **Qualitative results of the human pose forecasting.** Each column corresponds to a different method. Human pose history, prediction results, and ground truth are shown in green, red, and blue. TR [41] and LT [9] fail to predict the turn (red outline), whereas CA [27] and ours are successful in correctly predicting it (black outline).

VII. EXPERIMENTS

In this section, we present results from experiments to compare our human pose prediction algorithm with existing works. We introduce the experiment setup (Sec. VII-A), results (Sec. VII-B), and analysis (Sec. VII-C). We will then demonstrate the robot follow-ahead task with our algorithm and justify the need for our human pose prediction for this task (Sec. VII-D).

A. Experiment setup

Baselines. We include two human pose forecasting baselines: CA [27] and LT [9]. Similar to our setting, these methods utilize environmental information when predicting 3D human pose. LT [9] is the first work to forecast human poses by sequentially predicting a trajectory and estimating the joint poses. Similar to our network architecture, CA [27] applies GRU for human pose prediction. As mentioned in the CA [27] paper, CA outperforms LTD [22], DMGNN [42], and SLT [29]. Therefore, we do not include these methods in our experiments. In addition to CA and LT, we implement a pure Transformer-based method (indicated as TR) by taking the history of human poses as input and output predicted future poses.

Metrics. We use Mean Per Joint Position Error (**MPJPE**) to measure our performance in predicting the pose and the trajectory, which is a standard and widely used metric in human pose prediction [16]. MPJPE provides a quantitative measure of how close the predicted joint positions are to the true joint positions, averaged across all joints. A lower MPJPE indicates a better model performance. As in [27] and [9], we report hip position errors as the global transition error (path error) and local 3D pose errors in millimeters (mm).

Implementation details. We sequentially train our PathNet and PoseNet for 1200 epochs each using Adam [43] optimizer implemented in Pytorch [44]. The learning rate is set to 0.001 for the PathNet and $1e-5$ for the PoseNet. The learning rate scheduler has a gamma of 0.1 and a step size of 600. On the GTA-IM dataset, we set the trajectory map and occupancy map size as 40×40 and use them to represent $5m^2$ local space. We choose the same training and testing set as [27]. On the Real-IM dataset, we use a forecasting horizon of 3s and predicted occupancy map and trajectory map of size $8m^2$.

B. Human Pose Prediction

We compare our human pose prediction network with the baselines on both synthetic and the proposed real-world datasets. As shown in Table I and Table II, our method outperforms the baselines on both trajectory prediction and human pose forecasting. We improve path prediction significantly by providing additional map information beyond visibility in first-person view. On the pose prediction side, our method achieves lower error after the first second, which indicates that CA [27] is more accurate for short-term prediction, while ours is better for longer forecasting horizons, taking advantage of the trajectory prediction accuracy.

Inference time. In addition to accuracy, we compare inference speed on our onboard computer across all methods.

TABLE I: **Evaluation results in GTA-IM dataset**. We use the same training and testing set as CA [27] and use their reported numbers. Results show that our methods outperform the state-of-the-art methods. The results also show that by providing additional map information, trajectories are predicted significantly better, which yields lower pose errors.

	Path error (mm)				**3D Pose error (mm)**				
Method \ **Time (sec)**	0.5	1.0	1.5	2.0	0.5	1.0	1.5	2.0	mean (2s)
TR [41]	113.7	187.4	375.8	471.2	112.4	116.4	129.3	139.8	118.3
LT [9]	104	163	219	297	91	158	237	328	173
CA [27]	58.0	103.2	154.9	221.7	**50.8**	67.5	75.5	86.9	**61.4**
Ours (PathNet+GRU)	**52.5**	**99.6**	**107.3**	**113.8**	51.1	**63.6**	**70.7**	**75.0**	62.9
PathNet (partial)	123.8	149.3	237.0	290.9	72.9	83.5	114.7	126.9	110.4
PathNet (unknown)	120.5	145.1	232.6	286.1	72.4	82.8	105.5	118.0	83.5
Ours w/o $\mathcal{L}_{col}$	112.8	153.6	243.4	292.0	71.6	81.0	100.6	110.6	81.0
Ours w/o $\mathcal{L}_{map}$	132.0	184.4	320.1	365.1	73.9	86.3	129.6	146.1	93.3

TABLE II: **Evaluation results in Real-IM dataset**. We report results across both 2s and 3s prediction horizons. Our method performs better than the baselines on both path and pose prediction. As expected, the error increases along with the horizon for all methods.

	Path error (mm)						**3D Pose error (mm)**					
Method \ **Time (sec)**	1.0	1.5	2.0	3.0	mean (2s)	mean (3s)	1.0	1.5	2.0	3.0	mean (2s)	mean (3s)
TR [41]	150.5	249.1	295.7	439.1	154.3	240.3	89.9	91.8	93.8	101.2	89.8	92.4
CA [27]	153.9	255.7	257.6	374.3	253.2	257.4	**67.9**	80.3	86.4	109.2	69.7	80.7
Ours (PathNet+TR)	**145.8**	**165.7**	**186.9**	**240.2**	**150.9**	**178.4**	96.8	97.0	98.1	98.4	96.5	97.1
Ours (PathNet+GRU)							69.9	**75.5**	**78.1**	**84.5**	70.7	**75.0**
PathNet (partial)	202.0	319.5	372.4	511.5	251.9	350.7	80.9	88.6	92.7	100.8	83.2	92.9
PathNet (unknown)	183.3	316.7	390.9	583.2	192.7	308.8	69.3	76.6	81.2	101.9	**69.3**	76.5
Ours w/o $\mathcal{L}_{col}$	204.4	449.6	341.0	493.2	418.0	440.6	101.5	102.3	95.2	115.0	99.8	107.0
Ours w/o $\mathcal{L}_{map}$	162.1	271.8	441.4	763.3	171.9	316.1	68.5	80.4	92.9	132.0	69.8	90.5

Since CA [27] considers pair-wise point features between every 3D point and human joint across multiple time frames, it takes an average of $100.13ms$ to predict the human poses. In contrast, considering only 2D maps, our method has an average inference time of $32.12ms$, allowing us to respond faster in real-world applications.

C. Ablation Study

In this section, we investigate the impact of different components in our network on human pose prediction.

Map visibility. One of our key contributions is to provide the local environment map to our PathNet. Therefore, we study how different map visibility can affect the pose prediction accuracy. We compare our results, which utilize a fully known local map, with two baselines: 1) without access to the map and 2) with a map whose visible area is limited to the camera FOV, as if the robot is operating in a new environment. Results from Table I and II show that map information does play a critical role in improving prediction accuracy.

GRU vs. Transformers. Transformers [41] have been shown to outperform GRUs when there's a sufficient amount of data [24]. In this section, we compare the performance of these two modules by replacing GRU with Transformers in the PoseNet module. As shown in Table II, we observe that GRU is outperforming Transformers on the 3D pose prediction task. One possible reason is that our training set is not large enough, and GRU is sufficient in this case.

Loss terms. During the training process, we expect that by using $\mathcal{L}_{col}$ and $\mathcal{L}_{map}$, the predicted trajectory would avoid collision with the environment. As shown in Table I and II, we conduct an ablation study to investigate whether these loss terms are helpful. Results show that these collision loss terms do increase the accuracy of both short-term and long-term trajectory prediction.

D. Robot Follow-ahead

In this section, we investigate the performance of our algorithm combined with the robot system on the robot follow-ahead task. As shown in Fig. 5, we compare the planned robot path using the predicted trajectory against using the ground truth. We present additional qualitative results in the supplementary video.

Moreover, we investigate whether our human pose prediction helps with the robot follow-ahead task. We compare our controller, with the human pose prediction as the input, against two other controlling strategies: (1) a greedy approach implemented with Extended Kalman Filter (EKF) for estimating human position: we do not apply any future pose prediction. Instead, we use a uniform Gaussian distribution to propagate a future trajectory from the past human trajectory. We demonstrate the performance of a

myopic controller that directly reacts to the EKF output. (2) a Dynamic Programming-based optimal controller that is given a ground-truth future trajectory as part of the input (DP + g.t. traj.): This oracle algorithm serves as an upper-bound controller of this task.

We evaluate the robot's follow-ahead performance using the following metrics. 1) **Area**. Percentage of pixels in images that the actor occupies. 2) **Tracking time**. The number of frames in which the actor is detected in the image is divided by the total number of frames in a sequence. 3) **Distance**. The distance between the center of the human bounding box and the center of the image is normalized by the image width.

Experiment results are shown in Table III. Even though our algorithm does not have access to the future trajectory of the human, with its predictive capabilities, its performance is better than the myopic controller and is within 85% of this oracle-based upper-bound (Table III). The result shows that the robot follow-ahead performance can benefit from the human pose forecasting predictions.

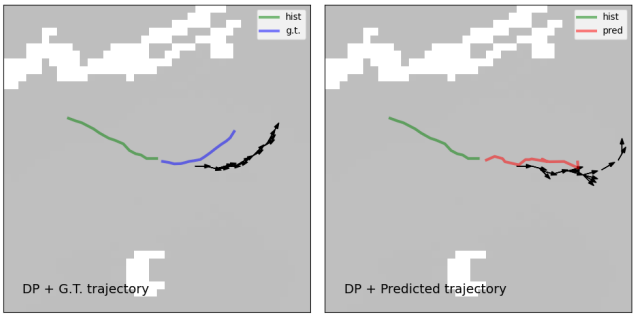


Fig. 5: **Robot Follow-ahead**. We visualize the planned path based on the human pose predictions. The human trajectory history, trajectory prediction, and the ground truth are shown in green, red, and blue. We visualize the planned robot path in arrows for each time step. We visualize the map as the background, white for obstacles and gray for free space.

TABLE III: **Follow-Ahead Evaluation.**

Method	Area ↑	Tracking Time↑	Distance↓
LB: EKF	0.241	0.67	0.147
Ours: DP + pred. traj.	0.302	0.85	0.151
UB: DP + g.t. traj.	**0.358**	**1.00**	**0.145**

VIII. CONCLUSION

This paper presented a method to predict human poses in indoor environments in order to accomplish the robot follow-ahead task. We proposed an architecture that first predicts the 2D human trajectory based on the occupancy map and then predicts the 3D human poses conditioned on the 2D trajectory. To validate our approach, we built a mobile robot system and collected a building-scale Real Indoor Motion Dataset (Real-IM) for human pose forecasting problems in large and complex environments. Through both synthetic and realistic experiments, we showed that our approach outperforms baselines and state-of-the-art methods on trajectory prediction and long-term human pose forecasting. In terms of run-time, it is three times faster. We also demonstrated successful robot follow-ahead by forecasting human poses in real-time.

Our system has its limitations. In general, most failure cases of the robot follow-ahead task stem from the rear camera losing track of the actor. Consequently, the robot is unable to locate the actor and to move to a position such that the actor remains within sight. Moreover, our human pose prediction method does not guarantee consistency across time and may lead to jerky motion under challenging scenarios, such as T-junctions. We provide some examples of these failures in the supplementary video. Meanwhile, we highlight that the planning algorithm for this robot follow-ahead problem can be further developed to incorporate the rear camera's limited field-of-view (FOV) and constrained robot kinematics. Additionally, relaxing the assumption of whole map visibility by using only a partial map can potentially allow our method to generalize to a new environment. We plan to address these challenges in our future work.

IX. ACKNOWLEDGEMENT

The authors thank all Robotic Sensor Network Lab members for the helpful discussions. This work is supported by the NSF NRI Grant #2022894.

REFERENCES

[1] Daniel M Ho, Jwu-Sheng Hu, and Jyun-Ji Wang. Behavior control of the mobile robot for accompanying in front of a human. In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 377–382. IEEE, 2012.

[2] Nikhil Karnad and Volkan Isler. Modeling human motion patterns for multi-robot planning. In *2012 IEEE International Conference on Robotics and Automation*, pages 3161–3166. IEEE, 2012.

[3] Mohammad Mahdavian, Payam Nikdel, Mahdi TaherAhmadi, and Mo Chen. STPOTR: Simultaneous Human Trajectory and Pose Prediction Using a Non-Autoregressive Transformer for Robot Following Ahead, September 2022. arXiv:2209.07600 [cs].

[4] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic Scene-Aware Motion Prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11354–11364, Montreal, QC, Canada, October 2021. IEEE.

[5] Zhe Wang, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes. Geometric Pose Affordance: 3D Human Pose with Scene Constraints, December 2021. arXiv:1905.07718 [cs].

[6] Jingbo Wang, Sijie Yan, Bo Dai, and Dahua Lin. Scene-aware Generative Network for Human Motion Synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12201–12210, Nashville, TN, USA, June 2021. IEEE.

[7] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023.

[8] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D Human Pose Ambiguities with 3D Scene Constraints, August 2019. arXiv:1908.06963 [cs].

[9] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term Human Motion Prediction with Scene Context, July 2020. arXiv:2007.03672 [cs].

[10] Payam Nikdel, Rakesh Shrestha, and Richard Vaughan. The Hands-Free Push-Cart: Autonomous Following in Front by Predicting User Trajectory Around Obstacles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4548–4554, May 2018. ISSN: 2577-087X.

[11] Yoonseon Oh, Sungjoon Choi, and Songhwai Oh. Chance-constrained target tracking for mobile robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 409–414, Seattle, WA, USA, May 2015. IEEE.

[12] Payam Nikdel, Richard Vaughan, and Mo Chen. LBGP: Learning Based Goal Planning for Autonomous Following in Front. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3140–3146, May 2021. ISSN: 2577-087X.

[13] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From Goals, Waypoints & Paths To Long Term Human Trajectory Forecasting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15213–15222, Montreal, QC, Canada, October 2021. IEEE.

[14] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneider. Uncertainty-aware Short-term Motion Prediction of Traffic Actors for Autonomous Driving, March 2020. arXiv:1808.05819 [cs, stat].

[15] Tim Salzmann, Hao-Tien Lewis Chiang, Markus Ryll, Dorsa Sadigh, Carolina Parada, and Alex Bewley. Robots That Can See: Leveraging Human Pose for Trajectory Prediction. *IEEE Robotics and Automation Letters*, pages 1–8, 2023. Conference Name: IEEE Robotics and Automation Letters.

[16] Kedi Lyu, Haipeng Chen, Zhenguang Liu, Beiqi Zhang, and Ruili Wang. 3D Human Motion Prediction: A Survey, March 2022. arXiv:2203.01593 [cs].

[17] Saeed Saadatnejad, Ali Rasekh, Mohammadreza Mofayezi, Yasamin Medghalchi, Sara Rajabzadeh, Taylor Mordan, and Alexandre Alahi. A generic diffusion-based approach for 3D human pose prediction in the wild, March 2023. arXiv:2210.05669 [cs].

[18] Payam Nikdel, Mohammad Mahdavian, and Mo Chen. DMMGAN: Diverse Multi Motion Prediction of 3D Human Joints using Attention-Based Generative Adversarial Network. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9938–9944, May 2023.

[19] Ye Yuan and Kris Kitani. DLow: Diversifying Latent Flows for Diverse Human Motion Prediction, July 2020. arXiv:2003.08386 [cs, eess].

[20] Arij Bouazizi, Adrian Holzbock, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. MotionMixer: MLP-based 3D Human Body Pose Forecasting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 791–798, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization.

[21] Baptiste Chopin, Naima Otberdout, Mohamed Daoudi, and Angela Bartolo. 3D Skeleton-based Human Motion Prediction with Manifold-Aware GAN, March 2022. arXiv:2203.00736 [cs].

[22] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning Trajectory Dependencies for Human Motion Prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9488–9496, Seoul, Korea (South), October 2019. IEEE.

[23] Lujing Chen, Rui Liu, Xin Yang, Dongsheng Zhou, Qiang Zhang, and Xiaopeng Wei. STTG-net: a Spatio-temporal network for human motion prediction based on transformer and graph convolution network. *Visual Computing for Industry, Biomedicine, and Art*, 5(1):19, December 2022.

[24] Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. PoseGPT: Quantization-based 3D Human Motion Generation and Forecasting, October 2022. arXiv:2210.10542 [cs].

[25] Angel Martinez-Gonzalez, Michael Villamizar, and Jean-Marc Odobez. Pose Transformers (POTR): Human Motion Prediction with Non-Autoregressive Transformers. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2276–2284, Montreal, BC, Canada, October 2021. IEEE.

[26] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A Spatio-temporal Transformer for 3D Human Motion Prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574, December 2021. ISSN: 2475-7888.

[27] Wei Mao, Richard I Hartley, Mathieu Salzmann, and others. Contact-aware human motion forecasting. *Advances in Neural Information Processing Systems*, 35:7356–7367, 2022.

[28] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. Towards Diverse and Natural Scene-aware 3D Human Motion Synthesis. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20428–20437, New Orleans, LA, USA, June 2022. IEEE.

[29] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing Long-Term 3D Human Motion and Interaction in 3D Scenes, June 2021. arXiv:2012.05522 [cs].

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[31] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral Human Pose Regression, September 2018. arXiv:1711.08229 [cs].

[32] Diogo C. Luvizon, Hedi Tabia, and David Picard. Human Pose Regression by Combining Indirect Part Detection and Contextual Information, October 2017. arXiv:1710.02322 [cs].

[33] Rover Robotics. Rover Robotics, 2023.

[34] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, and others. ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009. Issue: 3.2.

[35] Mathieu Labbé and François Michaud. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21831.

[36] T. Moore and D. Stouch. A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[37] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023.

[38] Qingyuan Jiang and Volkan Isler. Onboard View Planning of a Flying Camera for High Fidelity 3D Reconstruction of a Moving Actor, 2023. _eprint: 2308.00134.

[39] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966. Publisher: American Association for the Advancement of Science.

[40] Eitan Marder-Eppstein, Eric Berger, Tully Foote, Brian Gerkey, and Kurt Konolige. The Office Marathon: Robust Navigation in an Indoor Office Environment. In *International Conference on Robotics and Automation*, 2010.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, {\L}ukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[42] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic Multiscale Graph Neural Networks for 3D Skeleton Based Human Motion Prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 211–220, Seattle, WA, USA, June 2020. IEEE.

[43] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs].

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *CoRR*, abs/1912.01703, 2019. arXiv: 1912.01703.