

# Following Ahead Companion Robot

by

**Payam Nikdel**

M.Sc., Simon Fraser University, 2018

B.Sc., Shiraz University, 2015

Thesis Submitted in Partial Fulfillment of the  
Requirements for the Degree of  
Doctor of Philosophy

in the  
School of Computing Science  
Faculty of Applied Sciences

© **Payam Nikdel 2023**  
**SIMON FRASER UNIVERSITY**  
**Spring 2023**

Copyright in this work is held by the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.

# Declaration of Committee

**Name:** Payam Nikdel  
**Degree:** Doctor of Philosophy  
**Thesis title:** Following Ahead Companion Robot  
**Committee:** **Chair:** Manolis Savva  
Assistant Professor, Computing Science

**Mo Chen**  
Supervisor  
Assistant Professor, Computing Science

**Angelica Lim**  
Committee Member  
Assistant Professor, Computing Science

**Jason Peng**  
Examiner  
Assistant Professor, Computing Science

**Steven Waslander**  
External Examiner  
Professor, Institute for Aerospace Studies  
University of Toronto

# Abstract

Nowadays, most intelligent systems rely on interacting with humans. Two main functionalities of such systems are the ability to follow their users and to predict their future motions. This thesis develops robust methods for a companion robot that can follow humans and predict their motions in the future.

Predicting plausible human motion is one of the most critical and challenging parts of human-robot interaction (HRI) applications. We can categorize human motion prediction into probabilistic or deterministic approaches. The probabilistic approach tries to model the multi-modality of human motion; in contrast, the deterministic approach has one output per observation. In this thesis, we design two human motion prediction methods. One of them utilizes the multimodality of human motion for accurate predictions, while the other one is deterministic and fast.

Additionally, we design two human-following methods one based on reinforcement learning and the other using a human motion prediction model. The first work investigates a hybrid solution that combines deep reinforcement learning (RL) and classical trajectory planning for the following in-front application. As for the second method, we design a general human-following system with a fast non-autoregressive human motion prediction model.

**Keywords:** 3D human motion prediction; Human following; Companion Robot; Reinforcement Learning

# Dedication

*To my beloved father, Mehrdad Nikdel, and my dear mother, Forouzande Emami, whose endless affection, encouragement, and mentorship have been the driving force behind my personal and professional growth.*

# Acknowledgements

I want to express my deepest gratitude to everyone who has contributed to this thesis. First and foremost, I would like to thank my loving wife for her unwavering support and encouragement throughout my academic journey. I could not have accomplished this without her.

I am also thankful to my supervisor, Dr. Mo Chen, for his invaluable guidance, expertise, and motivation. His feedback was crucial in shaping my research and writing. Furthermore, Dr. Angelica Lim's mentorship was incredibly helpful in my growth as a researcher.

The contributions of my colleagues, especially Mohammad and Salar, deserve acknowledgment for providing valuable insights and assistance during my research. Their support and encouragement were critical to my success.

Lastly, I extend my heartfelt appreciation to my family, friends, and everyone who has supported me throughout this journey. Your encouragement and support have been a driving force behind my achievements, and I am deeply grateful for your presence in my life.

# Table of Contents

Declaration of Committee	ii
Abstract	iii
Dedication	iv
Acknowledgements	v
Table of Contents	vi
List of Tables	viii
List of Figures	ix
<b>1 Introduction</b>	<b>1</b>
<b>2 Human 3D Motion Prediction</b>	<b>3</b>
2.1 Related Work . . . . .	5
2.1.1 Observed sequence . . . . .	5
2.1.2 Human motion laws . . . . .	6
2.1.3 Model and Prediction Results . . . . .	8
2.1.4 Human Motion in Animation . . . . .	13
2.1.5 Datasets . . . . .	14
2.2 DMMGAN: Diverse Multi Motion prediction of 3D Human Joints using Attention-Based Generative Adversarial Network . . . . .	15
2.2.1 Problem Setup . . . . .	16
2.2.2 Method . . . . .	16
2.2.3 Experiments and Results . . . . .	21
2.2.4 Conclusion . . . . .	25
2.3 STPOTR: Simultaneous Prediction of Human Trajectory and Pose with Transformers . . . . .	26
2.3.1 Methodology . . . . .	28
2.3.2 Human Motion Prediction Experiments . . . . .	30

2.3.3	Conclusion . . . . .	33
2.4	Conclusion . . . . .	34
<b>3</b>	<b>Human Following</b>	<b>35</b>
3.1	Related Work . . . . .	37
3.1.1	Categorize of human following . . . . .	37
3.1.2	Human following using RL . . . . .	41
3.2	LBGP: Learning Based Goal Planning Approach for Autonomous Following in Front . . . . .	42
3.2.1	Problem Setup . . . . .	43
3.2.2	Method . . . . .	44
3.2.3	Simulated Experiments . . . . .	46
3.2.4	Real World Experiments . . . . .	50
3.2.5	Discussion . . . . .	54
3.2.6	Conclusion . . . . .	55
3.3	Human Following using STPOTR . . . . .	56
3.3.1	Robot Follow-Ahead via Human Motion Predictions . . . . .	57
3.3.2	Real World Experimental Results . . . . .	57
3.3.3	Conclusion . . . . .	59
3.4	Conclusion . . . . .	60
<b>4</b>	<b>Future work</b>	<b>61</b>
	<b>Bibliography</b>	<b>62</b>

# List of Tables

Table 2.1	Comparison of our systems versus two baselines for the 3D Pose experiment. . . . .	23
Table 2.2	Comparison of our systems versus two baselines for the full 3D motion experiment. . . . .	24
Table 2.3	Analytical comparisons between our developed model and the baselines introduced in [97] and [69] in terms of <i>ADE</i> and <i>FDE</i> for both human pose and trajectory predictions and Inference Duration (ID) . . . . .	30
Table 2.4	Our ablation study analytical comparisons . . . . .	33
Table 3.1	Comparison of our systems versus two baselines for all simulation trajectories. . . . .	48
Table 3.2	Comparison of our systems versus two baselines for <i>Straight</i> trajectory.	52
Table 3.3	Comparison of our systems versus two baselines for <i>S shape</i> trajectory.	53
Table 3.4	Comparison of our systems versus two baselines for <i>U-turn</i> trajectory.	54
Table 3.5	Robot follow-ahead comparative results for three tested scenarios. . .	58



# List of Figures

Figure 2.1	3D human motion prediction task. . . . .	5
Figure 2.2	<b>Left:</b> original human skeleton, <b>Middle:</b> local graph, a graph with learnable connections initialized from the original human skeleton [16], <b>right:</b> global graph with learnable connections for right knee joint [16]. . . . .	7
Figure 2.3	According to the type of prediction result, human motion prediction can be categorized as probabilistic or deterministic types. . . . .	8
Figure 2.4	Two basic types of RNN prediction methods: stack [20] and stream [59] structures. . . . .	9
Figure 2.5	Overview of POTR approach. POTR is a non-autoregressive human motion prediction method. For each pose in the input sequence, a network computes embeddings. Then, in parallel, the Transformer processes the sequence and decodes the attention embeddings. Lastly, A residual network predicts the sequence [60]. . . . .	11
Figure 2.6	HP-GAN, predicting multiple future poses from a single input sequence by feeding vector $z$ [8]. . . . .	12
Figure 2.7	DLow samples (stars) cover more modes (colored ellipses) in the latent space of a conditional variational autoencoder (CVAE) than CVAE samples. Because of this coverage, DLow can output a diverse future prediction in comparison to normal CVAE [97]. . . . .	12
Figure 2.8	Given a sequence of 3D human motions, our system generates a diverse set of future motions. The <i>3D pose prediction</i> module generates diverse 3D poses while <i>hip prediction</i> module estimates the human trajectory together forming a 3D human motion. The <i>discriminator</i> module distinguishes a real 3D human motion from a generated one. . . . .	15

Figure 2.9	System overview: Given a sequence of 3D human motion, our method generates $N$ future sequences of human 3D motion using a discriminator and four loss functions. Our system consists of three main parts. The first part is predicting the human 3D pose ( <i>3D Pose module</i> ) by receiving a history of the human 3D pose. The second part is the <i>Hip Prediction</i> module (more details in Fig. 2.10) which predicts the future position of the hip joint for each of the predicted human 3D poses. Finally, the discriminator module learns the distribution of the Human 3.6M dataset by distinguishing between generated and real data. The system uses the discriminator loss to generate sequences similar to the dataset distribution while using four supervised loss functions to promote accuracy and diversity. See Fig. 2.10 for Transformer Encoder architecture. . . . .	17
Figure 2.10	a) The Transformer Encoder [86] and b) the <i>Hip Prediction</i> module architectures. The <i>Hip Prediction</i> module, estimates the hip joint positions of each predicted 3D pose by receiving the history of the hip movements and the motion predicted by the <i>3D Pose</i> module. . . . .	18
Figure 2.11	Qualitative results of 3D pose predictions comparing our method, DMMGAN, to DLow in terms of diversity. . . . .	23
Figure 2.12	Qualitative results of 3D motion predictions comparing our method, to DLow in terms of diversity. . . . .	24
Figure 2.13	Robot follow-ahead via human motion prediction . . . . .	26
Figure 2.14	Our model structure. It simultaneously predicts human poses and trajectories from an observed 3D human joints sequence. It is constructed from two non-autoregressive transformers for pose and trajectory predictions as well as a Shared Attention module to share knowledge between the two for better predictions. An End Attention module is added to the end of each decoder for better modeling of the temporal dependencies. The blue-colored frames show the input sequence or frame and the red ones show the output. The rectangular frames show that the same frame (last input pose) is copied and used as the decoder input sequence and as a residual for decoder output. . . . .	28
Figure 2.15	Three samples of the predicted motion vs. ground truth. On each couple of figures (a to c) the left one shows the predicted motion given an observed sequence and the right one shows the ground truth. The blue-colored skeletons show the input sequence and the red and green ones show the model predictions and ground truth, respectively. Also, the trajectory of the hip is shown with dashed black lines. . . . .	32

Figure 3.1	Depending on the robot’s local position relative to the human, the person following can be categorized into (a) “behind the leader”, (b) “side-by-side”, (c) “in front of the leader” [29]. . . . .	37
Figure 3.2	In [70], via an EKF approach and hand-designed human models, they designed a following ahead system. . . . .	38
Figure 3.3	An intersection with two people walking side-by-side. a) going straight, b) leader guiding, c) inferring routes [40]. . . . .	40
Figure 3.4	A mobile robot following-ahead of a user. The robot must predict the user’s trajectory to stay in the correct relative position. In each time step in our proposed approach, the robot considers previous states of the joint system to generate a goal (blue dot). Then a trajectory planner navigates the robot towards the goal (green line). . . . .	42
Figure 3.5	Our relative coordinates system . . . . .	43
Figure 3.6	Reward based on the robot’s relative position to the person. Increasing from black (-1) to white (+1). . . . .	45
Figure 3.7	Visualization of person motion model. From left to right: moving straight, in different circles, in smoothed curves and using annotated simulated path of a human. . . . .	46
Figure 3.8	Visualize the trajectory of robot (arrows) and human (triangle) during the simulated trajectory experiment for our system (LBGP) and two baselines, HC and E2E. . . . .	50
Figure 3.9	Discounted cumulative rewards during training averaged over five runs for LBGP with or without curriculum and E2E. The shaded area represents half a standard deviation. . . . .	51
Figure 3.10	real world Examples: the robot (in arrows) and user (in triangles) trajectories is depicted. Row 1 and 2, <i>S shape</i> experiment in <i>ahead-right</i> and <i>behind</i> settings. Row 3 and 4: <i>U-turn</i> experiment in <i>ahead-left</i> and <i>behind</i> settings. . . . .	55
Figure 3.11	Robot follow-ahead via human motion prediction in the <i>U-Shaped</i> scenario using the STPOTR model. Opacity increases with time. . . . .	56
Figure 3.12	<i>Sit-to-Stand</i> (A) and <i>Stand-to-Sit</i> (B) use cases of STPOTR. Opacity increases with time. . . . .	58
Figure 3.13	Three samples of the robot follow-ahead tasks for U-Shaped, S-Shaped and straight line scenarios. The triangle and arrows show the human and robot motions, respectively. . . . .	59

# Chapter 1

## Introduction

The development of companion robots has revolutionized the field of human-robot interaction (HRI). These robots are designed to interact with humans naturally and intuitively, providing assistance and entertainment. Two key features of these companion robots are human motion prediction and human following, which allow them to react to human movements and engage in meaningful interactions.

3D human motion prediction has two main branches: human pose prediction and human trajectory prediction. Human pose prediction involves determining the relative position of each body joint with respect to the hip joint, while human trajectory prediction refers to the path of the hip joint as the body moves in 3D space.

Human motion prediction with respect to its outputs can be categorized into probabilistic and deterministic approaches. The probabilistic approach considers multiple possible future sequences for a given observed sequence by considering the multi-modality of human motion. This approach is favored in robotic applications as it provides more options and adaptability. However, it also poses a more complex optimization challenge compared to deterministic methods. On the other hand, deterministic methods focus on predicting a single, highly accurate future sequence based on an observed sequence. Although this approach is simpler to optimize, it does not take into account the diverse and multi-modal nature of human behavior, which can limit its practicality in real-world applications.

This thesis investigates human motion prediction and the development of robust human-following strategies for companion robots. Current human motion prediction methods often concentrate on predicting either human pose or human trajectory, which may need to be revised for robotics applications. To address these limitations, we present two human motion prediction methods: one that leverages the multi-modal nature of the human motion to provide multiple precise predictions and another that prioritizes determinism and speed for real-time robotics applications. Both methods predict human motion, including 3D pose and trajectory, which is crucial for various robotics applications. We present these works in two publications at the International Conference on Robotics and Automation (ICRA) [56, 69].

Additionally, we design two human-following methods, one based on reinforcement learning and the other using a human motion prediction model. The first method combines deep reinforcement learning (RL) and classical trajectory planning for a following-in-front application. The second method is a general human-following system that uses a fast non-autoregressive human motion prediction model.

## Chapter 2

# Human 3D Motion Prediction

Human motion prediction is the task of predicting the future movements of a user based on the observed past motion. Human motion prediction can be used in various applications, such as robotics, gaming, and animation. The goal is to model a human’s complex, multi-dimensional motion patterns and use that information to predict future movements. This is typically done using deep learning algorithms, such as Recurrent Neural Networks (RNNs) or Generative Adversarial Networks (GANs), which are trained on large datasets of human motion data.

The input to a human motion prediction model can be 2D or 3D data, for instance, video footage or motion capture data, and the output is one or multiple predictions of the future motion of the human. The accuracy of the prediction depends on factors such as the quality of the training data, the choice of algorithm, and the computational resources available.

An essential ability of an intelligent system interacting with humans is to estimate plausible human body poses and trajectories in 3D space. The advancements in artificial intelligence have led to numerous industrial applications for such algorithms in areas such as human-robot interactions (HRI), autonomous driving, and visual surveillance [19, 26, 76]. In particular, precise prediction of human 3D body motion is crucial for various robotic applications, including robot navigation and crowd control [12, 71]. Although there have been significant advances in deep learning, predicting human motion accurately remains challenging due to the complexity of human behavior. Some researchers focus on predicting only human trajectory [2, 100]. Among these, Agand et al. [2] developed a probabilistic and optimal approach for human navigational intent inference. Although these approaches are useful for some applications, they tend to lose important information, such as the strong correlation between human trajectory and the movement of other body parts.

In this chapter, we first conduct a comprehensive review of existing works in the field of human motion prediction by categorizing different methods for representing human motion into probabilistic and deterministic approaches. Then we propose two innovative solutions for human motion prediction in robotics applications.

The first solution, referred to as DMMGAN (Diverse Multi Motion prediction of 3D human joints using attention-based Generative Adversarial Network), involves the development of a deep generative architecture that predicts a diverse set of possible human body motions. The model utilizes a transformer-based encoder and a GRU combined with a GAN to provide multiple, accurate predictions for the 3D human trajectory and pose. This method offers a real-time solution for diverse human motion prediction that has the potential use case in robotics and autonomous car applications. We explore this solution in Section 2.2.

The second solution, STPOTR (Simultaneous Prediction of Human Trajectory and Pose with Transformers), is a non-autoregressive transformer model that simultaneously predicts both human trajectory and body pose. It is designed to be fast and accurate and demonstrates better performance compared to previous works for a robotics application. The model uses the estimated human motions from images to predict future frames, and the results are utilized in a real-world robotic scenario for the robot follow-ahead task. STPOTR, similar to DMMGAN, simultaneously predicts human pose and trajectory and achieves acceptable accuracy for both predictions. An ablation study also shows the benefit of the shared attention module in improving the model's performance. As STPOTR can run four times faster than DMMGAN, it is an excellent alternative for robotic tasks that require speed and accuracy. We explore the method and its results for human motion prediction in Section 2.3, while a separate discussion of the human-following use case of STPOTR can be found in Section 3.3.

Both works are published at the IEEE International Conference on Robotics and Automation (ICRA) 2023 [56, 69]. This chapter is based on the research presented in these papers.

## 2.1 Related Work

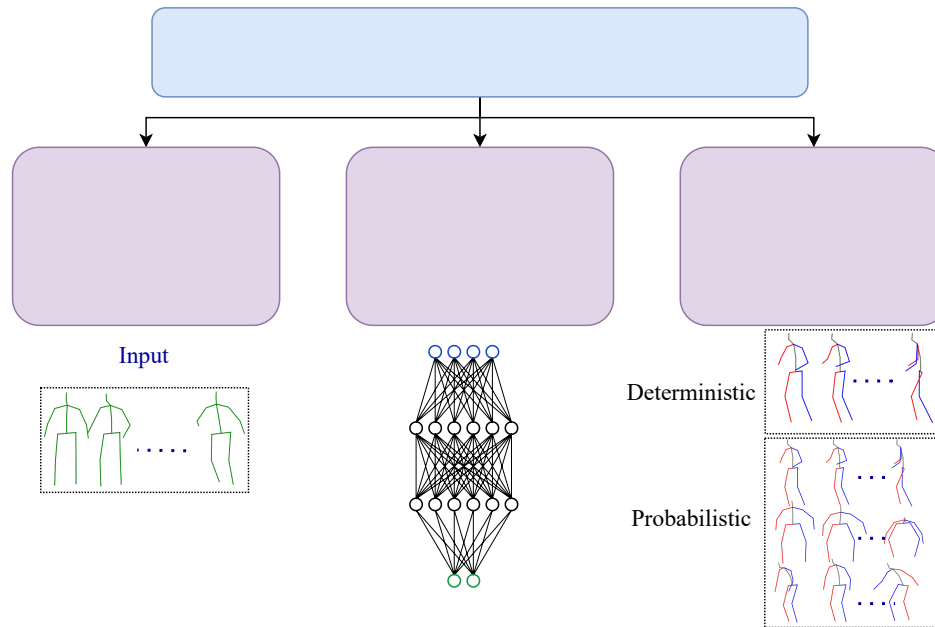


Figure 2.1: 3D human motion prediction task.

The illustration of the human motion prediction task can be found in Fig. 2.1. According to Lyu et al., it can be broken down into three key components [55]:

- Observed sequence: Human motion representation is an important part of the human motion prediction pipeline. The understanding of human behavior can be enhanced by using different representations.
- Prediction model: Network structure design is the second part. In the human motion prediction task, the prediction model is the main part of most methods.
- Prediction results: Finally, the third part is the type of predictions. Human motion prediction methods can be categorized into probabilistic or deterministic approaches. Probabilistic methods try to model human motion similar to the real world with multiple possibilities, while deterministic ones only output one possible outcome.

### 2.1.1 Observed sequence

In human motion prediction, we usually represent human motion by a skeleton kinematic tree. This skeleton is composed of the 3D position of human joints. In this thesis, we focused only on joints that can be detected using a motion capture system.

We must first encode human motion into an observed sequence to predict it. Here we go over the different lines of research on the human motion concerning their input and



how they encode human motion. In the following, we go over different types of observed sequences based on the literature.

#### 2.1.1.1 Human body structures

In this category, the researcher tries to learn a structure for the human body or optimize the learning based on the human body structure. Guo et al. improve human motion prediction by learning the local structure of the human skeleton. They divide the human skeleton into five non-overlapping sections. Then predict the next 3D pose by learning one representation for each section [27]. As another example, Li et al. divide human motion using 1D convolution layers. They design hierarchical convolutional layers. In each level, two neurons are linked together if and only if they are from adjacent joints [49]. Their convolutional hierarchical autoencoder can encode the human 3D pose based on human bone structure constraints. In this line of work, the authors always enforce some prior knowledge related to human body structure with a simple network structure which results in limited performance improvements.

#### 2.1.2 Human motion laws

In the second strategy, human motion is regarded as a movement relative to another joint. This can include both acceleration and velocity. In [90], Wang et al. represent human motion by velocity and acceleration instead of typical 3D position. They prove that this representation can simplify the learning of human motion prediction. Their approach is limited to the short-term prediction of fewer than 400 milliseconds.

##### 2.1.2.1 Mathematical representation

Researchers use the mathematical encoding of the 3D motion to encode the human motion representation better. They leverage their prior knowledge of human motion into the training by using existing mathematical representations. Here we review the Graph, Motion trajectory, and joint angles methods.

**1. Graph:** Our bone structure can easily be described as a graph with joints as vertices and bones as the edges. This resemblance leads the researcher to use Graph representation for human motion [57, 58]. As one illustration, Jain et al. represent the human body using a spatial-temporal graph [38]. One of the main limitations of their work is designing the graph manually, which limits the flexibility and advantages of their method. As an improvement, Cui et al. designed a trainable adjacency matrix for the graph [16]. Specifically, they designed two parameterized graphs for learning the relationship between joints in the human skeleton and during human motion. One of the graphs learns the kinematic links between the human skeleton, and the other graph learns the global relation. For example, when a person runs, there is a strong correlation between the right and left hand. Their adjacency matrix is

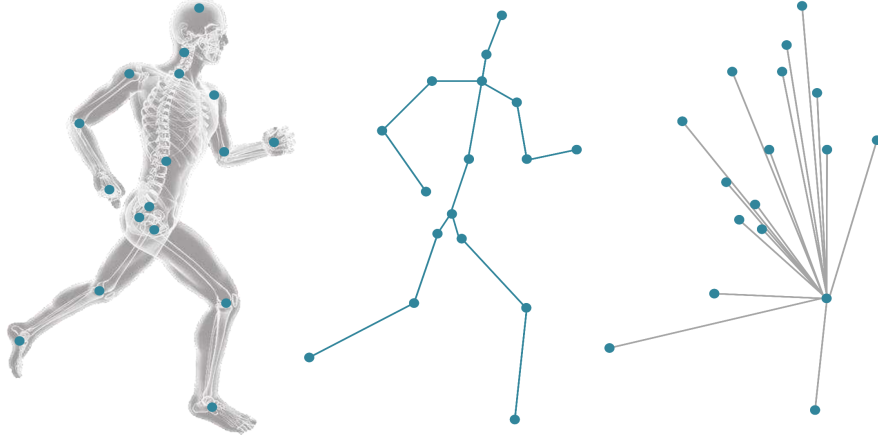


Figure 2.2: **Left**: original human skeleton, **Middle**: local graph, a graph with learnable connections initialized from the original human skeleton [16], **right**: global graph with learnable connections for right knee joint [16].

part of the network parameters. Fig. 2.2 illustrates the global graph on the right and the kinematic graph in the middle. Similarly, in [15, 83], they learn a graph without explicitly encoding it. According to the results of these studies, graph networks can improve the training of human motion prediction.

**2. Motion trajectory:** In this category, human motion trajectory is the path of human joints. Here, researchers try to model human motion in trajectory space instead of motion space. Many studies used trajectory space to encode human motion[57, 58]. In these works trajectory of each joint is defined as:

$$T_j = (t_{j,1}, t_{j,2}, t_{j,3}, \dots, t_{j,N}) \quad (2.1)$$

Here  $j$  denotes the  $j^{th}$  joint, and  $N$  denotes the frame number.

Parallel to this representation, Liu et al. use the displacement of frames as its trajectory representation [52]. The authors propose to encode joints' trajectories using a combination of frame-wise velocity and final state information. For this task, they initialize a Graph Convolutional Network (GCN) with connections for adjacent joints. Later using this GCN, they learn some new implicit connections. Using this representation, their method learns some useful cues from the velocity of joints instead of only training using the connection between adjacent joints. The result of their work shows improvement in training with better performance.

**3. Joint Angles:** Joint angle is a common representation of human body motion in predicting human movements. A joint angle describes the angle between adjacent body segments, while a joint 3D pose represents the 3D position of each endpoint of a joint in the human body. Martinez et al. introduce a human motion prediction model that utilizes joint angle representations and recurrent neural networks. The authors evaluate the effectiveness

of their approach on the Human 3.6M dataset [59]. Li et al. propose modifying the loss function by directly calculating the Euclidean difference of the Euler angle [49]. This is an improvement over a loss based on an exponential map representation of each joint as demonstrated in previous works [25, 59]. The authors illustrate the improved performance of their proposed approach.

### 2.1.3 Model and Prediction Results

In this section, we go over different models and evaluate them based on their prediction models and the type of prediction results.

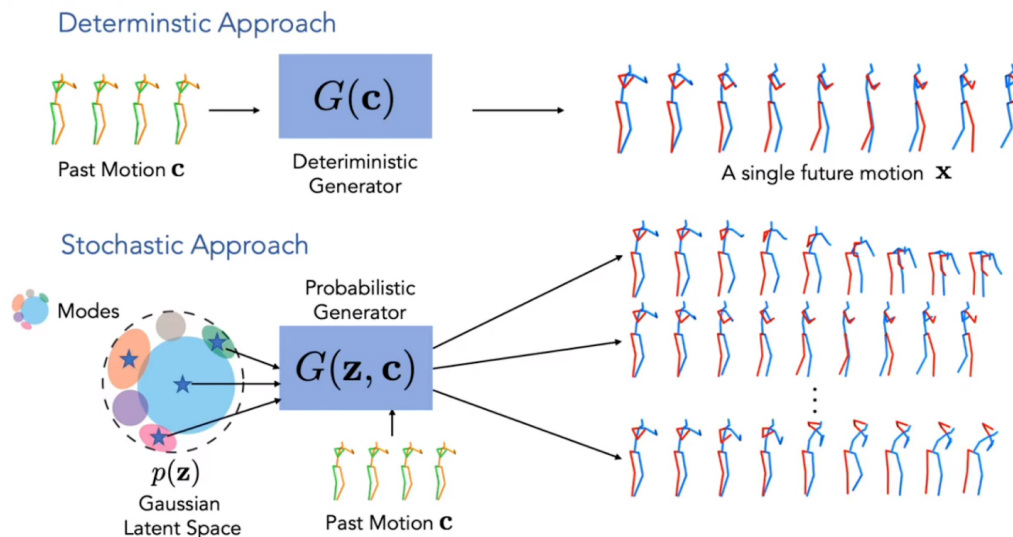


Figure 2.3: According to the type of prediction result, human motion prediction can be categorized as probabilistic or deterministic types.

Based on the prediction type, human pose prediction methods can be categorized into probabilistic or deterministic [55] methods. In probabilistic methods, similar to how our brain works, they predict multiple future motion sequences for an observed motion sequence. While deterministic methods aim to predict a single sequence more accurately, their practicality in some robotic applications is reduced because they do not consider the diversity and multimodality of human behavior. Arguably, the probabilistic approach is preferred in robotic applications as it provides more assurance by considering a set of possible scenarios. However, probabilistic methods may reduce the accuracy of each individual predicted sequence.

Alongside the type of prediction, we can also divide the human motion prediction task into the human trajectory and human 3D pose. In this context, human trajectory refers to the human’s path in 3D space, such as the path of the hip joint, and 3D pose refers to the location of the human body’s joints relative to a fixed joint. For solving both problems, seq2seq models have been utilized successfully with room for improvement.

Many human trajectory prediction methods have been developed for autonomous driving systems [1, 23, 74], in which the primary goal is to predict the future trajectory of pedestrians to avoid colliding with them. Recently, some works attempted to use transformers [86] to predict multiple possible human trajectories. A limitation of current methods is that very few of them attempt to predict both human poses and trajectories simultaneously. Furthermore, works on human trajectory predictions are sometimes limited due to only considering the hip movements and ignoring other joints. However, joints can provide valuable information about how the hip may move in the 3D space.

In what follows, we go over different network architectures to solve the human motion prediction task for the probabilistic and deterministic prediction types.

### 2.1.3.1 Deterministic Prediction

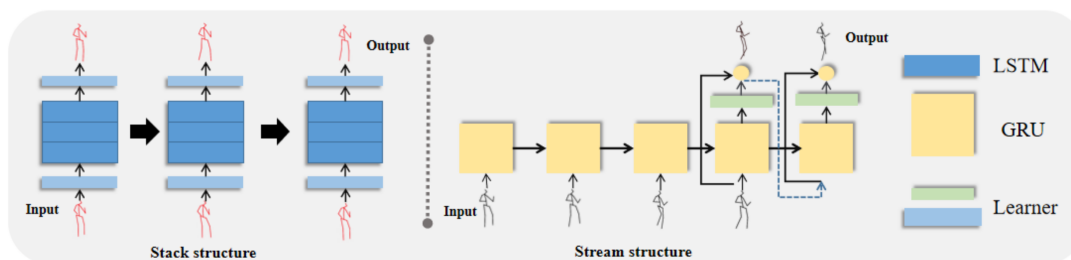


Figure 2.4: Two basic types of RNN prediction methods: stack [20] and stream [59] structures.

In the past decade, with the popularity of deep learning, sequence-to-sequence (seq2seq) prediction methods such as those involving Recurrent Neural Networks (RNN) [20, 59] have shown promising results and have become a viable alternative to conventional human motion prediction methods [46, 79]. Many deep learning approaches use RNN-based models to predict human motion. The preliminary use cases of RNNs in human motion predictions can be seen in LSTM-3LR [59] and ERD [20] architectures. LSTM-3LR and ERD both have similar network architecture. These methods both suffer from accumulated errors and discontinuities in the initial frames. To improve these shortcomings, Martinez et al. proposed res-GRU [59]. In res-GRU, they define the decoder based on the combination of the velocity of the joints and the previous predicted frame. They also add noise to the input during the training to improve the network accuracy and to reduce drifting. Their result shows an improvement over the previous methods for both accumulated errors and discontinuities in the initial frames. In a more recent study, Jain et al. proposed structural RNNs (SRNNs). In SRNNs, they hand-designed a spatial-temporal graph and combined it with RNNs. Their structure helped to improve the prediction accuracy. Then in [22], they show how combining a dropout auto-encoder with LSTM-3LR can mitigate the drifting problem. By adding a dropout auto-encoder to the LSTM output, their model can reduce drift error caused by

noisy predictions. Another interesting work in this area is Video Inference for Human Body Pose and Shape Estimation (VIBE)[42]. Using a video sequence of someone moving, VIBE estimates their 3D pose and shape. To exploit the nature of human motions, Vibe Frist extracts image features using CNN and processes them using recurrent neural networks. They train their network using a discriminator trained on distinguishing between fake and real sequences using AMASS dataset.

All of these approaches are classified as autoregressive models, which are a type of neural network that generates a sequence of outputs based on previous predictions. Nevertheless, autoregressive models have two significant drawbacks. Firstly, they are prone to accumulating prediction errors, meaning that any errors in prior predictions are carried over to subsequent predictions, leading to substantial errors over time. Secondly, they are not parallelizable, resulting in high computational requirements during testing[60]. Recently, several methods have tried to prevent the drift issue by including adversarial losses and enhancing prediction quality. Among these methods, [25] proposes adversarial geometry-aware encoder-decoder (AGED) by using geodesic body measurements as an adversarial loss. In AGED, they use two discriminators to learn both the continuity and fidelity of the predictions. Their result improves both reducing the discontinuities and more accurate predictions. As a result of using adversarial training, their method is difficult to train and stabilize. Moreover, to better embed the joint dependencies, some methods combine their algorithm with spatio-temporal modeling to better learn the relation between all the joints in a single frame or a sequence of frames [21, 48].

With the improvement of transformer models [86], in a few studies, they have been employed to solve the human pose prediction problem. Aksan et al. proposed an autoregressive transformer to learn to decouple spatio-temporal representations [3]. They achieved acceptable results in terms of accuracy; however, the autoregressive nature caused the algorithm to be slow. Conversely, González et al. proposed a non-autoregressive version called Pose Transformer (POTR), which performed faster with lower accuracy [60]. To learn the temporal dependencies, they use the main encoder-decoder structure of transformers [86]. Their encoder and decoder networks use GCNs and Multi-Layer Perceptron (MLP) layers to determine spatial dependencies between joints in one frame. During training and testing, the last observed frame is copied and used as their decoder input with a residual connection to the decoder output. Therefore the decoder would learn the sequence offset with respect to the last seen frame. In all methods described above, the hip joint and, sometimes, the heading are fixed, which makes them impractical for robotic applications. The design of our deterministic motion model is partially inspired by Pose Transformer (POTR) [60]. However, all mentioned methods consider a fixed hip joint and even, in some cases, fixed heading, which makes them impractical for many robotic applications. Therefore, we have made multiple improvements to the model and data structure to make it suitable for robotic tasks.

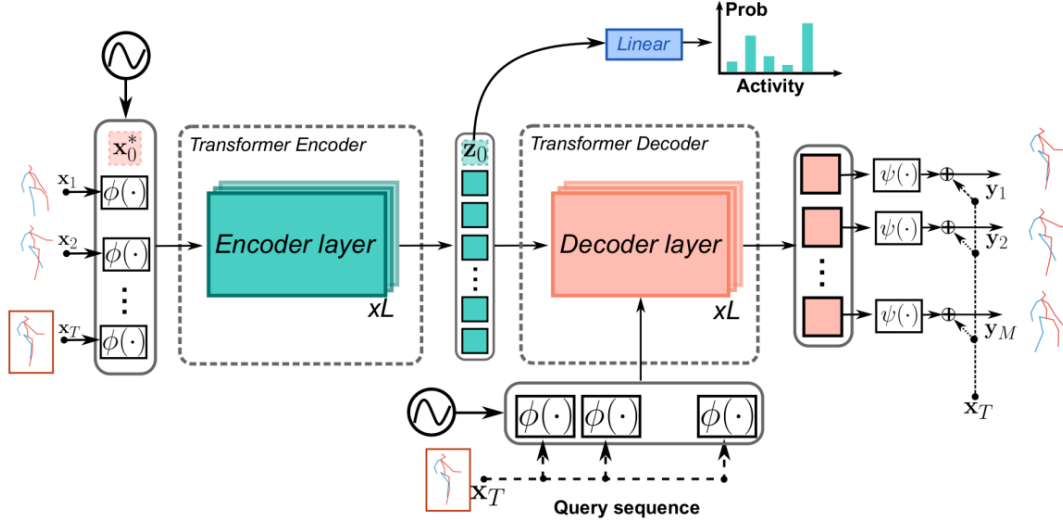


Figure 2.5: Overview of POTR approach. POTR is a non-autoregressive human motion prediction method. For each pose in the input sequence, a network computes embeddings. Then, in parallel, the Transformer processes the sequence and decodes the attention embeddings. Lastly, A residual network predicts the sequence [60].

### 2.1.3.2 Probabilistic Prediction

In this category, the future is probabilistic similar to how humans behave. Therefore, there may be multiple predictions based on the same prior poses. Probabilistic approaches gained popularity with the development of GANs. These methods [4, 7, 8, 50, 87] usually use CGANs, Conditional Variational Autoencoders (CVAEs), or diffusion models. As examples of these probabilistic methods, Yan et al. [95] developed a Motion Transformation Variational Autoencoder (MT-VAE) to generate multiple diverse and plausible motion sequences for facial and full-body motions from an observed sequence. More recently, Agand et al. [2] developed a probabilistic and optimal approach for human navigational intent inference. They predict the probability distribution over human kinematic states using only trajectory and not the entire 3D pose.

As one of the first applications of GANs in human pose prediction, HPGAN [8] proposes a sequence-to-sequence model for predicting multiple future human poses. HPGAN uses an implementation of Wasserstein GAN with the gradient penalty. Their network train to learn the probability distribution of its data. As a result, they can predict different future poses using the same input sequence by changing a vector  $z$ . However, they still have the problem of discontinuity of the motion, and they haven't compared their result with deterministic approaches. Similarly, BiHMP-GAN [45] tries to fix the mode collapse problem of the GAN framework by using bidirectional GAN. BiHMP-GAN provides comparisons with deterministic approaches, and their result shows an improvement over determinist approaches.

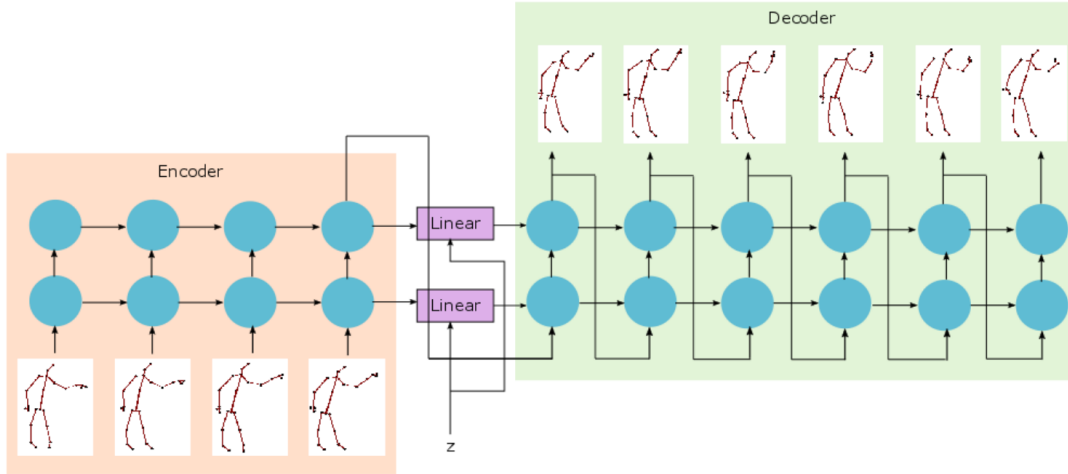


Figure 2.6: HP-GAN, predicting multiple future poses from a single input sequence by feeding vector  $z$  [8].

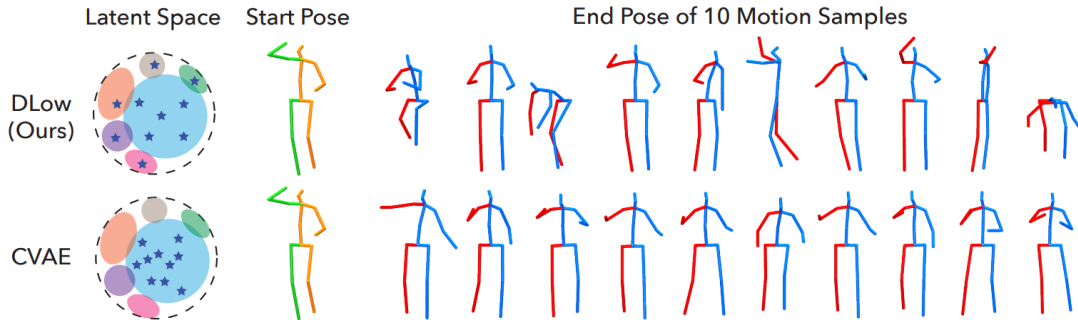


Figure 2.7: DLow samples (stars) cover more modes (colored ellipses) in the latent space of a conditional variational autoencoder (CVAE) than CVAE samples. Because of this coverage, DLow can output a diverse future prediction in comparison to normal CVAE [97].

Most generative methods focus on learning the distribution of the dataset and not on the diversity of the prediction. These generative models randomly sample from a Gaussian latent space and extract motion samples from them, which may produce similar samples. As one of the state-of-the-art methods in diverse human motion prediction, DLow [97] focuses on improving the diversity of the samples. The authors train a mapping function that samples diversely using a pretrained CVAE. They train a set of learnable mapping functions with correlated latent space that uses an energy-based formulation based on pairwise sample distance to diversify the samples. Their result shows a significant improvement over the previous methods for both the accuracy and diversity metrics.

In a recent study, Barquero et al. suggest the use of diffusion models for human motion prediction [7]. Previous works on human motion prediction have shifted from a deterministic approach to a stochastic one, recognizing the unpredictable nature of human behavior. However, these works have mainly focused on predicting highly diverse motion distributions,

similar to DLow [97], which may generate fast and divergent motions while overlooking the importance of predicting low-speed diverse motions in certain scenarios like assistive robotics or surveillance. This bias towards fast and divergent motion can produce unnatural or inconsistent motions. To overcome these limitations, [7] introduced the BeLFusion method, which builds a latent space to disentangle behavior from poses and motion, thereby promoting diversity from a behavioral perspective. The method uses conditional latent diffusion models, resulting in improved performance in human motion prediction while ensuring that the predicted motions are coherent with the immediate past and more realistic. Additionally, the method provides complementary metrics for assessing the statistical similarities between predicted and intrinsic dataset diversity, contributing to a more comprehensive evaluation pipeline for stochastic human motion prediction. Overall, the BeLFusion method enhances previous approaches to human motion prediction by promoting diversity from a behavioral perspective and ensuring that the predicted motions are both diverse and realistic.

#### 2.1.4 Human Motion in Animation

Human motion prediction in animation has been a fundamental problem in computer animation for decades. Various approaches have been proposed to generate realistic and purposeful human movement. One such approach is the use of Motion VAEs, which learn data-driven generative models of human movement using autoregressive conditional variational autoencoders [51]. The latent variables of the learned autoencoder define the action space for the movement and govern its evolution over time. Planning or control algorithms can then use this action space to generate desired motions. Another approach is the use of a neural state machine, which is a data-driven framework to guide characters to achieve goal-driven actions with precise scene interactions [82]. This framework enables the modeling of multi-modal scene interaction behaviors purely from data, making it versatile for various scene interaction tasks such as sitting on a chair, avoiding obstacles, opening and entering through a door, and picking. A third approach is the use of Phase-Functioned Neural Networks, which can produce higher quality results than time-series autoregressive models such as LSTMs [31]. This network architecture deals explicitly with the latent variable of motion relating to the phase, making it appropriate for controlling characters in interactive scenes such as computer games and virtual reality systems.

In a recent study, Peng et al. generated physically simulated character animations using adversarial imitation learning on unlabeled motion clips [72]. The resulting embeddings can be used to learn a hierarchical skill-conditioned policy that produces versatile animations. This framework enables characters to learn reusable skill embeddings from unstructured datasets and apply them to new tasks. A character can use these learned skills to run to a target and knock it over.

To the best of our knowledge, although most techniques for human motion prediction in animation forecast both trajectory and human pose, they have not yet been applied to any



robotics task. Certain approaches can be computationally intensive, as they do not have any restrictions on computation, and animation generation can be time-consuming, focusing instead on utilizing reinforcement learning or other methods to control motion prediction. This thesis will not delve into such techniques and instead concentrate on human motion prediction employing datasets such as Human 3.6M [36].

### 2.1.5 Datasets

A dataset is a crucial component of any machine learning method, particularly for deep learning models where data plays a more significant role compared to classical approaches. Over the past few decades, researchers have developed and released numerous datasets to enhance our understanding of human motion. In the following, we will provide a brief overview of four of these datasets.

#### 2.1.5.1 Human 3.6M:

Human 3.6M is a large dataset with seven actors (four other actors without ground truth data). It includes 3.6 Million 3D Human poses captured using four different viewpoints. For each actor, 15 actions are recorded using a high-speed motion capture system at 50 Hz. They also include the 3D scan of each actor and time of flight (depth) data<sup>1</sup>.

#### 2.1.5.2 CMU motion capture:

In this dataset, they used 41 markers on their human subject and 12 infrared cameras to record them. They recorded 144 subjects, including running, walking, basketball, etc. This dataset has been recorded by Carnegie Mellon University <sup>2</sup>.

#### 2.1.5.3 The Archive of Motion Capture as Surface Shapes (AMASS):

To generate AMASS, they used 15 optical markers on the human body. It is a large dataset containing 11265 motions and 344 subjects. It includes the parameterized human motion model <sup>3</sup>.

#### 2.1.5.4 NTU RGB-D:

NTU RGB-D combines RGB videos, depth maps, 3D motion data, and depth videos using three Kinect cameras. It has a total of 4 Million frames divided into 60 action classes <sup>4</sup>.

<sup>1</sup><http://vision.imar.ro/human3.6m/>

<sup>2</sup><http://mocap.cs.cmu.edu/>

<sup>3</sup><https://amass.is.tue.mpg.de/>

<sup>4</sup><https://rose1.ntu.edu.sg/dataset/actionRecognition/>

## 2.2 DMMGAN: Diverse Multi Motion prediction of 3D Human Joints using Attention-Based Generative Adversarial Network

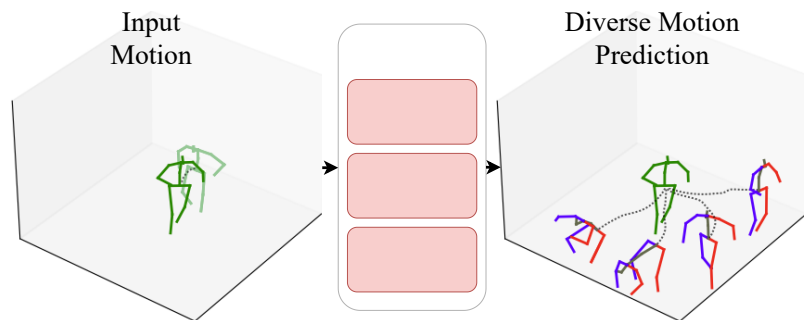


Figure 2.8: Given a sequence of 3D human motions, our system generates a diverse set of future motions. The *3D pose prediction* module generates diverse 3D poses while *hip prediction* module estimates the human trajectory together forming a 3D human motion. The *discriminator* module distinguishes a real 3D human motion from a generated one.

The prediction of 3D human motion can be separated into Human Pose Prediction and Human Trajectory Prediction. Pose refers to the position of body joints relative to the hip joint, while Trajectory refers to the path of the hip joint as the body moves in 3D space. Seq2seq models have been effectively used to solve these problems, with room for improvement. As we discussed earlier, the prediction of human motion can be approached either probabilistically or deterministically [55]. Probabilistic methods predict multiple possible future motion sequences, providing more assurance and better capturing the diverse nature of human behavior, but potentially sacrificing accuracy; in addition, they are harder optimization problems. Deep generative models, such as VAEs and GANs, have demonstrated considerable accuracy in such methods. One notable state-of-the-art method, DLow [97], employs deep generative models and a unique sampling technique for multi-future pose predictions. Deterministic methods, on the other hand, aim for a more accurate single prediction, but ignore the diverse and multi-modal nature of human behavior, which limits their usefulness in some robotics applications. Additionally, human trajectory predictions are sometimes limited in their consideration, focusing only on hip movements and disregarding other joints, despite their potential to provide valuable information about hip movement in space.

In DMMGAN, we combine the benefit of both probabilistic and deterministic methods to provide multiple accurate predictions for both 3D human trajectory and pose. We hope this opens doors to practical use in real robotic applications. To generate multiple future human motions, we use a conditional generative adversarial network (CGAN) with a transformer-

based encoder for better encoding of the observed sequence. At the end, a GRU combined with a GAN provides multiple future predictions autoregressively.

Our contributions in DMMGAN are:

- We propose a novel deep generative architecture involving transformer-based encoders to predict a diverse set of possible human body motions.
- We provide a real-time solution for diverse 3D human motion prediction, including both pose and trajectory prediction, which can potentially be more suitably used for robotics and autonomous car applications.
- In addition to providing both pose and trajectory predictions, our work achieves better accuracy compared to the state-of-the-art models in standard evaluation metrics.

### 2.2.1 Problem Setup

Our framework predicts a diverse set of human motions. The input is a sequence of 3D body motion  $S = \{S_{t-\alpha}, S_{t-\alpha+1}, \dots, S_t\}$  of the past human’s skeleton movements captured up to the current time  $t$  where  $S_i \in \mathbb{R}^{51}$  represents the 3D positions of 17 human joints at time  $i$ . The outputs of our system are  $N$  possible sequences of future 3D human motion  $O_i^\gamma = \{O_{t+1}^\gamma, \dots, O_{t+\zeta}^\gamma\}$  where  $\gamma \in 1, \dots, N$  is the sequence number and  $\zeta$  is the forecast duration. We divide the human 3D motion into two parts so that  $S_i = (S_i^H, S_i^P)$  and  $O_i = (O_i^H, O_i^P)$ . The position of the hip joint is denoted by  $S^H$  and  $O^H$  for input and output hip trajectories. The relative positions of all joints with respect to the hip joint (called 3D pose, or just *pose*), denoted by  $S^P$  and  $O^P$  for input and output 3D pose sequences.

### 2.2.2 Method

The overall framework of our system is summarized in Fig. 2.9. Our method learns to generate valid and rich human motions by leveraging the Human 3.6M dataset [36]. It divides the prediction of human 3D motion into predicting the joints motion relative to the hip joint (3D pose) and predicting the 3D position of the hip joint in the global frame for each predicted 3D pose (human trajectory). We estimate the human trajectory by considering both the predicted 3D pose and the trajectory history.

Specifically, we design our model to benefit from both paired and unpaired data by introducing four supervised losses and a discriminator loss respectively. Here, given a sequence of 3D motion  $\{S_{t-\alpha}, \dots, S_t\}$ , a transformer encoder learns representation of the input in a latent space. Then, a generator uses this latent representation to output  $N$  future motions. To train our system, we use 5 losses. The *Best Loss* finds the best match with shortest distance to the ground truth data. The *Teacher Forcing Loss* improves the final prediction by randomly feeding ground truth instead of the model prediction in the decoding phase. Similar to the Best Loss, the Teacher Forcing Loss only applies for the output that matches

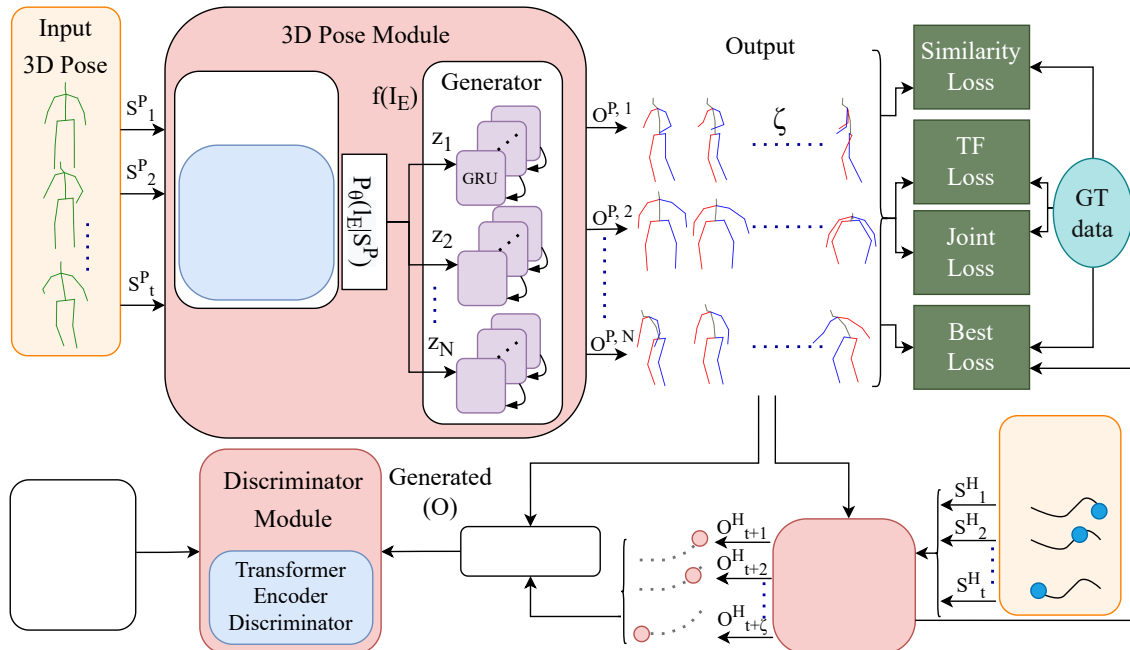


Figure 2.9: System overview: Given a sequence of 3D human motion, our method generates  $N$  future sequences of human 3D motion using a discriminator and four loss functions. Our system consists of three main parts. The first part is predicting the human 3D pose (*3D Pose module*) by receiving a history of the human 3D pose. The second part is the *Hip Prediction* module (more details in Fig. 2.10) which predicts the future position of the hip joint for each of the predicted human 3D poses. Finally, the discriminator module learns the distribution of the Human 3.6M dataset by distinguishing between generated and real data. The system uses the discriminator loss to generate sequences similar to the dataset distribution while using four supervised loss functions to promote accuracy and diversity. See Fig. 2.10 for Transformer Encoder architecture.

the most closely with the ground truth. The *Similarity Loss* promotes diversity by penalizing the pairwise distance between the  $N$  generated sequences, and lastly, we use the *Joint Loss* to encourage joint length constraints. We combine these losses with the *Discriminator Loss* to generate plausible sequences matching the Human 3.6M dataset [36].

### 2.2.2.1 Model Architecture

Our model consists of three main modules, the first module is the *3D pose module*, which generates  $N$  sequences of human 3D pose (relative to the hip joint). The second module is the *Hip Prediction module*, which predicts the trajectory of the hip joint in the global frame for each predicted human 3D pose. Finally, the last module is the *Discriminator module*, which learns the distribution of the dataset by distinguishing between the real and generated 3D sequences of human’s motion.

**1. 3D Pose Module:** The 3D Pose module consists of two parts, as shown in Fig. 2.9. The first part is the encoder. Given a sequence of human 3D pose  $S^P$ , it outputs a latent representation  $l$  that encodes the past motion  $P_\theta(l_E|S^P)$ . Our encoder network uses a Transformer architecture, as shown in Fig. 2.10, to learn meaningful information over a sequence of 3D poses, similar to the model introduced by Vaswani et al. [86].

The second part is the generator. It forecasts  $N$  sequences of human 3D pose  $O^{P,1}, \dots, O^{P,N}$  given the past latent representation  $l_E$ . Instead of using random variables as the input of the generator to forecast the future, we design our network to learn a mapping from the latent representation to  $N$  priors  $z = f(l_E)$ . Then it initializes  $N$  generator networks with Gated Recurrent Units (GRU) [13], each of which forecasts a sequence of future 3D pose based on their prior  $P_{\phi_n}(O_n^P|z_n), n \in \{1, \dots, N\}$ .

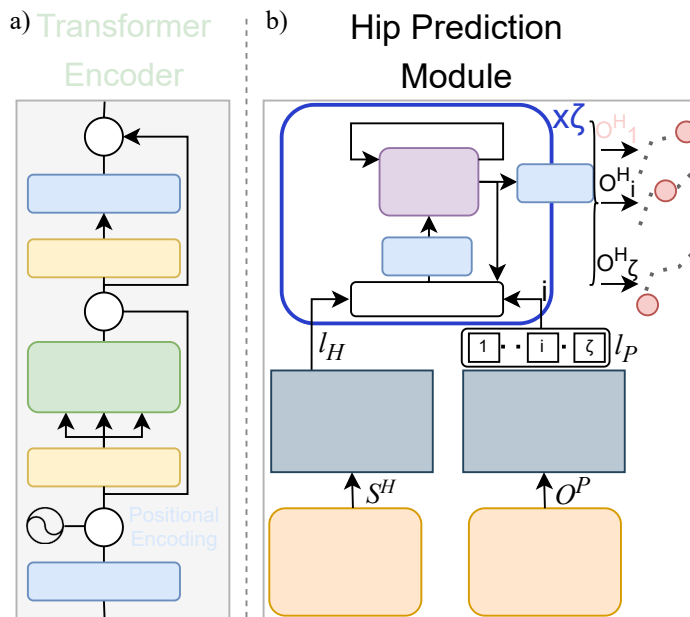


Figure 2.10: a) The Transformer Encoder [86] and b) the *Hip Prediction* module architectures. The *Hip Prediction* module, estimates the hip joint positions of each predicted 3D pose by receiving the history of the hip movements and the motion predicted by the *3D Pose* module.

**2. Hip Prediction Module:** The second module is the *hip prediction* module. Given the 3D pose predictions  $O^P$  and the trajectory history  $S^H$ , it estimates the position of the hip for each predicted 3D pose.

Fig. 2.10 shows the architecture of the *Hip Prediction* module. It uses two Transformer encoders. The first one learns a representation  $l_H$  from the observed hip movements  $S^H$  and the second one learns a representation  $l_P$  from a predicted 3D pose sequence. If the transformer embedding has  $\sigma$  dimensions and the input has a length of  $\alpha$  frames and we

predicted a 3D pose sequence with  $\zeta$  frames, the output of  $S^H$  has  $\alpha\sigma$  and  $S^P$  has  $\zeta\sigma$  dimensions. The GRU gets the concatenation of  $l_{P,i}$ ,  $l_H$  and the previous output of the GRU as its input, to predict the position of the hip at time  $i$  for  $i = 1, \dots, \zeta$ .

**3. Discriminator Module:** The last module is the discriminator. Here we use a Transformer-based Encoder architecture shown in Fig. 2.10a. The input of the discriminator is the full human 3D motion, consisting of the hip trajectory and the 3D pose trajectory. The discriminator needs to distinguish between the real and the generated data (Fig. 2.9).

### 2.2.2.2 Model Training

During training, we exploit paired data by introducing four supervised losses to promote the diversity and accuracy of the predictions. We also benefit from unpaired data by using a discriminator that learns to distinguish between the real and generated data. In the following we use the ground truth,  $GT$ , term to refer to the paired data,  $GT^P$  and  $GT^H$  to refer to the ground truth paired 3D pose and hip trajectory respectively.

**1. Discriminator Loss:** We implement the discriminator loss based on the Wasserstein Generative Adversarial Network (WGAN) [5]. To make the training more stable we used the Gradient Penalty (GP) version of the WGAN. If  $f$  is the discriminator network, the GP WGAN critic’s loss function is defined as follows:

$$\mathcal{L}_{cWGAN} = \mathbb{E}_{O \sim P_g} [f(O)] - \mathbb{E}_{GT \sim P_r} [f(GT)] \quad (2.2)$$

$$\mathcal{L}_{cGP} = \mathcal{L}_{WGAN} + \lambda \mathbb{E}_{\bar{x} \sim P_{\bar{x}}} [(\|\nabla_{\bar{x}} f(\bar{x})\|_2 - 1)^2] \quad (2.3)$$

where (2.2) is the original critic loss function of WGAN method and the last term of (2.3) is the gradient penalty term. Consider a line connecting real ( $P_r$ ) to generated ( $P_g$ ) distributions.  $P_{\bar{x}}$  is the distribution of these samples and  $\lambda$  is the weight of the gradient penalty.

The second part of the discriminator loss function is the generator objective. The objective of the generator is to minimize the distance between  $P_g$  and  $P_r$  by maximizing the expectation of the generated samples:

$$\mathcal{L}_g = - \mathbb{E}_{O \sim P_g} [f(O)] \quad (2.4)$$

**2. Best Loss:** Given a sequence of human’s 3D motion, our model predicts multiple forecasts of future motions. Using the discriminator loss, these forecasts would be similar to the distribution of the dataset. The Best Loss minimizes the distance between the closest prediction and the  $GT$  data using mean squared error (MSE). The Best Loss is defined as

follows:

$$\mathcal{L}_{best} = \sum_{T=t+1}^{t+\zeta} MSE(O_T^\Gamma, GT_T) \quad (2.5)$$

$$\text{where } \Gamma = \arg \min_{\gamma=1, \dots, N} \sum_{T=t+1}^{t+\zeta} D(O_T^{P,\gamma}, GT_T^P) \quad (2.6)$$

$$\text{and } D(O_t^\Gamma, GT_t) = \sum_{T=t+1}^{t+\zeta} \sum_{j=1}^{17} d(O_{t,j}^\Gamma, GT_{t,j}) \quad (2.7)$$

Here,  $D$  is the distance between two 3D motion predictions and  $d$  is the Euclidean distance between two joints.

**3. Teacher Forcing Loss:** After calculating the predicted sequence that matches with the  $GT$ , the *Teacher Forcing* (TF) loss is calculated by randomly using the next frame from the  $GT$  instead of the last prediction in the GRU (Fig. 2.9 Generator). The TF loss can be especially useful in reducing the final displacement error as the model can learn to predict the next frames by using a combination of the  $GT$  and its own predictions [91].

**4. Similarity Loss:** We define the *Similarity* loss to increase the variety of the model predictions. We first find the distance between each pair of the predicted human 3D pose. Then select the two predictions,  $\Gamma_1$  and  $\Gamma_2$ , with the shortest distance.

$$\Gamma_1, \Gamma_2 = \arg \min_{\substack{\gamma_1 \in \{1, \dots, N\}, \\ \gamma_2 \in \{1, \dots, N\} \setminus \gamma_1}} \sum_{T=t+1}^{t+\zeta} D(O_T^{P,\gamma_1}, O_T^{P,\gamma_2}) \quad (2.8)$$

We can define the distance of each two joints of  $\Gamma_1$  and  $\Gamma_2$  by:

$$distJoints_j = \sum_{T=t+1}^{t+\zeta} d(O_{T,j}^{P,\Gamma_1}, O_{T,j}^{P,\Gamma_2}) \quad (2.9)$$

Then we apply the negative of MSE to the joints that exceed the average *Similarity loss* threshold of  $\epsilon$ . We can define the *Similarity loss* as follows:

$$\mathcal{L}_{similarity} = -\frac{1}{16} \sum_{j=0}^{16} distPenalize_j^2, \text{ where} \quad (2.10)$$

$$distPenalize_j = \begin{cases} 0 & \text{if } distJoints_j < \epsilon \\ distJoints_j & \text{otherwise} \end{cases} \quad (2.11)$$

To make the training more stable we use the *Similarity loss* only during the first  $M$  steps of the training.

**5. Joint Loss:** As human’s bone length stay the same, joint Loss works as a regularizer that helps the model by forcing it to keep the bone length similar over time. If  $V$  is the set of vertices of a graph representing all human joints and  $E$  is the edges of this graph representing all human bones, then the joint loss is defined as follows:

$$\mathcal{L}_{joint} = \sum_{(i,j) \in E} \sum_{\gamma=1}^N MSE(J_{i,j}^{P,\gamma}, J_{i,j}^{P,GT}) \quad (2.12)$$

$$\text{where } J_{i,j}^{P,\gamma} = \frac{1}{\zeta} \sum_{T=t+1}^{t+\zeta} (d(O_{T,i}^{P,\gamma}, O_{T,j}^{P,\gamma})), \quad (2.13)$$

$$J_{i,j}^{P,GT} = \frac{1}{\zeta} \sum_{T=t+1}^{t+\zeta} (d(GT_{T,i}^P, GT_{T,j}^P)) \quad (2.14)$$

### 2.2.2.3 Data Preprocessing

To improve the model prediction and avoid over-fitting, we convert each 3D position in a sequence of human motion to a relative coordinate system based on the position of the hip joint at the time  $t$ . We also normalize each skeleton 3D pose ( $\mu = 0, \sigma = 1$ ).

### 2.2.2.4 Dataset

For our experiments and training, we use the Human 3.6M dataset [36]. Human 3.6M is a large dataset with 7 actors<sup>5</sup>. For each actor, there are 15 actions that are recorded using a high-speed motion capture system at 50 Hz. Similar to DLow [97], we use 17 joints skeleton and train on actors S1, S5, S6, S7 and S8 while testing on S9 and S11. For future prediction, our model observes 0.5 seconds sequence of human’s body motion to forecast the next 2 seconds.

## 2.2.3 Experiments and Results

Our method is specifically designed to forecast 3D motions that are suitable for autonomous car or robotics applications. It can predict the human 3D pose (position of joints relative to the hip joint) while predicting their trajectory (hip joint) separately. Most of the previous works only predict the human 3D pose without the human’s hip trajectory.

Here we designed two experiments. The first one evaluates our 3D pose prediction without the trajectory prediction module. Then in the second experiment, we evaluate our full system. For both experiments, we used the same model (DMMGAN). Our model can run at 10 frames per second (FPS) on a GeForce 1080 GPU. Since most robotics applications require the observation to come with a frequency of fewer than 10 FPS, we train our model

<sup>5</sup>There are 4 other actors without ground truth data



and the baselines using the Human3.6M [36] at 10 FPS. For DLow and our methods, we predict 10 sequences per observation ( $N = 10$ ).

To evaluate our model versus the baselines we measure the accuracy and diversity using the following metrics (we are using the evaluation metrics similar to [97, 98]):

**1. Average Pairwise Distance (APD):** Evaluates diversity among the predictions. We calculate the APD by averaging the pairwise distance between all pairs of 3D pose samples between the predictions. The APD is calculated as  $\frac{1}{N \times (N-1)} \sum_{i=1}^N \sum_{j \neq i}^N \|O_i^P - O_j^P\|$ .

**2. Average Displacement Error (ADE):** Mean squared distance between the ground-truth and the closest prediction. We define the ADE for both the 3D pose and the hip trajectory movements. We first calculate the closest prediction index,  $\Gamma$ , using the 3D pose predictions by:  $\Gamma = \arg \min_{\gamma=1, \dots, N} \sum_{T=t+1}^{t+\zeta} D(O_T^{P,\gamma}, GT_T^P)$ . Then use this index to calculate the ADE for both the 3D pose and the trajectory:  $ADE_p = \sum_{T=t+1}^{t+\zeta} D(O^{P,\Gamma_T}, GT_T^P)$  and  $ADE_h = \sum_{T=t+1}^{t+\zeta} D(O^{H,\Gamma_T}, GT_T^H)$ .

**3. Final Displacement Error (FDE):** Mean squared distance between the final ground-truth and the closest final prediction. Similar to ADE, we first calculate the closest final prediction index by  $\mathfrak{J} = \arg \min_{\gamma=1, \dots, N} D(O_{\gamma,t+\zeta}, GT_{t+\zeta})$ . Then we calculate the FDE for both the 3D pose and the trajectory:  $FDE_p = D(O_{t+\zeta}^{P,\mathfrak{J}}, GT_{t+\zeta}^P)$  and  $FDE_h = D(O_{t+\zeta}^{H,\mathfrak{J}}, GT_{t+\zeta}^H)$ .

**4. Multi-modal ADE (MADE):** To evaluate our system’s ability to generate multi-modal predictions, we used the multi-modal version of ADE [97, 98]. The MADE uses multi-modal  $GT$  future motions by grouping similar past motions.

**5. Multi-modal FDE (MFDE):** Similar to MADE, The MFDE is the multi-modal version of FDE [97, 98].

### 2.2.3.1 3D Pose Experiment

In the first experiment, we evaluate our 3D Pose generation module. Here, we compare our method against two baselines. The first one is DLow [97], the state-of-the-art in diverse human 3D pose forecasting which outperforms all the currently known methods to the best of our knowledge. The authors of DLow [97] provide detailed comparisons to several other state-of-the-art methods and show that DLow outperforms them. We will omit comparisons to these other methods and compare directly with DLow. The second baseline is STPOTR [56], our other work focuses on 3D human motion prediction for robotics applications. STPOTR predicts only one future motion so we cannot use it for multi-modal evaluation.

Approach	APD	ADE	FDE	MADE	MFDE
	$\uparrow$	(m) $\downarrow$	(m) $\downarrow$	(m) $\downarrow$	(m) $\downarrow$
DMMGAN (Ours)	<b>5.81</b>	<b>0.44</b>	<b>0.52</b>	<b>0.54</b>	<b>0.60</b>
DLow	5.53	0.48	0.61	0.55	0.63
STPOTR	NA	0.50	0.75	NA	NA

Table 2.1: Comparison of our systems versus two baselines for the 3D Pose experiment.

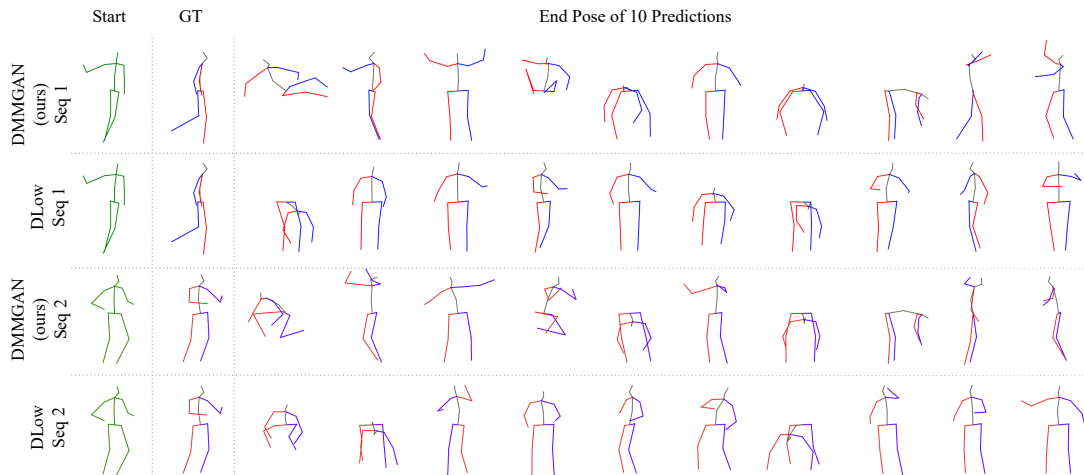


Figure 2.11: Qualitative results of 3D pose predictions comparing our method, DMMGAN, to DLow in terms of diversity.

Table 2.1 shows the results of this experiment. Our method outperforms both of the baselines and achieves the highest diversity while keeping both ADE and FDE lowest. Our method also has the highest coverage of the multi-modal ground-truth (MADE and MFDE). Also, we visually evaluate our method against DLow, in Fig. 2.11, we visualize the 10 end poses of our predictions versus the DLow for 2 random samples. In both methods, we can see a comparable accuracy against the ground-truth data (GT). Although the diversity of our method is close to DLow, closer examination of Seq 1 shows that our method predicted sitting down, crouching, lying down, walking left and right, while DLow has qualitatively less diverse samples.

### 2.2.3.2 Full 3D Motion Experiment

The second experiment evaluates our full system. In order to compare our system with a state-of-the-art diverse 3D motion model, we repurposed and retrained DLow [97] to forecast the human’s trajectory by adding the hip joint to the joints **Adapted DLow** predicts. We also compare our system (DMMGAN) with STPOTR [56], which is one of the few works that provides full 3D motion (pose and hip) prediction. We also include two variations of our models as an ablation study. The first model is **MMGAN** which is our full system

trained without the *similarity loss* and the second one is called **HipOnly** which is our *Hip Prediction* module without the 3D pose prediction inputs. The HipOnly model evaluates the impact of the predicted 3D pose data on the accuracy of the trajectory estimation. (Fig. 2.10b without the right 3D pose Transformer encoder).

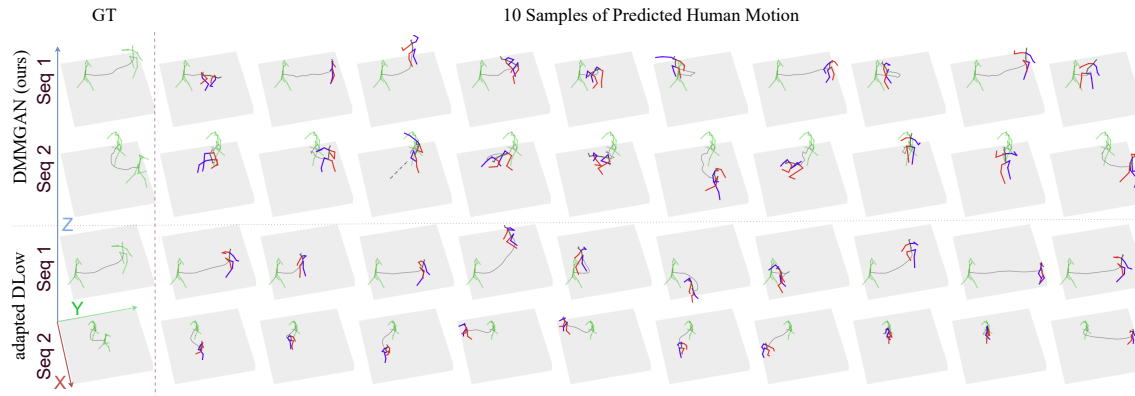


Figure 2.12: Qualitative results of 3D motion predictions comparing our method, to DLow in terms of diversity.

Approach	APD $\uparrow$	ADE (m) $\downarrow$		FDE (m) $\downarrow$		MADE (m) $\downarrow$		MFDE (m) $\downarrow$		
		Pose	Trajectory	Pose	Trajectory	Pose	Trajectory	Pose	Trajectory	
Adapted DLow	5.55	0.483	0.195	0.621	0.457	0.563	0.306	0.649	0.553	
STPOTR	NA	0.507	0.139	0.758	0.277	NA	NA	NA	NA	
ours:										
DMMGAN (ours)	<b>5.81</b>	0.443	0.122	0.520	0.228	<b>0.540</b>	<b>0.192</b>	<b>0.597</b>	<b>0.342</b>	
MMGAN	2.01	<b>0.422</b>	<b>0.104</b>	<b>0.494</b>	<b>0.190</b>	0.589	0.198	0.665	0.360	
HipOnly	NA	NA	0.156	NA	0.306	NA	NA	NA	NA	

Table 2.2: Comparison of our systems versus two baselines for the full 3D motion experiment.

Based on the result of this experiment (Table 2.2), our method outperforms the baselines by achieving the highest diversity while keeping the ADE and FDE lowest. In Fig. 2.12, we compare our prediction versus Adapted DLow and the ground-truth (GT) qualitatively<sup>6</sup>. In these examples, Adapted DLow predicted only walking movement while DMMGAN could capture more diverse motions.

The *HipOnly* model achieved a higher FDE and ADE compared to our model, which shows the benefit of using an attention-based 3D pose generator during trajectory forecasting. The results also highlight the impact of the *similarity loss* on the diversity of the predicted 3D motions. Our model without the similarity loss, MMGAN, achieved APD of 2 versus 5.8 for our full system. It is interesting to note that by removing the *similarity loss*, the model achieves a lower ADE and FDE with the cost of less diverse predictions.

<sup>6</sup>Please refer to <https://youtu.be/osJuFbtJsMg> for more examples.

## 2.2.4 Conclusion

We proposed DMMGAN, a novel method to predict diverse human motions. DMMGAN combines a generative adversarial network with Transformer based encoders to generate both the trajectory and the 3D pose of human motions. DMMGAN is capable of simultaneously predicting multiple plausible future human motions.

Our implementation outperformed the previous state of the art in diverse human 3D pose prediction while also predicting the human’s trajectory.

## 2.3 STPOTR: Simultaneous Prediction of Human Trajectory and Pose with Transformers

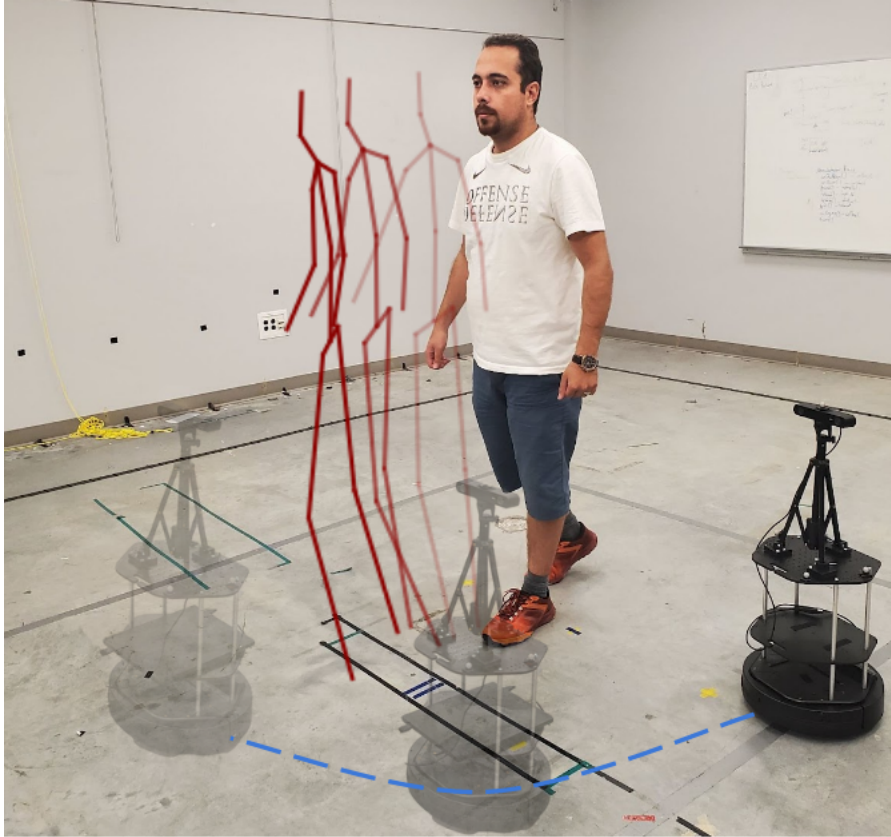


Figure 2.13: Robot follow-ahead via human motion prediction

In the second human motion prediction work, we introduce an accurate and fast non-autoregressive transformer for simultaneous prediction of human trajectory and body poses. We expand the capability of robots to perform the follow-ahead task and variations of this task through development of a neural network model to predict future human motion from an observed human motion history.

We propose a non-autoregressive transformer architecture to leverage its parallel nature for easier training and fast, accurate predictions at test time. The proposed architecture divides human motion prediction into two parts: 1) the *human trajectory*, which is the hip joint 3D position over time, and 2) the *human pose* which is the 3D position of all other joints over time with respect to a fixed hip joint.

We propose to make the two predictions simultaneously, as the shared representation can improve the model performance. Therefore, the model consists of two sets of encoders and decoders. First, a multi-head self-attention module is applied to encoder outputs to

improve the human trajectory. Second, another multi-head self-attention module is applied to encoder outputs concatenated with decoder outputs to facilitate the learning of temporal dependencies. Our model is well-suited for robotic applications in terms of test accuracy and speed, and compares favorably with respect to state-of-the-art methods in terms of other metrics. We demonstrate the real-world applicability of our work via the *Robot Follow-Ahead* task, a challenging yet practical case study for our proposed model. We go over these results in Section 3.3. Our code and data are available at the following Github page: <https://github.com/mmahdavian/STPOTR>

In summary, our contributions are as follows:

- We solve the robot follow-ahead task with better performance with respect to previous methods and demonstrated multiple benefits for taking human body pose into account in the robot follow-ahead task, including new following behaviors that were not previously possible.
- To the best of our knowledge, we are the first to simultaneously predict human pose and trajectory and utilize the results in a real-world robotic scenario.
- We achieve a reasonable accuracy for both human trajectory and body pose predictions with respect to the state-of-the-art methods.
- Using ablation studies, we show that our proposed shared attention module allows human body pose information to improve human trajectory prediction.
- We demonstrate our method in numerous human-following tasks on a real robot.

### 2.3.1 Methodology

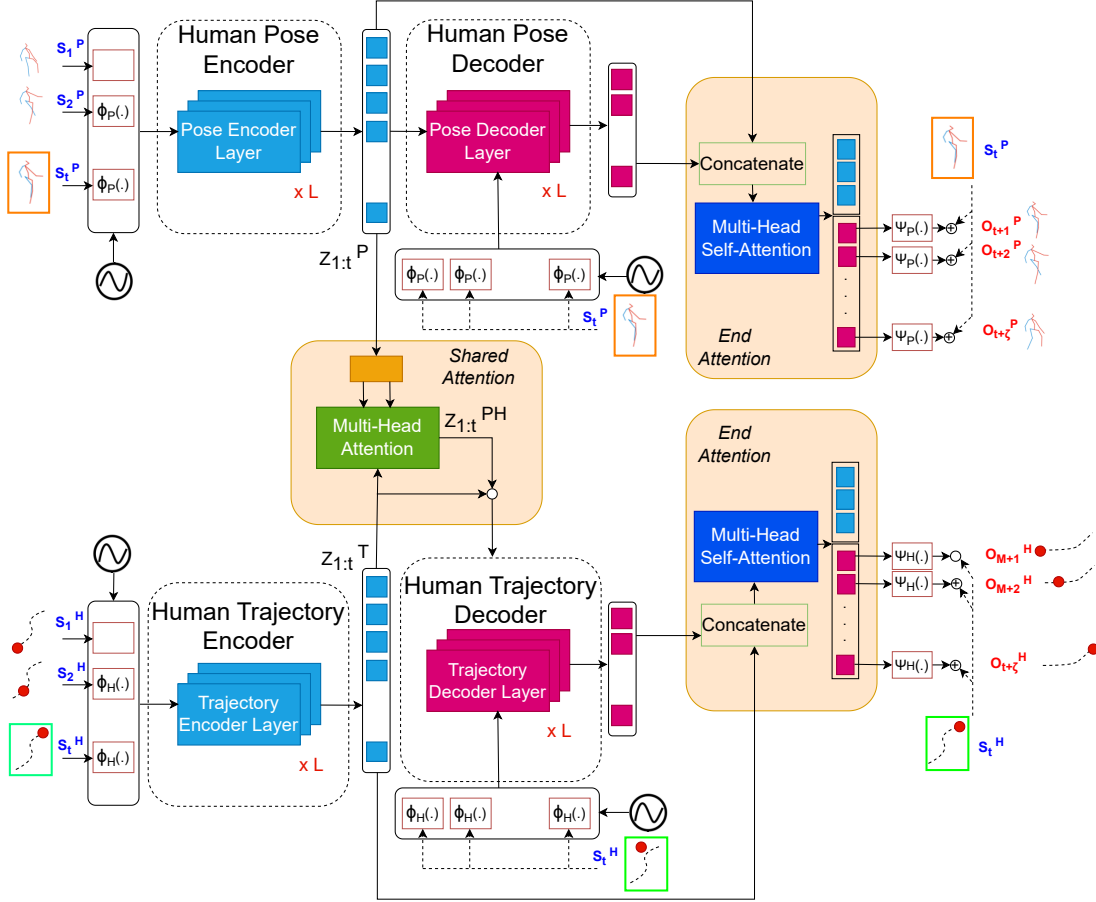


Figure 2.14: Our model structure. It simultaneously predicts human poses and trajectories from an observed 3D human joints sequence. It is constructed from two non-autoregressive transformers for pose and trajectory predictions as well as a Shared Attention module to share knowledge between the two for better predictions. An End Attention module is added to the end of each decoder for better modeling of the temporal dependencies. The blue-colored frames show the input sequence or frame and the red ones show the output. The rectangular frames show that the same frame (last input pose) is copied and used as the decoder input sequence and as a residual for decoder output.

We divide the prediction task into two interdependent parts. The first part is to predict the future 3D hip trajectory,  $O_{t+1:t+\zeta}^H$ , from previously observed ones,  $S_{t-\alpha:t}^H$ . The hip is the standard joint position for representing the 3D human position purpose [1, 23, 74]. Next, as the second part of the problem, we aim to predict the future 3D human pose sequence,  $O_{t+1:t+\zeta}^P$ , from the observed ones,  $S_{t-\alpha:t}^P$ . Here a 3D human pose is defined as all joints' relative 3D position with respect to the fixed hip joint. The superscript  $H$  and  $P$  denote the human trajectory and pose sequence, and  $\zeta$  is the forecast duration. We aim to solve the two parts simultaneously, as the features transferred in between can improve the predictions.

In this work, we propose to solve this problem by conditional sequence modeling where the goal is to train the set of parameters of a non-autoregressive transformer.

In our model, we follow the main structure of the autoregressive [86] and non-autoregressive [60] transformers with multiple improvements and adjustments. Fig. 2.14 shows the structure of our model architecture. The model simultaneously predicts the human pose (upper section) and trajectory (lower section). The encoders and decoders are composed of  $L$  layers, each with the structure in [86], containing multi-head, self- or encoder-decoder attention layers as well as fully-connected layers. The encoders receive a sequence of 3D human poses  $S_{t-\alpha:t}^P$  or hip trajectory  $S_{t-\alpha:t}^H$ , and generate the two sequences of embeddings  $Z_{t-\alpha:t}^P$  and  $Z_{t-\alpha:t}^H$ . While the main structure of the transformer model learns the temporal dependencies, two networks are added ( $\phi$  and  $\psi$ ) as pose encoder (GCN-based) and pose decoder (MLP-based) to identify the spatial dependencies between the joints in each frame. The pose and trajectory encoding networks,  $\phi_P$  and  $\phi_H$ , are GCNs that learn the spatial relationship between the body joints. The weight of the graph edges represented by the adjacency matrix is used to compute embeddings of dimension  $D$  for the human pose and human trajectory vectors in the input sequences  $S_{t-\alpha:t}^P$  and  $S_{t-\alpha:t}^H$ . In order to modify the model to perform in a non-autoregressive manner, the last frame of input sequences,  $S_t^P$  and  $S_t^H$ , were copied and used as *query sequences* for decoders input. The model generates pose and trajectory predictions  $O_{t+1:t+\zeta}^P$  and  $O_{t+1:t+\zeta}^H$ , in parallel using the networks  $\psi_P$  and  $\psi_H$ , from the decoder outputs and a residual connection containing the query sequences. Therefore, the decoders learn the offsets with respect to the last seen frame.

One of the benefits of our architecture is that we can share the representation between human pose and trajectory prediction modules. In order to fully benefit from the combination of human poses and hip trajectory, we have added a multi-head attention module called *Shared Attention* to apply attention between pose and trajectory encoder outputs as shown in the middle of Fig. 2.14. First, we apply a linear layer to the pose encoder embedding,  $Z_{t-\alpha:t}^P$ , to change the dimension from pose to trajectory embedding size. Then, we pass it with a copy as well as the trajectory encoder embedding,  $Z_{t-\alpha:t}^H$ , to the multi-head attention module. We then add the multi-head attention output,  $Z_{t-\alpha:t}^{PH}$ , with the hip trajectory encoder output to use it in the hip trajectory decoder. The added multi-head attention can improve the hip trajectory prediction compared to solely relying on hip trajectory history, since the human pose changes are related to how humans move overall. In Section 2.3.2.7 we investigate how this attention module can help our model predict more accurately.

In addition, we have added a multi-head attention layer to the end of each decoder called *End Attention*. This module can help the model to better learn the temporal dependencies between all frames. We concatenate the pose and trajectory encoders output with decoders output and apply a self-attention module. Then we output the last encoded features with the same length as the target sequence length. To convert them to the actual sequence of future 3D human pose  $O_{t+1:t+\zeta}^P$  and hip trajectory  $O_{t+1:t+\zeta}^H$ , the model uses a



Table 2.3: Analytical comparisons between our developed model and the baselines introduced in [97] and [69] in terms of *ADE* and *FDE* for both human pose and trajectory predictions and Inference Duration (ID)

Method	$ADE_{Pose}$ (m)	$FDE_{Pose}$ (m)	$ADE_{Traj}$ (m)	$FDE_{Traj}$ (m)	ID (msec)
DLow [97]	0.48	0.62	0.19	0.45	20
DMMGAN [69]	0.44	0.52	0.12	0.23	100
HipOnly [69]	NA	NA	0.15	0.30	18
Ours	0.50	0.75	0.13	0.27	25

pose and trajectory decoder ( $\psi$ ). We discuss the impact of this module in the ablation study presented in Section 2.3.2.7.

## 2.3.2 Human Motion Prediction Experiments

In this section, we first describe the dataset used to train our model, implementation details, baselines, and metrics. Then, we show the performance of our human motion prediction method with respect to baselines. Finally, we present the results of ablation studies to demonstrate the effectiveness of different parts of our proposed architecture.

### 2.3.2.1 Dataset

To train the human motion prediction model, we used the well-known and standard Human3.6M dataset [35]. It contains the 3D joint position of seven actors performing 15 activities, including walking, sitting, and smoking. Traditionally, this dataset has been used as a benchmark for human pose prediction [55], but we utilize it for human trajectory prediction as well. As explained before, we extracted the hip trajectory of each actor for the human trajectory prediction and all other joints’ relative position with respect to the fixed hip for human pose prediction. Conventionally, for this dataset, one reduces the frame rate from 50 Hz to 25 Hz [3, 60, 97]; however, we used 10 Hz, a more suitable frame rate for robotic purposes as it is fast enough, reduces the complexity of our model, and speeds up predictions at test time. Also, we followed the standard input and output duration of our human pose prediction baseline, DLow [97] which are 0.5 sec (5 frames) for input and 2 sec (20 frames) for the output.

### 2.3.2.2 Training

We used Pytorch as our deep learning framework. The model was trained with AdamW [53] for 250 epochs with a learning rate of  $10^{-4}$  and a batch size of 16. The model was trained after 50K steps with warm-up scheduled in the first 10K steps. During warm-up, the learning rate gradually increases from zero to  $10^{-4}$  which increases training stability.

### 2.3.2.3 Model Hyperparameters

Based on experience, we set the embedding dimensions to  $D_{Pose} = 512$  for pose prediction and  $D_{Traj} = 64$  for trajectory prediction. Also, the fully-connected dimension in our encoders and decoders was set to 2048. The encoders and decoders each contain four layers of pre-normalized [94] multi-head attention modules with eight attention heads. Here, “pre-” or “post-normalized” refers to whether the normalization layer is the first layer in the multi-head attention module or the last one.

### 2.3.2.4 Baselines

As our baselines, we compared our work with two state-of-the-arts in human pose and trajectory predictions suitable for robotic purposes. We used DLow [97] as our first baseline as a fast and accurate method in human pose prediction. This method has the best performance for pose prediction out of all other methods except for DMMGAN [69]. Since this method only predicts human poses at 25 Hz, we retrained it for simultaneous human pose and hip trajectory predictions at 10 Hz with hip joint motion added to the predictions to be able to compare directly. As a more accurate but slower method, we compared our results with DMMGAN [69] that simultaneously predicts human pose and trajectory for robotic purposes. As another baseline for trajectory predictions, we compare our method with a simple GRU-based method called *Hip Only* introduced as a trajectory prediction baseline in [69]. In this baseline, a GRU is applied to the human trajectory after passing through a transformer encoder. To the best of our knowledge, these are the only available methods to compare with simultaneous human pose and trajectory predictions suitable for real-world robotic purposes. Other prior methods – and DLow [97] without any modifications – either only predict human body pose relative to the fixed hip or heading [3, 60] without predicting the hip trajectory in 3D space or are not fast enough.

### 2.3.2.5 Metrics

In order to compare our results with the baselines, we use the conventional Average Displacement Error (ADE) and Final Displacement Error (FDE) [55] metrics. ADE is the average of the  $L_2$  distance over all time steps between ground truth and prediction. FDE is the  $L_2$  distance between the last ground truth and predicted frames. We compared both metrics for both pose and trajectory predictions. As another important factor for real-time robotic purposes, we compared the algorithms’ speed at test time.

### 2.3.2.6 Results

Table 2.3 quantitatively compares our method to the baselines. The achieved  $ADE_{Pose}$  is comparable to the state-of-the-art DLow [97] paper and DMMGAN [69]. Also, we have

achieved better results for trajectory prediction with respect to DLow and *Hip Only* [69]. All training and testing were done on a laptop with an Intel CPU Core i9-9980HK CPU and RTX 2080 Max-Q GPU. Due to the non-autoregressive nature of our method, we were able to achieve much better computation speed at test time compared to DMMGAN, and similar computation speed compared to DLow. However, our method has slightly worse but comparable  $ADE_{Pose}$  and  $FDE_{Pose}$  with respect to DLow and  $ADE_{Traj}$  and  $FDE_{Traj}$  with respect to DMMGAN [69]. This result was expected as discussed in [60]: The non-autoregressive nature of the model reduces the model’s capability in modeling correlation between frames which increases model error. Another reason is that DLow and DMMGAN predict multiple possible predictions for an input sequence and  $ADE_{Pose}$  and  $FDE_{Pose}$  are calculated for the most similar predicted sequence to the ground truth; thus, they are somewhat similar to ensemble methods in spirit.

Note that for our robotic follow-ahead task, to work smoothly, we need to make the predictions, pre-processing (3D human pose estimation) and post-processing (robot trajectory planning) in less than 100 msec as the frame rate of the model input is 10 Hz. Therefore, the DMMGAN [69] was not a suitable choice for this task. On the other hand, DLow’s trajectory prediction accuracy was not adequate. Therefore, our method provided the most suitable model in terms of both accuracy and speed. For robotic purposes, our accuracy is adequate as demonstrated in Section 3.3.2, and the fast computation speed at test time as needed. Also, Fig. 2.15 qualitatively shows three samples of the predicted motion with respect to the ground truth.

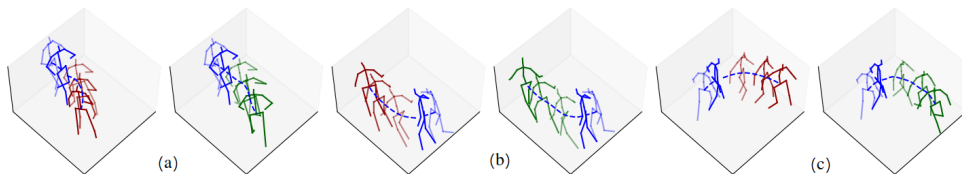


Figure 2.15: Three samples of the predicted motion vs. ground truth. On each couple of figures (a to c) the left one shows the predicted motion given an observed sequence and the right one shows the ground truth. The blue-colored skeletons show the input sequence and the red and green ones show the model predictions and ground truth, respectively. Also, the trajectory of the hip is shown with dashed black lines.

### 2.3.2.7 Ablation Study

We performed an ablation study to evaluate the training process and the effectiveness of different modules used in our model. To show one of the major advantages of our method, we discuss the effect of the *Shared Attention* module used for better trajectory predictions. We compare the current results with the cases that 1) no such module is applied (*No Shared Attn*) and 2) the shared attention module is applied only for pose predictions (*Shared Attn-Pose Only*). Also, we study the effect of the *End Attention* module added to the end of

Table 2.4: Our ablation study analytical comparisons

Model	$ADE_{Pose}$ (m)	$FDE_{Pose}$ (m)	$ADE_{Traj}$ (m)	$FDE_{Traj}$ (m)
<i>No Shared Attn</i>	0.50	0.75	0.16	0.33
<i>Shared Attn-Pose Only</i>	0.51	0.76	0.16	0.33
<i>No End Attn</i>	0.52	0.77	0.18	0.33
<i>Post Normalized</i>	0.51	0.76	0.17	0.32
Ours	<b>0.50</b>	<b>0.75</b>	<b>0.13</b>	<b>0.27</b>

each decoder which aims to better model temporal dependencies by removing this module (*No End Attn*). Finally, we compare the achieved accuracy with the post-normalized [94] multi-head attention modules.

As one can see in Table 2.4, the shared attention module has improved the trajectory prediction by incorporating the human pose representation while predicting trajectory. The same module degraded the pose prediction and we believe there are two reasons for it. First, in some of the dataset motions, the body limbs have random movements, such as random hand waving during walking, which makes the predictions harder. Second, while the body pose can be informative for predicting the hip trajectory, the reverse may not be true, as given a hip trajectory, there are often still a lot of degrees of freedom for the body pose. Also, the end attention module applied to concatenation of encoder and decoder outputs improved the model performance by better modeling the temporal dependencies between input and output frames. In addition, the post-normalized structure for multi-head attentions was not able to perform as well as the current pre-normalized version.

### 2.3.3 Conclusion

Our work on STPOTR focused on real-world robotic applications of simultaneous human trajectory and motion prediction. We utilized two parallel non-autoregressive transformers, which we modified for our specific purposes. Through evaluation against various baselines, we obtained promising results in terms of speed, and the accuracy of human motion prediction, which suggests that our method is well-suited for robotic purposes. In particular, our use of the non-autoregressive method led to a 4x increase in speed performance compared to DMMGAN. A section of the upcoming chapter will focus on a real world application of STPOTR and compare its performance with the current state of the art.

## 2.4 Conclusion

In this Chapter, we design two human motion prediction models for robotics applications. The first model is designed to address the issue of multimodality in human motions, while the second one is focused on improving the speed of the prediction process and making it suitable for various robotics applications.

The first model, DMMGAN, is a new approach to predicting diverse human motions. It uses a combination of a generative adversarial network and Transformer-based encoders to generate both the 3D pose and trajectory of human motion. Our implementation outperformed the previous state of the art, DLow [97], in diverse human 3D pose prediction while also predicting the human’s trajectory.

In the second model, STPOTR, a simultaneous human trajectory and motion prediction was presented for real-world robotic applications. The model utilizes two parallel non-autoregressive transformers, which were modified for this purpose. The results showed a reasonable level of performance in terms of speed, pose, and trajectory prediction compared to other baselines, making it suitable for use in robotics applications.

## Chapter 3

# Human Following

Rapid technical developments will bring robots into our everyday lives. There are various applications where robots could usefully follow a human user around to assist them, for example, a golf caddy or self-driving luggage. Notably, Boston Dynamics’ LS3 legged robots (unpublished, derived from BigDog [92]) had a well-developed person-following capability to act as load-carrying mules.

In human-robot interaction, robots are often required to stay close to a human user. This is necessary for a variety of situations, such as capturing physical activity on video or monitoring elderly individuals. Different methods of following a human include following from behind, following in front, and following side by side [29, 32]. While following from behind is a well-studied area, following a user from the front presents a greater challenge. To follow a person from behind, one can use a Proportional Integral Derivative (PID) controller to keep the person at the center and maintain a safe distance [47]. In contrast, following a person from the front requires the robot to predict the person’s future trajectory and navigate to a point on that trajectory while keeping a safe distance.

Research shows that in following scenarios, the user often glances behind for reassurance that the robot is within a safe distance. On the other hand, being in front of the user can provide benefits such as improved safety and more convenient interaction. This is seen in applications such as autonomous shopping carts, self-driving luggage, and autonomous guide dogs.

In our research, we propose two methods for human following. The first method is a hybrid solution that combines deep RL and classical trajectory planning for the front-following application. Our deep RL module makes high-level decisions by implicitly estimating the human trajectory and producing short-term navigational goals. The trajectory planner then executes these goals at a low level to smoothly navigate the robot in front of the user. We use curriculum learning in the deep RL module to achieve high returns efficiently. The second method is a general human-following system that uses a fast non-autoregressive human motion prediction model to learn the human’s goals explicitly. In STPOTR, we decouple the robot’s decision-making process from human motion prediction, whereas in LBGP, we

adopt a one-shot approach that simultaneously calculates the robot's goal and performs human motion prediction.

This chapter is based on the two papers we published at the IEEE International Conference on Robotics and Automation (ICRA 2021 and ICRA 2023) [56, 71]

## 3.1 Related Work

### 3.1.1 Categorize of human following

Person following has been studied for ground [68, 73, 89], aerial [33, 54] or even underwater environments [37, 99]. Following from behind is the dominant scenario in these studies.

In classical methods, the person following problem has been broken down into three sub-modules: localization of the user, pathfinding, and trajectory tracking [47, 88]. Recent years have seen a rise in the popularity of learning to perform navigational tasks directly from sensor inputs [73]. These techniques involve learning the task in simulation first and then possibly transferring the policy to the real world or generalizing the policy to unseen environments [24].

One branch of human following deepening on the social force model (SFM) introduced by Helbing and Molnar in [28]. SFM models human motion in terms of attractive and repulsive forces. In SFM, repulsive forces refer to forces created by other people or environmental obstacles, while attractive force is toward the user’s desired goal. A line of work in the human following is based on the SFM idea. [18] is one of the first examples of human following that incorporates the SFM concept. Their work considers forces between human-human, human-robot, and human-obstacle. A genetic optimization algorithm is used to learn the parameters of the human-robot forces. Their assumptions include knowing pedestrians’ and robots’ goals. Repiso et al. improve this model by using a predictive model to predict the desired goal of the robot using an SFM [78]. They calculate the desired goal based on the person’s movement, the group’s movement, and the other people around (moving obstacles). Their model attempts to learn a human’s desired goal by combining the companion force and hand-picked environment structure that may not always hold. For example, it can never model the multi-modality of human behavior or be generalizable to different environments.

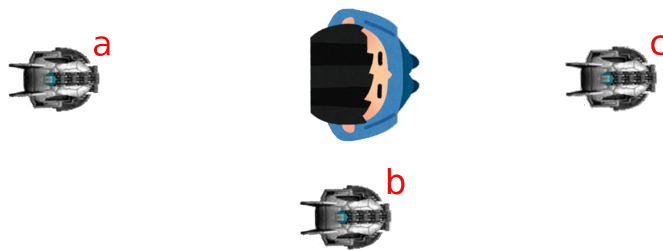


Figure 3.1: Depending on the robot’s local position relative to the human, the person following can be categorized into (a) “behind the leader”, (b) “side-by-side”, (c) “in front of the leader” [29].

Ho et al. [29] divide the person following into three categories based on the robot’s local position with respect to the human: 1) following behind the leader, 2) side-by-side with the leader, and 3) ahead of the leader. An implementation of the following behind the leader



can be accomplished with a simple proportional controller that attempts to keep the person at a fixed distance and in the middle of the sensor field of view. The other two tasks are considerably more challenging as they require prediction of the user’s movements [64]. For example, for smooth following-ahead through an intersection, the robot needs to predict which direction the user will take. In the following, we go over these three categories.

### 3.1.1.1 Following behind

Most of the previous work on the person following has involved following a user from behind. In [47], Leigh et al. present a human-centered tracking framework that classifies laser data as human or not human. The detected person positions are tracked using a Kalman Filter, and then they apply separate PID controllers to obtain the angular and linear velocities of the robot. An interesting resource-limited example is Yao et al. [96], where the Georgia Tech Miniature Autonomous Blimp detects and follows a person using a monocular camera. They use a Haar face detector and a KLT feature tracker to track the user. In another work, Sun et al. [84] present a following behind the leader behavior by using an SFM based on the human goal, wall, other pedestrians, and other obstacles forces. In spite of the fact that their model is capable of factoring in pedestrian comforts using SFM, they assume a really simplistic model of a pedestrian that would not be applicable in real-world scenarios.

### 3.1.1.2 Following ahead

Behavioral experiments suggest that in following behind scenarios, the user frequently looks behind out of curiosity or to ensure the robot is within a safe distance [39]. Conversely, following in front can assist a person in different applications. Consider an autonomous shopping cart, self-driving luggage, or autonomous guide dog; in all these applications, it is best if the robot is in front of the user. The user not only feels safer but also can interact with the robot more conveniently.

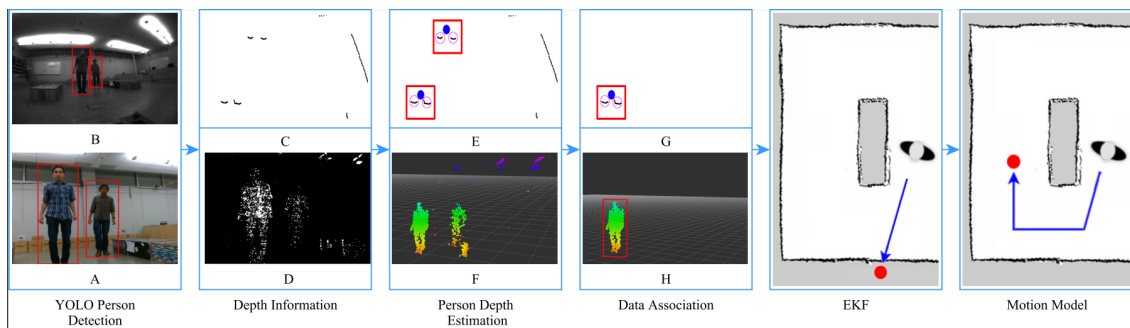


Figure 3.2: In [70], via an EKF approach and hand-designed human models, they designed a following ahead system.

In one of the first efforts, Ho et al. [30] assumed a nonholonomic human model and estimated the human’s linear and angular velocity via a Kalman filter. Their proposed motion planner did not perform well for some relatively complex scenarios. Cifuentes et al. [14] propose an approach based on a human gait model that uses a wearable Inertial Measurement Unit (IMU) for estimating orientation. In [29], they calculate human orientation using a Kalman filter with a nonholonomic human model for estimating human linear and angular velocities. At the same time, a special-purpose robot motion controller aims to align the human-robot poses such that the robot follows from the front. Eui-Jung et al. [39] present a holonomic motion model for tracking a human while staying ahead. In [85], Tominaga et al. present another front-following system using simple visual servoing that tries to keep a person (marked with an AR tag) in the center of the robot’s view. The person’s heading is not considered, and the robot can easily lose the person at sharp turns. A more recent work by Moustiris et al. [64] describes a front-following model that uses a modified dynamic window planner without considering the current heading of the person, which is important information when predicting motion relatively far into the future. Their method is extended in [65], where they assume that the user’s relative heading can be estimated by how far the user is offset from the middle of the robot’s field of view. Although this cue may often be helpful, it cannot predict how the user’s heading will change due to interactions with walls and other obstacles.

The authors in [70] developed an Extended Kalman Filter (EKF) approach by combining 2D Lidar and a fish-eye camera to detect and track a person. A velocity-based heading estimator and human model that accounted for obstacles helped to correct the EKF predicted position. However, this method did not account for human body pose and thus are limited in its use cases. For example, the EKF method cannot perform well when the human is nearly stationary.

### 3.1.1.3 Follow side-by-side

The last category of human following is side-by-side following. It is not uncommon for people to walk side-by-side in a group of people since they communicate together [63]. Walking side-by-side can facilitate communication while providing personal space and eye contact. In comparison to following behind, making a social companion robot that follows side-by-side can be an excellent alternative. For example, for a companion robot in a shopping mall or at an elderly care center, it is expected that the robot talks to people while they are heading to their destination. In the same way as following ahead, following side-by-side is challenging as the robot must know or predict the user’s trajectory, and if the prediction is incorrect, the robot will be stuck behind the user.

An early version of the side-by-side following is introduced in [41]. They projected their future position based on the person’s current velocity and orientation. The technique works well for walking straight but can fall apart if people change their directions. Morales et al.

provide a follow side-by-side model that combines the planning of the human and robot [62]. A major shortcoming of their work is that the robot must know the human’s destination, which is not known in real life. Similarly, in [61], they have a side-by-side following method for a shopping environment. They also assume to know the destination of the human. Murakami et al. propose to use a two-state model for an indoor laboratory environment [66]. Their environment only includes one intersection with four predefined sub-goal positions. Their method simply moves forward and, as soon as it gets to the intersection, uses an angle-based method to pick one of the predefined goals. Their method can only work for simple scenarios and cannot scale to real-world cases. They perform a follow-up study in [40] to remove the need to know the predefined goals. They use the method proposed by [34] to find the position of sub-goals using trajectories of the pedestrians. These sub-goals are calculated statistically for their environment using offline data from 6088 trajectories. But still, their method is pretty limited and cannot scale well to the different environments as it requires offline data of trajectories for each environment. They also use hand-picked features to pick these sub-goals, which may not hold well for different environments.

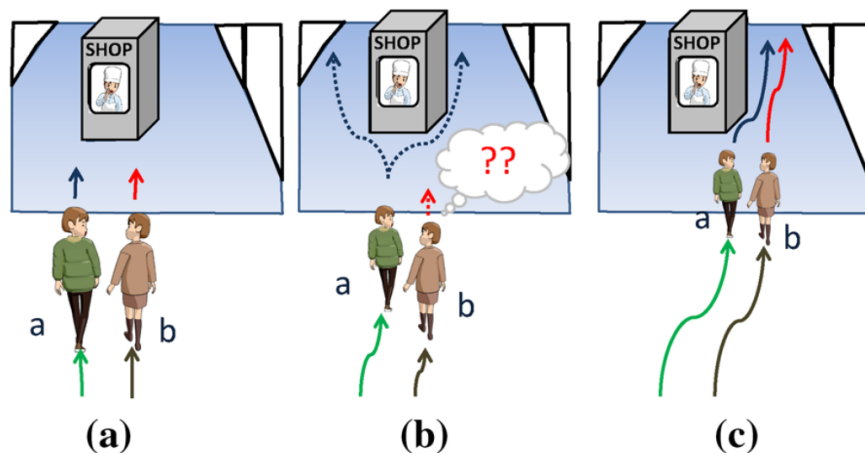


Figure 3.3: An intersection with two people walking side-by-side. a) going straight, b) leader guiding, c) inferring routes [40].

In a recent study, Gil et al. propose combining SFM with RL. They calculated the robot’s linear and angular velocities by combining the velocity of an RL agent with an SFM model. Here they use the SFM to have human-aware navigation by considering moving pedestrians and obstacles in their environment. The RL agent performs two tasks: point-to-point navigation and path following. In this case, they use DDPG as the RL agent and use the SFM only to avoid large obstacles since the repulsive force is around zero when the robot is far from them. Both tasks assume the robot knows the human’s destination, which simplifies solving the problem and makes it less applicable to a real world application.

### 3.1.2 Human following using RL

In recent years, a variety of research shows the capability of deep reinforcement learning (RL) to solve challenging game problems [10, 80]. Applying it to robotic problems can help to address navigational tasks while considering the user intents [11, 44]. Deep RL can implicitly account for robot dynamics and enable a continuous interaction between the robot and its environment. In the staying-in-front problem, deep RL can also implicitly predict a person’s future trajectory and continuously update the predictions to provide a smooth real-time experience for the user.

Several studies used deep RL for related navigational tasks. Dewantara et al. proposed a guiding behavior that optimizes parameters of a social force model using Q-Learning [17]. In [11], Chen et al. presented a relational graph deep RL approach for robotic crowd navigation [11]. Using this relational graph, they encoded higher-order interactions between agents and used it to anticipate the future. Besides, A curriculum learning approach has been used to increase the efficiency of RL training. Narvekar and Stone formulated a curriculum sequencing problem as a Markov Decision Process [67]. They show how curriculum learning can reduce training time. Kulhanek et al. presented another RL-based navigation agent [44], which learns to navigate in an environment using only the raw images. They proposed pre-training the network by transferring the learned policies from one environment to another and gradually increasing the environment’s complexity. Bansal et al. proposed a navigational framework for combining optimal control and learning [6]. Their learning-based perception module produces a series of way-points that guides a robot toward the goal. In a recent study, we propose the Learning Based Goal Planning (LBGP) approach to address the problem of staying in front of the user. LBGP is a hybrid approach that combines deep RL and classical trajectory planners. Our results show that combining deep RL with classical methods can greatly improve performance while maintaining its safety. Additionally, we demonstrate the benefits of using curriculum learning to train the agent on increasingly challenging human motions. This method is more efficient and achieves higher returns compared to training without a curriculum.

### 3.2 LBGP: Learning Based Goal Planning Approach for Autonomous Following in Front



Figure 3.4: A mobile robot following-ahead of a user. The robot must predict the user’s trajectory to stay in the correct relative position. In each time step in our proposed approach, the robot considers previous states of the joint system to generate a goal (blue dot). Then a trajectory planner navigates the robot towards the goal (green line).

In our first work on the human following, we propose the Learning Based Goal Planning (LBGP) approach to address the problem of staying in front of the user. LBGP is a hybrid approach that uses the combination of deep RL and classical trajectory planners (see Figure 3.4). Our results show that combining deep RL with classical methods can greatly improve performance while maintaining its safety. We also show the benefits of using curriculum learning to train the agent on increasingly challenging human motions. Compared to training without a curriculum, our method trains the policy more efficiently while achieving a higher return. To generalize our model to unseen and real world inputs, we add a Gaussian noise to our observations.

We demonstrate favorable results in simulation and real world experiments compared to previous work. Our ablation studies show the benefits of our hybrid approach and curriculum. In particular, we show the effectiveness of our hybrid approach through zero-shot transfer of the policy trained in simulation to the real world. Example of our system can be found in the supplementary video<sup>1</sup>. In summary, our main contributions are as follows:

- We combine a classical robotic trajectory planner with deep RL to improve the safety and generalizability of our system.
- We use curriculum learning to reduce the training time while improving the final return.

<sup>1</sup><https://youtu.be/XSOudPFPMmA>

- By evaluating our system in the simulation using a Clearpath Jackal robot and in the real world using a Turtlebot 2 robot, we show that our system can be more reliable and efficient for front-following compared to an End-to-End learning or a purely classical hand-crafted approach.
- We demonstrate that unlike the End-to-End learning approach, the policy trained using our method can directly transfer to the real world without any re-training.

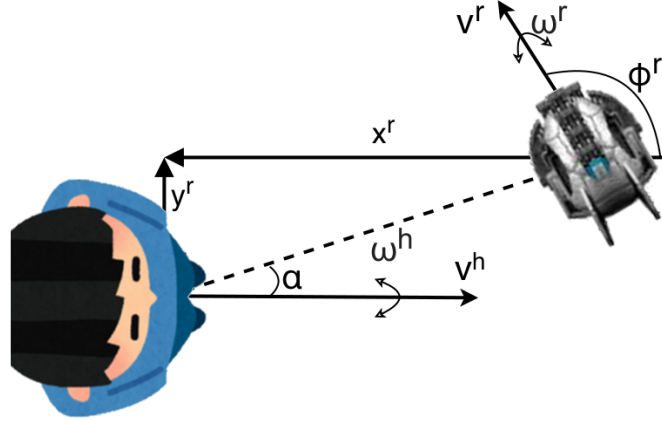


Figure 3.5: Our relative coordinates system

### 3.2.1 Problem Setup

In this work, we study the problem of keeping an autonomous robot in front of a walking person. We assume an obstacle free environment in which the robot should avoid collision with the human. We represent the global state of the human and robot with  $(X_t^h, Y_t^h, \Phi_t^h, v_t^h, \omega_t^h)$ ,  $(X_t^r, Y_t^r, \Phi_t^r, v_t^r, \omega_t^r)$ , respectively.  $(X_t, Y_t)$  is the position,  $\Phi_t$  is the orientation,  $v_t$  is the linear velocity and  $\omega_t$  is the angular velocity at time  $t$ .

To make our approach transferable to real world and avoid over-fitting we use a relative state of the robot with respect to human, denoted  $z_t^r$ :

$$z_t^r = (x_t^r, y_t^r, \varphi_t^r) \quad (3.1)$$

$$\text{where } \begin{bmatrix} x_t^r \\ y_t^r \\ \varphi_t^r \end{bmatrix} = \begin{bmatrix} \cos(\Phi_t^h) & -\sin(\Phi_t^h) & 0 \\ \sin(\Phi_t^h) & \cos(\Phi_t^h) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_t^r - X_t^h \\ Y_t^r - Y_t^h \\ \Phi_t^r - \Phi_t^h \end{bmatrix}$$

For the purpose of calculating rewards, we define  $\alpha_t = \arctan(y_t^r, x_t^r)$  as the person-robot angle.

We also define a similar notation for the  $i$ th previous state of the human relative to their current state at time  $t$ :

$$z_{t-i}^h : (x_{t-i}^h, y_{t-i}^h, \varphi_{t-i}^h) \quad (3.2)$$

$$\text{where } \begin{bmatrix} x_{t-i}^h \\ y_{t-i}^h \\ \varphi_{t-i}^h \end{bmatrix} = \begin{bmatrix} \cos(\Phi_t^h) & -\sin(\Phi_t^h) & 0 \\ \sin(\Phi_t^h) & \cos(\Phi_t^h) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{t-i}^h - X_t^h \\ Y_{t-i}^h - Y_t^h \\ \Phi_{t-i}^h - \Phi_t^h \end{bmatrix}$$

As part of the observation, we consider a history of relative coordinates for both the robot and human. These relative coordinates are all respective to the latest position of the human. This relative system is visualized in Figure 3.5.

### 3.2.2 Method

Our key insight in this work is to combine a deep RL module with a classical trajectory planner. The agent uses our implementation of Deep Deterministic Policy Gradient (D4PG) [9] algorithm to generate a short-term navigational goal. A Time Elastic Bands (TEB) motion planner is used to navigate toward this goal, while treating the person as a dynamic obstacle. Crucially, our approach differs from typical policies trained with RL, which directly output an agent’s actions.

#### 3.2.2.1 Observations and Navigational Goals

The observation is a stack of robot and human relative states  $(z_t^r, z_{t-1}^h, z_{t-1}^r, \dots, z_{t-9}^h, z_{t-9}^r)$ , with  $t$  being the current time step (see equations (3.1) and (3.2)). We stack states up to the last 10 frames (at 5 FPS).

The quantities are continuous and scaled to  $[-1, 1]$ . In simulation, we capture all the variables from Gazebo simulator. In real world, it can be obtained by a motion capture system or human detection algorithms (e.g. YOLOv2 [77]) with RGB-D inputs (this approach was previously used in [68]). To improve the transferability of our approach to the real world, we add Gaussian noise to the observations in simulation.

The output of our policy network is a target position relative to the person. This position is a short-term navigational goal based on the implicitly estimated path of the user. We feed this output to the TEB local planner to navigate the robot in a smooth trajectory.

#### 3.2.2.2 Reward

We define the reward function such that the agent receives a higher reward if it stays in front of the person at a desired distance of 1.5 m and negative reward if it is far away, too close or behind the person. The reward is scaled to  $[-1, +1]$ . Figure 3.6 shows the reward function based on relative coordinates of the robot to human. Mathematically, the agent

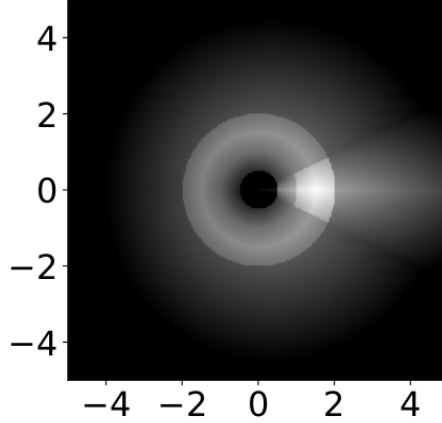


Figure 3.6: Reward based on the robot’s relative position to the person. Increasing from black (-1) to white (+1).

reward  $R$  consists of two parts,  $R_d$  for person-robot distance and  $R_o$  for person-robot angle, defined as follows:

$$R_d = \begin{cases} -1, & D < 0.5 \text{ or } 5 < D \\ -(1 - D), & 0.5 < D < 1 \\ 0.5(0.5 - |D - 1.5|), & 1.0 < D < 2 \\ -0.25(D - 1) & 2 < D < 5 \end{cases}$$

$$R_o = \begin{cases} 0.5((25 - |\alpha|)/25), & |\alpha| < 25 \\ -0.25|\alpha|/180, & |\alpha| > 25 \end{cases}$$

$$R = \min(\max(R_o + R_d, -1), 1)$$

where  $D$  is the distance between the robot and the person, and  $\alpha$  is the angle between the person-robot vector and the person-heading vector (person-robot angle, in short). We terminate the episode if the agent is too close ( $D < 0.5$  m) or far away ( $D > 5$  m).

### 3.2.2.3 Policy Training Environment

Our LBGP system is implemented in ROS [75] and trained in the Gazebo robot simulator [43]. We use a Turtlebot 3 burger robot to represent the person and a Clearpath Jackal robot as the robot. The person is controlled using our person motion model. We design a world in the Gazebo simulator with four replicas of an environment each containing one learning agent. Three of agents explore the environment while the last one exploits the policy. This setup is arranged to mitigate the exploration exploitation trade-off. The simultaneously collected trajectory data are added to replay buffer to update the model weights.



### 3.2.2.4 Curriculum Learning

To improve learning efficiency, we employ curriculum learning to train the agent in a series of tasks with increasing difficulty. These tasks are defined based on the human trajectory. We start with a straight line and move to more difficult trajectories as we go further in the training. In our curriculum, there are four difficulty levels: straight, circles, smoothed curves and simulated human trajectories explained below (see Figure 3.7). At each difficulty level, the robot is randomly spawned at positions between 1 to 2.5 meters away from the person with uniformly random orientations. The details of each difficulty level is elaborated below.

**1. Straight:** The person moves with an initial random linear velocity throughout the episode.

**2. Circles:** The person moves in a circle with a different radius each time. We create the circular motion by selecting a random initial linear velocity in the range  $[0.2, 0.6]$  m/s and a random angular velocity in the range  $[0.3, 0.8]$  rad/s.

**3. Smoothed curves:** The person moves in random curves generated by following linear velocities  $V_l^t$  and angular velocities  $V_a^t$  with initial values similar to those in the Circles difficulty:

$$V_l^t = V_l^{t-1} - (V_l^{t-1} - R_1^{t-1})/3.$$

$$V_a^t = V_a^{t-1} - (V_a^{t-1} - R_2^{t-1})/3.$$

where  $R_1, R_2$  are random numbers between  $[0, 1]$  and  $[-1, 1]$  respectively.

**4. Simulated trajectories:** We first arbitrarily “draw” trajectories by moving a robot using a joystick in Gazebo to cover the space while recording robot coordinate points in 10 different occupancy grid map. The total length of all trajectories is roughly 50 meters. To add variety to the data, we add the reverse of each trajectory to the library of trajectories as well. During training, the person starts at a random point and tracks the above trajectories using a proportional integral derivative (PID) controller.



Figure 3.7: Visualization of person motion model. From left to right: moving straight, in different circles, in smoothed curves and using annotated simulated path of a human.

### 3.2.3 Simulated Experiments

In this section, we present our experiments in simulation. We compare LBGP (our system) with two baselines: the state-of-the-art Hand Crafted Following Ahead (HC) method in [68] and an End-to-End learning Following Ahead (E2E) approach. The HC system exploits

EKF to predict the position of the user and then navigates to a point ahead of the predicted position using a trajectory planner. For sake of consistency, we use the same TEB motion planner as in HC. For E2E approach, we use the same D4PG implementation with curriculum learning, but instead of a navigational goal, the policy directly outputs the robot’s linear and angular velocities scaled to  $[-1, 1]$  m/s and  $[-2, 2]$  rad/s, respectively.

We conduct three experiments with different human trajectories. In each experiment, we report the mean person-robot angle  $\alpha$ , the mean robot-user distance  $D$  and the episode accumulated reward. The results of all the three experiments are included in Table 3.1. In all experiments, the robot has no prior knowledge of the planned trajectory of the human.

### 3.2.3.1 Straight

Human Trajectory	Approach	Distance mean $\pm$ std	Orientation mean $\pm$ std	Reward
Straight ahead	LBGP	$1.53 \pm 0.2$	$7.1 \pm 6.7$	28.31
	HC	$1.35 \pm 0.2$	$-1.1 \pm 1.6$	<b>33.44</b>
	E2E	$1.75 \pm 0.3$	$8.9 \pm 7.2$	27.50
Straight behind	LBGP	$1.59 \pm 0.2$	$64.3 \pm 62.6$	<b>10.50</b>
	HC	$1.10 \pm 0.2$	$86.1 \pm 66.6$	1.30
	E2E	$1.54 \pm 0.5$	$91.8 \pm 63.1$	-3.13
Turning ahead	LBGP	$1.90 \pm 0.3$	$-5.7 \pm 10.3$	<b>24.96</b>
	HC	$1.04 \pm 0.2$	$-12.6 \pm 11.6$	20.20
	E2E	$1.96 \pm 0.3$	$-7.5 \pm 4.6$	22.44
Turning behind	LBGP	$1.69 \pm 0.3$	$81.7 \pm 67.8$	<b>0.40</b>
	HC	$1.07 \pm 0.3$	$83.9 \pm 83.4$	-5.03
	E2E	$1.62 \pm 0.5$	$55.1 \pm 81.1$	0.17
Turning inside	LBGP	$1.72 \pm 0.3$	$1.6 \pm 23.6$	<b>23.04</b>
	HC	$0.98 \pm 0.3$	$3.6 \pm 24.1$	8.37
	E2E	$2.11 \pm 0.5$	$-1.5 \pm 18.7$	11.36
Turning outside	LBGP	$1.56 \pm 0.2$	$-33.1 \pm 19.8$	<b>13.97</b>
	HC	$1.07 \pm 0.2$	$-22.8 \pm 18.7$	13.68
	E2E	$1.32 \pm 0.2$	$56.3 \pm 72.1$	2.53
Trajectory one	LBGP	$1.37 \pm 0.3$	$10.4 \pm 15.4$	<b>27.17</b>
	HC	$1.09 \pm 0.3$	$-8.8 \pm 34.8$	0.91
	E2E	$1.65 \pm 0.2$	$22.9 \pm 42.4$	16.60
Trajectory two	LBGP	$1.54 \pm 0.2$	$-4.7 \pm 62.2$	<b>15.86</b>
	HC	$1.12 \pm 0.4$	$-57.0 \pm 45.8$	-5.83
	E2E	$1.62 \pm 0.2$	$-1.7 \pm 72.4$	10.99
Trajectory three	LBGP	$1.59 \pm 0.3$	$12.6 \pm 81.0$	<b>11.79</b>
	HC	$1.15 \pm 0.4$	$6.2 \pm 75.8$	-8.02
	E2E	$1.92 \pm 0.3$	$11.0 \pm 80.1$	2.15

Table 3.1: Comparison of our systems versus two baselines for all simulation trajectories.

The first experiment was conducted on straight human motion trajectory to compare the behaviour of the three methods. The human simply starts moving forward with a constant linear velocity of 0.6 m/s. We spawn the robot relative to person in two initial settings, **Ahead**: ( $D = 1.5\text{m}$ ,  $\alpha = 0^\circ$ ) and **Behind**: ( $D = 1.5\text{m}$ ,  $\alpha = 180^\circ$ ).

We compare our results with HC and E2E (Table 3.1). For the Ahead setting, HC achieves the highest return. HC can achieve a better results as the incorporated EKF relies on linearity of human motion and it can optimally follow the straight line. In LBGP training, we apply Gaussian noise thus the robot may slightly deviate to the sides. In the Behind setting, our approach achieves the highest reward as it has learned to keep a safe distance with the human by setting further navigational goals for TEB compared to HC.

### 3.2.3.2 Turning

We assess different approaches with turning trajectories. In this case, the person moves with a linear velocity of 0.3 m/s and angular velocity of 0.3 rad/s. To cover a large variety of initial conditions, we evaluate four positions of the robot relative to person, **Ahead**: ( $D = 1.5\text{m}$ ,  $\alpha = 180^\circ$ ), **Behind**: ( $D = 1.5\text{m}$ ,  $\alpha = 0^\circ$ ), **Ahead-inside-the-turn**: ( $D = 1.5\text{m}$ ,  $\alpha = 45^\circ$ ) or **Ahead-outside-the-turn**: ( $D = 1.5\text{m}$ ,  $\alpha = -45^\circ$ ). The result of this experiment shows that our LBGP achieves the highest return in all the settings (Table 3.1).

### 3.2.3.3 Simulated trajectories

We designed three simulated trajectories to further evaluate our system. Similar to the training phase, we employ PID controllers for the simulated human to follow totally unseen trajectories, and the learning agent attempts to stay in front of the simulated human. Figure 3.8 shows the robot’s trajectories corresponding to the three different human trajectories. In this experiment, we can see a more noticeable difference in performance of LBGP compared to both baselines (Table 3.1). Compared to E2E, our method likely performs better as the usage of a trajectory planner abstracts navigation to predicting a goal, while the E2E method needs to implicitly learn the dynamics of the system. Learning accurate dynamics can be challenging and may expose the E2E to over-fitting. Our method, LBGP, also outperforms the HC system, since LBGP predicts a goal based on a history of the human trajectory as opposed to using a linear human motion model as in HC.

### 3.2.3.4 Ablation Study

We performed an ablation study to evaluate the effectiveness of different modules and training procedures of our LBGP approach. We compare the performance of our approach to two variants of it: 1) without curriculum learning (*LBGP-no-curriculum*), and 2) without a trajectory planner (*E2E*, same as described in Section 3.2.3). As shown in Figure 3.9, both

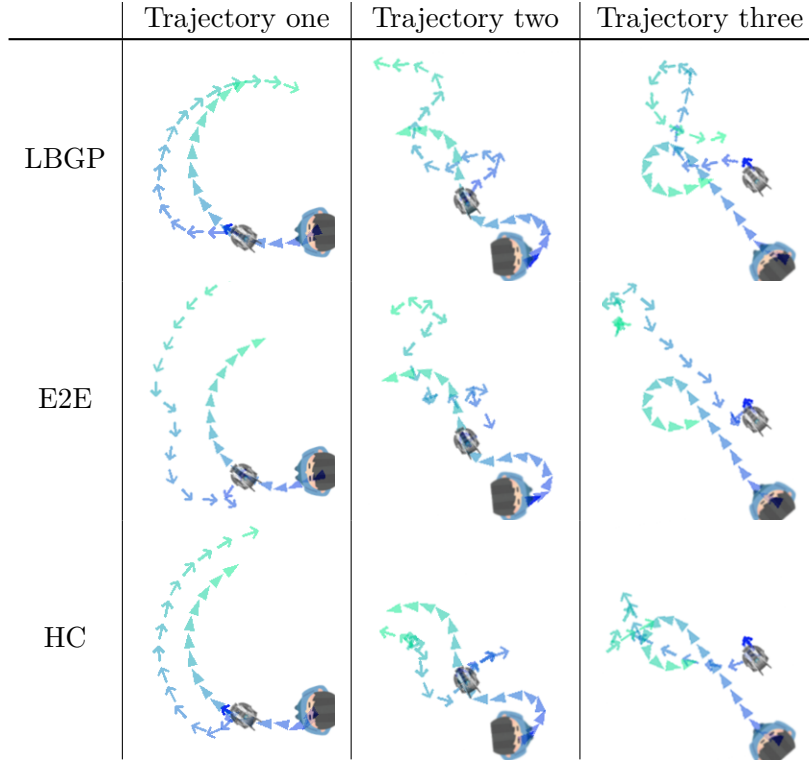


Figure 3.8: Visualize the trajectory of robot (arrows) and human (triangle) during the simulated trajectory experiment for our system (LBGP) and two baselines, HC and E2E.

*LBGP-no-curriculum* and *E2E* have slow learning curves and reach a lower discounted cumulative reward bound compared to LBGP, our proposed method.

### 3.2.4 Real World Experiments

We test LBGP on a TurtleBot 2 hardware testbed, and evaluate the transferability of the policy trained in simulation, using our approach, to the real world. We also compare our method’s sim2real ability with the two baselines, HC and E2E, defined in Section 3.2.3. We performed three experiments each with 4 different initial relative states. In short, our approach demonstrates successful zero-shot sim2real transfer of the policy.

To keep the experiments consistent between different approaches, all the initial states of human and robot along with the trajectory of human are marked on the ground with color tapes. In each experiment, we report the total discounted cumulative reward, the mean person-robot angle ( $\alpha$ ) and the mean person-robot distance ( $D$ ) as a measures of the follow-ahead quality. To make the accumulated reward a fair evaluation, we keep a constant number of time step for each setting. We use a motion capture system to record the robot and person’s states. For all the experiments we use the policies we trained in simulation with no changes. In each setting, we terminated the experiment as soon as the robot hits the person or gets more than three meters away from the person. As with the simulations,

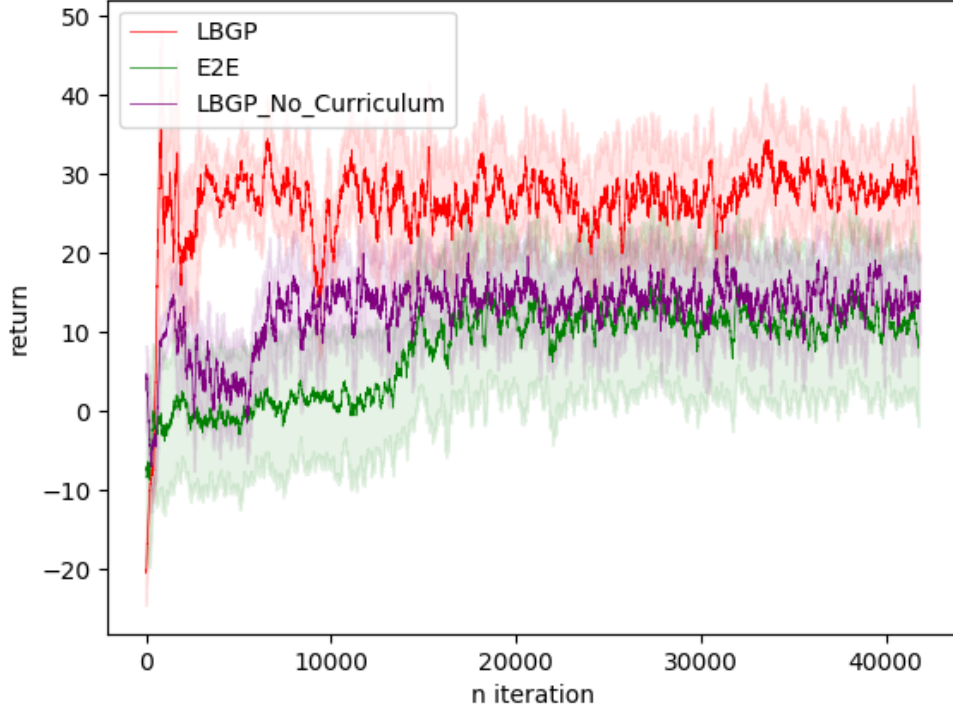


Figure 3.9: Discounted cumulative rewards during training averaged over five runs for LBGP with or without curriculum and E2E. The shaded area represents half a standard deviation.

in every real world experiment, the robot has no knowledge of the planned trajectory of the human.

### 3.2.4.1 *Straight* Trajectory

In this experiment, the initial positions of robot relative to human are as follows: **Ahead**: ( $D = 2\text{m}$ ,  $\alpha = 0^\circ$ ), **Ahead-right**: ( $D = 1.5\text{m}$ ,  $\alpha = 45^\circ$ ), **Ahead-left**: ( $D = 1.5\text{m}$ ,  $\alpha = -45^\circ$ ) and **Behind**: ( $D = 1.2\text{m}$ ,  $\alpha = 180^\circ$ ). In each setting, the person intends to navigate with a constant forward speed toward a goal located at 7 meters of its initial position. The four settings along with the result of the *Straight* experiment is reported in Table 3.2. In this experiment the EKF model of HC can correctly predict the human trajectory and it achieves the highest reward only for the Ahead setting. For all the other settings, our LBGP method achieves the highest performance. It is likely because the policy in LBGP is trained to keep the safety distance with the human. E2E failed to accomplish the following ahead task due to collision with person (Ahead and Ahead-right settings) or drifting away in reverse direction (Behind setting). Likely, E2E learns to navigate only with the specific simulated robot dynamics and unable to generalize to a new robot dynamics in the real world experiments.

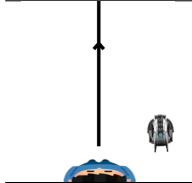
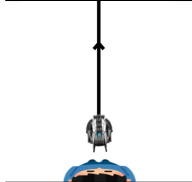
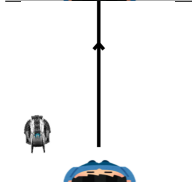
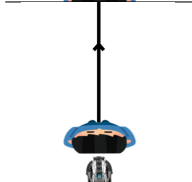
Human Trajectory	Approach	Distance mean $\pm$ std	Orientation mean $\pm$ std	Reward
	LBGP	$1.63 \pm 0.4$	$-8.4 \pm 26.6$	<b>27.42</b>
	HC	$1.24 \pm 0.6$	$-11.8 \pm 16.4$	26.31
	E2E	Failed	Failed	Failed
	LBGP	$1.24 \pm 0.3$	$2.1 \pm 14.5$	40.92
	HC	$1.14 \pm 0.2$	$0.5 \pm 3.0$	<b>61.08</b>
	E2E	$1.14 \pm 0.3$	$-0.1 \pm 10.0$	h41.59
	LBGP	$1.61 \pm 0.4$	$30.2 \pm 34.6$	<b>15.68</b>
	HC	$1.28 \pm 0.7$	$16.8 \pm 21.6$	14.24
	E2E	Failed	Failed	Failed
	LBGP	$1.85 \pm 0.3$	$34.0 \pm 95.2$	<b>-6.79</b>
	HC	$1.73 \pm 0.3$	$-105.7 \pm 51.0$	-15.69
	E2E	Failed	Failed	Failed

Table 3.2: Comparison of our systems versus two baselines for *Straight* trajectory.

### 3.2.4.2 *S shaped* Trajectory

In the second experiment, we evaluate our system for an *S* shaped trajectory. The initial relative position of robot is exactly similar to the *Straight* trajectory. The user deliberately follows a *S shape* path for all the settings. As shown in Table 3.3, LBGP achieves the highest return in all four settings. When the person travels along an *S shaped* trajectory, it is important to consider a history of the person to predict its future trajectory and a simple EKF as in HC cannot correctly predict the complexity of this motion. Figure 3.10 visualizes examples of the robot and human trajectories for Ahead-right and Behind settings. Similar to the *Straight* experiment, E2E failed three out of four settings by colliding with the user.

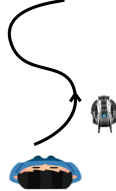
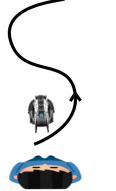


Human Trajectory	Approach	Distance mean $\pm$ std	Orientation mean $\pm$ std	Reward
	LBGP	$2.09 \pm 0.3$	$-4.9 \pm 26.7$	<b>2.30</b>
	HC	$1.14 \pm 0.5$	$-14.2 \pm 78.4$	-9.42
	E2E	Failed	Failed	Failed
	LBGP	$1.86 \pm 0.4$	$16.9 \pm 28.3$	<b>5.15</b>
	HC	$1.04 \pm 0.3$	$38.9 \pm 60.3$	-6.83
	E2E	$1.41 \pm 0.4$	$-35.4 \pm 53.0$	3.45
	LBGP	$1.81 \pm 0.6$	$35.8 \pm 33.1$	<b>11.74</b>
	HC	$1.60 \pm 0.8$	$65.3 \pm 50.0$	-9.29
	E2E	Failed	Failed	Failed
	LBGP	$1.30 \pm 0.2$	$38.2 \pm 70.9$	<b>7.84</b>
	HC	$1.69 \pm 0.4$	$12.2 \pm 151.6$	-18.64
	E2E	Failed	Failed	Failed

Table 3.3: Comparison of our systems versus two baselines for *S shape* trajectory.

### 3.2.4.3 *U-turn* Trajectory

Lastly, we evaluate the LBGP when the person perform a U-turn. The initial positions of robot relative to human are as follows: **Ahead**: ( $D = 1.7\text{m}$ ,  $\alpha = 0^\circ$ ), **Ahead-left** ( $D = 2\text{m}$ ,  $\alpha = -55^\circ$ ), **Ahead-far-left** ( $D = 3.6\text{m}$ ,  $\alpha = -75^\circ$ ) and **Behind** ( $D = 1.2\text{m}$ ,  $\alpha = 180^\circ$ ). Table 3.4 shows the four settings along with the result of the *U-turn* experiment, and LBGP consistently accumulates the highest reward. For a challenging U-turn trajectory, it is important for the robot to “notice” these specific walking patterns and react spontaneously. This cannot be done in HC method as HC anticipate future based on the heading of the person. Examples of robot and human trajectories for Ahead-left and Behind settings is visualized in Figure 3.10. For instance, in the Ahead-left setting, LBGP predict the turn early and avoid getting far away from the person. On the other hand, E2E has trouble transferring the policy to the real world.



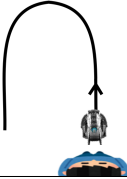
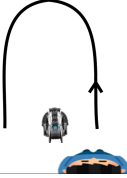
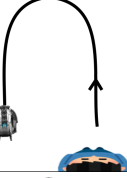

Human Trajectory	Approach	Distance mean $\pm$ std	Orientation mean $\pm$ std	Reward
	LBGP	$1.99 \pm 0.2$	$-16.8 \pm 18.6$	<b>14.91</b>
	HC	$1.01 \pm 0.5$	$-55.2 \pm 57.0$	-11.33
	E2E	Failed	Failed	Failed
	LBGP	$1.34 \pm 0.4$	$20.9 \pm 37.8$	<b>18.73</b>
	HC	$1.15 \pm 0.3$	$-28.7 \pm 98.1$	-8.09
	E2E	$1.75 \pm 0.4$	$43.3 \pm 26.7$	10.63
	LBGP	$1.91 \pm 0.7$	$36.2 \pm 36.3$	<b>16.52</b>
	HC	$1.31 \pm 0.9$	$-6.1 \pm 43.8$	-13.14
	E2E	Failed	Failed	Failed
	LBGP	$1.53 \pm 0.3$	$44.5 \pm 58.5$	<b>20.03</b>
	HC	$1.12 \pm 0.3$	$-1.7 \pm 120.3$	-11.83
	E2E	$1.82 \pm 0.3$	$67.7 \pm 45.4$	-7.43

Table 3.4: Comparison of our systems versus two baselines for *U-turn* trajectory.

### 3.2.5 Discussion

#### 3.2.5.1 Comparison to the Hand Crafted method

Our results show that our proposed hybrid approach for following ahead outperforms the HC method in both the simulation and real world. LBGP is able to create a complex model of the environment with a better abstraction of the human motion model as opposed to a linear EKF in the HC. Another advantage of RL is the large amount of training data that can be obtained in a simulated environment. This allows LBGP to better predict human trajectories (implicitly) compared to a hand-crafted method.

#### 3.2.5.2 Comparison to the End-to-End method

Although E2E achieves a comparable performance in simulation, it is unreliable in the real world. Using E2E, the robot collided with the user in all three real world experiments. We also saw the robot shaking a lot when we used E2E. After investigating, we identified the dynamics of the real world robot differ from the simulated one, which prevents E2E from extending the learned behaviour to the real world. In contrast, LBGP overcomes this model mismatch by abstracting away the dynamics using the TEB trajectory planner. This

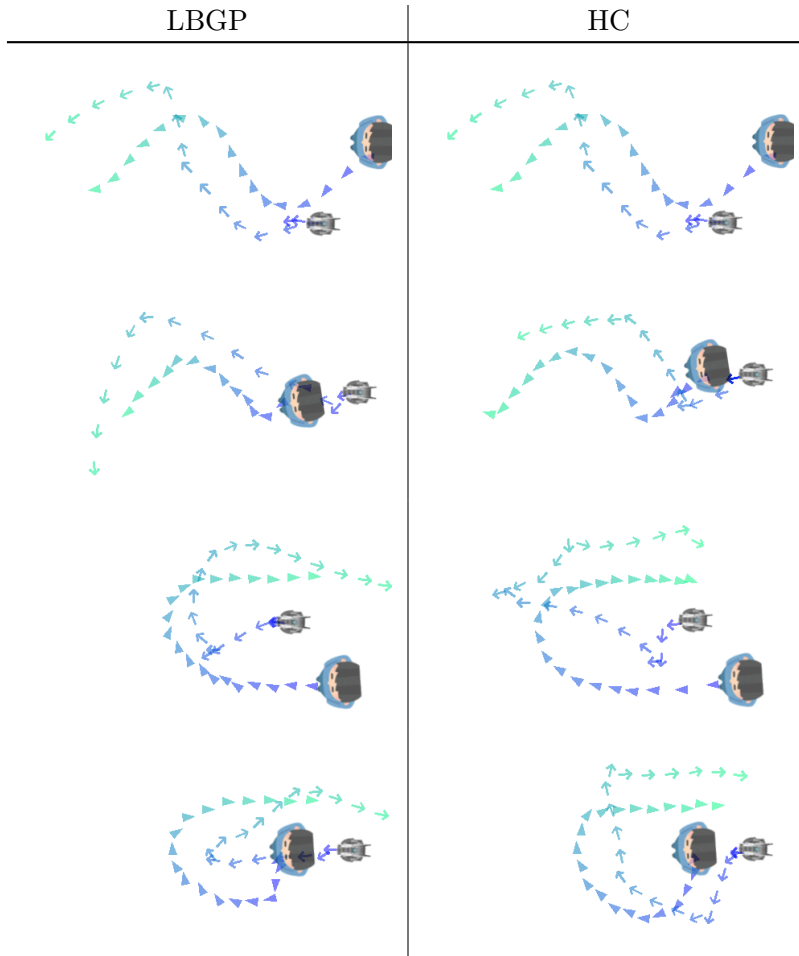


Figure 3.10: real world Examples: the robot (in arrows) and user (in triangles) trajectories is depicted. Row 1 and 2, *S shape* experiment in *ahead-right* and *behind* settings. Row 3 and 4: *U-turn* experiment in *ahead-left* and *behind* settings.

planner also helps our system to avoid any collision with the person while staying at a safe distance.

### 3.2.6 Conclusion

We propose LBGP, a follow-ahead method that uses both reinforcement learning and point based navigation. We address the limitations of classical methods and end-to-end approaches by combining Deep RL and a classical motion planner. Our implementation outperforms previous work in an obstacle-free environment [68]. To train our deep RL model, we used curriculum learning by gradually increasing the difficulty of the person motion model to learn a robust policy for front following. Our results show that using a planner improves the generalizability and safety of the trained policy compared to an End-to-End method and allows us to perform zero-shot sim2real transfer successfully.

### 3.3 Human Following using STPOTR

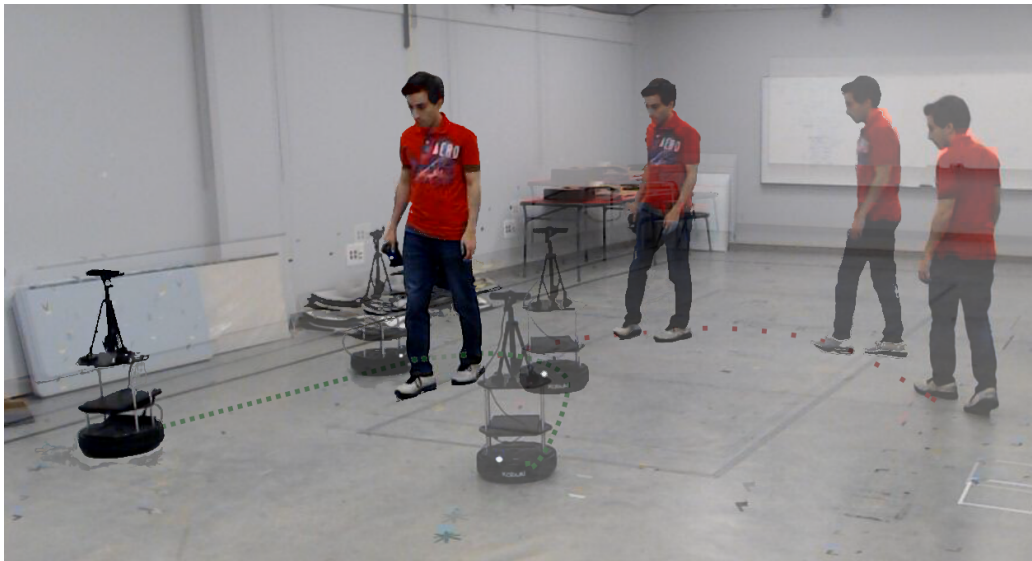


Figure 3.11: Robot follow-ahead via human motion prediction in the *U-Shaped* scenario using the STPOTR model. Opacity increases with time.

In this section, we demonstrate the real world application of the STPOTR [56]. We use STPOTR trained model to perform the challenging task of robot follow-ahead and compare our method’s performance with a hand-crafted (HC) [70] and our RL-based (LBGP) [71] methods.

The primary difference between STPOTR and LBGP lies in their strategy for predicting human motion during human following. LBGP employs a one-shot approach that predicts both robot goals and human motion, whereas STPOTR uses a two-shot approach that first predicts human motion and then calculates the robot’s goal based on that prediction. Decoupling the robot’s decision-making and motion prediction can improve the approach’s generalizability and reduce the need for retraining when switching between different types of human following. For example, LBGP can only be used for following ahead since we only train the RL agent in this scenario, and to apply it to other scenarios, retraining is required. To demonstrate this benefit of STPOTR, we conducted a real-world experiment involving various human following scenarios, including sit-to-stand and follow-beside.

We test the same scenarios in the two baselines that are follow-ahead on a *Straight line* and *S-Shaped* and *U-Shaped* curves. Similarly, the robot starting points are on four sides of the user’s initial location (left, right, in front, and behind). In order to evaluate the follow-ahead task, a reward function similar to [71] based on the relative position of the robot and the human is used.

### 3.3.1 Robot Follow-Ahead via Human Motion Predictions


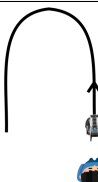

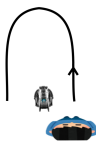

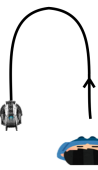






In order to use the trained model for the robot follow-ahead task in the real world, it was retrained with Gaussian noise added to the input sequence to improve the robustness to noisy inputs. A ZED2 camera was used as a 3rd person viewer for human pose estimation and a turtlebot2 [81] robot was used as the testing platform. We used a 3rd person camera to abstract away hardware complications such as limited field of view. Our focus is on demonstrating that the trained model for human motion prediction is useful for the follow-ahead task. At each moment, the ZED2 camera captured an image and estimated the current human body 3D joints' positions. Then we sent the last 0.5-second frames (5 frames) to the STPOTR model and predicted the user's motion in the next 2 seconds (20 frames). We calculated the future human heading using the line created by the left and right hip joints positions on the 20th frame, and chose the point 1.5 meters in front, oriented in the same direction as the robot goal pose (position and orientation). Then we used the Time Elastic Band (TEB) trajectory planner [93] to move the robot toward the goal pose. This planner also helps our follow-ahead system to stay a safe distance from the person at all times to avoid any collisions with the person while staying at a safe distance.

### 3.3.2 Real World Experimental Results

Table 3.5 compares the achieved reward by our robot follow-ahead method with respect to our baselines [70, 71]. The reported rewards are the average reward values of two tests on two different users. We were able to achieve a much higher reward in the *S-Shaped* scenarios which is a complicated motion and comparable results in other ones. Our method only performed poorly when the robot was placed on the human's left side during *U-Shaped* motion which can be due to the far distance between the human and robot during the initial periods of the motion. Fig. 3.13 shows a few of the follow-ahead motions in different scenarios.

Crucially, note that the LBGP [71] used a motion capture system for localizing the human and robot, which greatly simplifies the human-following problem, whereas we present a more realistic method that uses the much noisier 3D human pose estimation of the ZED2 camera.

Table 3.5: Robot follow-ahead comparative results for three tested scenarios.

Human Trajectory	Method	Reward	Human Trajectory	Method	Reward
	Ours	25.94		Ours	13.11
	LBGP	<b>27.42</b>		LBGP	<b>14.91</b>
	HC	26.31		HC	-11.33
	Ours	25.31		Ours	8.89
	LBGP	40.92		LBGP	<b>18.73</b>
	HC	<b>61.08</b>		HC	-8.09
	Ours	7.62		Ours	-10.92
	LBGP	<b>15.68</b>		LBGP	<b>16.52</b>
	HC	14.24		HC	-13.14
	Ours	-5.14		Ours	7.37
	LBGP	-6.79		LBGP	<b>20.03</b>
	HC	-15.69		HC	-11.83
	Ours	<b>22.02</b>		Ours	<b>24.15</b>
	LBGP	2.30		LBGP	5.15
	HC	-9.42		HC	-6.83
	Ours	<b>14.33</b>		Ours	<b>8.28</b>
	LBGP	11.74		LBGP	7.84
	HC	-9.29		HC	-17.64

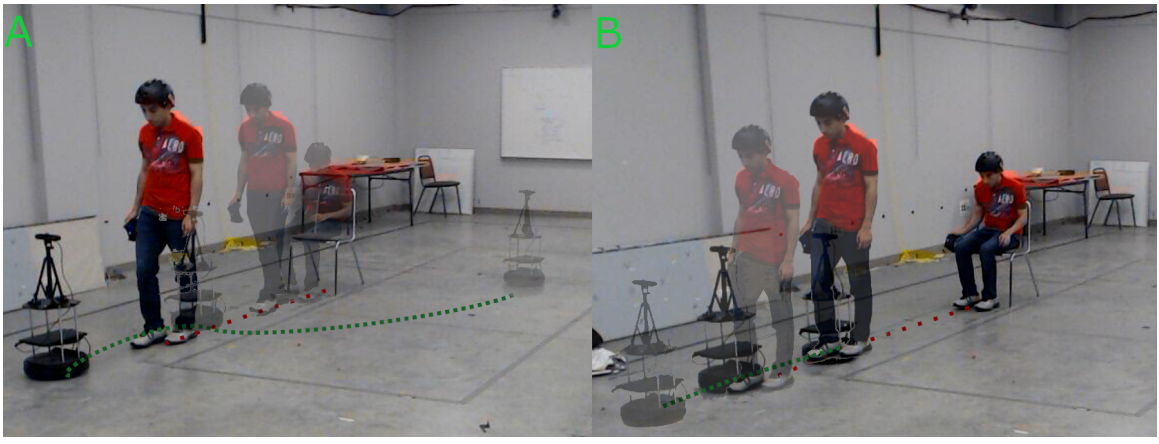


Figure 3.12: *Sit-to-Stand* (A) and *Stand-to-Sit* (B) use cases of STPOTR. Opacity increases with time.

Perhaps most importantly, we were also able to account for much more detail in the human motion for the follow-ahead task. This enables our system to easily generalize to many different scenarios involving different human motions such as *Sit-to-Stand* and *Stand-to-Sit* (illustrated in Fig. 3.12), as well as to different variations of human following such as *follow-beside* or keeping a *variable distance* with the human depending on the human walking speed or surrounding environment. These are very difficult, if not impossible, scenarios and tasks for our baselines. For example, the RL-based LBGP [71] method would require retraining of the policy for every variation of the human-following task. LBGP also does not account for the human body pose, and the HC [70] method, in addition, does not consider the human heading. The application of our algorithm in these different scenarios and task variations can be found at [https://www.youtube.com/playlist?list=PLuLzEWWNu1\\_p1bjUHhWUHRMFOLmLpUCpM](https://www.youtube.com/playlist?list=PLuLzEWWNu1_p1bjUHhWUHRMFOLmLpUCpM)

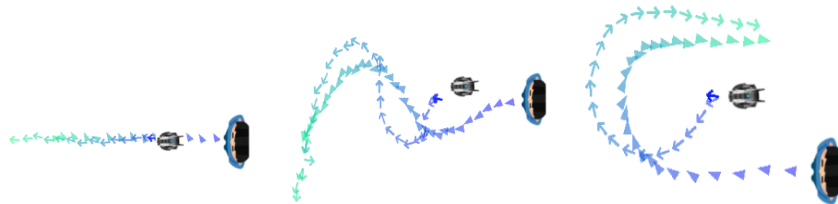


Figure 3.13: Three samples of the robot follow-ahead tasks for U-Shaped, S-Shaped and straight line scenarios. The triangle and arrows show the human and robot motions, respectively.

### 3.3.3 Conclusion

Our work showcased the efficacy of STPOTR in predicting human motion in a real world robotics task involving the robot follow-ahead, where we obtained results that were either better or comparable to those achieved by previous methods.

One significant advantage of our method is its potential for generalizability. By predicting human motion, our method can be adapted for various types of human following, including following ahead, or side-by-side, making it more versatile than current state-of-the-art methods. This opens up new possibilities for applications in various areas, such as service robots, autonomous vehicles, and augmented reality.

Overall, our results demonstrate the potential of STPOTR to enhance human-robot interaction by accurately predicting human motion and enabling robots to follow humans more effectively in various scenarios.

### 3.4 Conclusion

In this Chapter, we introduce two approaches to the human following, one that utilizes Reinforcement Learning (RL) to implicitly learn the human’s goal and the other that employs a human motion prediction model to explicitly forecast the human’s future movements.

Our first approach, LBGP, combines reinforcement learning and point-based navigation. We addressed the shortcomings of traditional methods and end-to-end approaches by integrating Deep RL with a classical robotics planner. Our implementation outperforms prior work in an obstacle-free environment and uses curriculum learning to train the deep RL model. Our results indicate that using a classical robotics planner enhances the generalizability and safety of the policy compared to end-to-end methods, and enables us to perform zero-shot sim2real transfer effectively.

In our second approach, we utilize our implementation of STPOTR to predict the user’s future 3D motion, which results in improvement over existing follow-ahead methods. Additionally, we demonstrate the applicability of this method to various human following categories.

## Chapter 4

# Future work

This thesis has presented promising approaches for human motion prediction and human-following for a companion robot. However, there are still several areas for future work in this field.

Firstly, we can investigate the integration of additional sensory information, such as visual or auditory cues, to enhance the accuracy and robustness of the developed human motion prediction methods. Additionally, we can explore the use of more advanced machine learning techniques, such as diffusion models, to model the complex and dynamic nature of human motion.

Secondly, we can extend the developed human-following methods to different scenarios and environments, such as crowded areas or dynamic environments, and evaluate their performance in these settings. Furthermore, we can investigate the development of human-robot interaction strategies that enable natural and intuitive communication between the companion robot and humans, which can enhance the overall user experience and acceptance of the technology.

Finally, we can investigate the development of multi-agent systems that enable multiple robots to collaborate and coordinate with each other for human-following tasks, which can have significant implications for future robotics applications.



# Bibliography

- [1] Lina Achaji, Thierno Barry, Thibault Fouqueray, Julien Moreau, Francois Aioun, and Francois Charpillet. PreTR: Spatio-Temporal Non-Autoregressive Trajectory Prediction Transformer. *arXiv preprint arXiv:2203.09293*, 2022.
- [2] Pedram Agand, Mahdi Taherahmadi, Angelica Lim, and Mo Chen. Human navigational intent inference with probabilistic and optimal approaches. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8562–8568. IEEE, 2022.
- [3] 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. IEEE, 2021.
- [4] Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Lars Petersson, and Stephen Gould. A stochastic conditioning scheme for diverse human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5223–5232, 2020.
- [5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.
- [6] Somil Bansal, Varun Tolani, Saurabh Gupta, Jitendra Malik, and Claire Tomlin. Combining optimal control and learning for visual navigation in novel environments. In *Conference on Robot Learning*, pages 420–429. PMLR, 2020.
- [7] German Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction. *arXiv preprint arXiv:2211.14304*, 2022.
- [8] Barsoum, Emad and Kender, John and Liu, Zicheng. HP-GAN: Probabilistic 3D Human Motion Prediction via GAN. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1499–149909, 2018.
- [9] Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.
- [10] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris

- Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.
- [11] Changan Chen, Sha Hu, Payam Nikdel, Greg Mori, and Manolis Savva. Relational graph learning for crowd navigation. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [12] Changan Chen, Sha Hu, Payam Nikdel, Greg Mori, and Manolis Savva. Relational graph learning for crowd navigation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10007–10013. IEEE, 2020.
- [13] KyungHyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *CoRR*, abs/1409.1259, 2014.
- [14] Carlos A Cifuentes, Anselmo Frizera, Ricardo Carelli, and Teodiano Bastos. Human–robot interaction based on wearable IMU sensor and laser range finder. *Robotics and Autonomous Systems*, 62(10):1425–1439, 2014.
- [15] Qiongjie Cui and Huaijiang Sun. Towards accurate 3d human motion prediction from incomplete observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4801–4810, 2021.
- [16] Qiongjie Cui, Huaijiang Sun, and Fei Yang. Learning dynamic relationships for 3d human motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6519–6527, 2020.
- [17] B. S. B. Dewantara and J. Miura. Generation of a socially aware behavior of a guide robot using reinforcement learning. In *2016 International Electronics Symposium (IES)*, pages 105–110, 2016.
- [18] Gonzalo Ferrer, Anais Garrell, and Alberto Sanfeliu. Robot companion: A social-force based approach with human awareness-navigation in crowded environments. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1688–1694. IEEE, 2013.
- [19] Gonzalo Ferrer and Alberto Sanfeliu. Bayesian human motion intentionality prediction in urban environments. *Pattern Recognition Letters*, 44:134–140, 2014.
- [20] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International conference on Computer Vision*, pages 4346–4354, 2015.
- [21] Jiajun Fu, Fuxing Yang, and Jianqin Yin. Learning Constrained Dynamic Correlations in Spatiotemporal Graphs for Motion Prediction. *arXiv preprint arXiv:2204.01297*, 2022.
- [22] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pages 458–466, 2017.

- [23] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer Networks for Trajectory Forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10335–10342. IEEE, 2021.
- [24] Alex Goldhoorn, Anais Garrell, R. Alquézar, and A. Sanfeliu. Continuous real time pomcp to find-and-follow people by a humanoid service robot. *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 741–747, 2014.
- [25] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial Geometry-Aware Human Motion Prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786–803, 2018.
- [26] Mahir Gulzar, Yar Muhammad, and Naveed Muhammad. A survey on motion prediction of pedestrians and vehicles for autonomous driving. *IEEE Access*, 2021.
- [27] Xiao Guo and Jongmoo Choi. Human motion prediction via learning local structure representations and temporal dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2580–2587, 2019.
- [28] Dirk Helbing and Pé ter Molnár. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286, may 1995.
- [29] D. M. Ho, J. S. Hu, and J. J. Wang. Behavior control of the mobile robot for accompanying in front of a human. In *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME Int. Conf.*, pages 377–382. IEEE, July 2012.
- [30] 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.
- [31] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.
- [32] J. Hu, Jyun-Ji Wang, and D. M. Ho. Design of sensing system and anticipative behavior for human following of mobile robots. *IEEE Transactions on Industrial Electronics*, 61:1916–1927, 2014.
- [33] Sungsik Huh, D. Shim, and J. Kim. Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of uavs. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3158–3163, 2013.
- [34] Tetsushi Ikeda, Yoshihiro Chigodo, Daniel Rea, Francesco Zanlungo, Masahiro Shiomi, and Takayuki Kanda. Modeling and prediction of pedestrian behavior based on the sub-goal concept. *Robotics*, 10:137–144, 2013.
- [35] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013.

- [36] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.
- [37] M. Islam, Michael Fulton, and Junaed Sattar. Toward a generic diver-following algorithm: Balancing robustness and efficiency in deep visual detection. *IEEE Robotics and Automation Letters*, 4:113–120, 2019.
- [38] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5308–5317, 2016.
- [39] E. J. Jung, B. J. Yi, and S. Yuta. Control algorithms for a mobile robot tracking a human in front. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ Int. Conf.*, pages 2411–2416. IEEE, Oct 2012.
- [40] Deneth Karunarathne, Yoichi Morales, Takayuki Kanda, and Hiroshi Ishiguro. Model of side-by-side walking without the robot knowing the goal. *International Journal of Social Robotics*, 10(4):401–420, 2018.
- [41] Yoshinori Kobayashi, Yuki Kinpara, Erii Takano, Yoshinori Kuno, Keiichi Yamazaki, and Akiko Yamazaki. Robotic wheelchair moving with caregiver collaboratively. In De-Shuang Huang, Yong Gan, Phalguni Gupta, and M. Michael Gromiha, editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 523–532, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [42] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020.
- [43] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, 2004.
- [44] Jonáš Kulháněk, Erik Derner, Tim de Bruin, and Robert Babuška. Vision-based navigation using deep reinforcement learning. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–8. IEEE, 2019.
- [45] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. Bihmp-gan: Bidirectional 3d human motion prediction gan. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8553–8560, 2019.
- [46] Andreas M Lehrmann, Peter V Gehler, and Sebastian Nowozin. Efficient nonlinear markov models for human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1314–1321, 2014.
- [47] Angus Leigh, Joelle Pineau, Nicolas Olmedo, and Hong Zhang. Person Tracking and Following with 2D Laser Scanners. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 726–733. IEEE, 2015.

- [48] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Multiscale Spatio-Temporal Graph Neural Networks for 3D Skeleton-Based Motion Prediction. *IEEE Transactions on Image Processing*, 30:7760–7775, 2021.
- [49] Yanran Li, Zhao Wang, Xiaosong Yang, Meili Wang, Sebastian Iulian Poiana, Ehtzaz Chaudhry, and Jianjun Zhang. Efficient convolutional hierarchical autoencoder for human motion prediction. *The Visual Computer*, 35(6):1143–1156, 2019.
- [50] Xiao Lin and Mohamed R Amer. Human motion modeling using DVGANS. *arXiv preprint arXiv:1804.10652*, 2018.
- [51] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)*, 39(4):40–1, 2020.
- [52] Zhenguang Liu, Pengxiang Su, Shuang Wu, Xuanjing Shen, Haipeng Chen, Yanbin Hao, and Meng Wang. Motion prediction using trajectory cues. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13279–13288, 2021.
- [53] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv preprint arXiv:1711.05101*, 2019.
- [54] Jacobo Jimenez Lugo and A. Zell. Framework for autonomous on-board navigation with the ar.drone. *Journal of Intelligent & Robotic Systems*, 73:401–412, 2013.
- [55] Kedi Lyu, Haipeng Chen, Zhenguang Liu, Beiqi Zhang, and Ruili Wang. 3d human motion prediction: A survey. *Neurocomputing*, 489:345–365, 2022.
- [56] 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. In *IEEE International Conference on Robotics and Automation (ICRA), 2023 IEEE Int. Conf.*, 2023.
- [57] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *European Conference on Computer Vision*, pages 474–489. Springer, 2020.
- [58] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning Trajectory Dependencies for Human Motion Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9489–9497, 2019.
- [59] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017.
- [60] Angel Martínez-González, Michael Villamizar, and Jean-Marc Odobez. Pose transformers (POTR): Human motion prediction with non-autoregressive transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2276–2284, 2021.
- [61] Yoichi Morales, Takayuki Kanda, and Norihiro Hagita. Walking together: Side-by-side walking model for an interacting robot. *Journal of Human-Robot Interaction*, 3(2):50–73, 2014.

- [62] Luis Yoichi Morales Saiki, Satoru Satake, Rajibul Huq, Dylan Glas, Takayuki Kanda, and Norihiro Hagita. How do people walk side-by-side? using a computational model of human behavior for a social robot. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 301–308, 2012.
- [63] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 5(4):e10047, 2010.
- [64] G. P. Moustris and C. S. Tzafestas. Assistive front-following control of an intelligent robotic rollator based on a modified dynamic window planner. In *Biomedical Robotics and Biomechanics (BioRob), 2016 6th IEEE Int. Conf.*, pages 588–593. IEEE, June 2016.
- [65] G. P. Moustris and C. S. Tzafestas. Intention-based front-following control for an intelligent robotic rollator in indoor environments. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, Dec 2016.
- [66] Ryo Murakami, Luis Yoichi Morales Saiki, Satoru Satake, Takayuki Kanda, and Hiroshi Ishiguro. Destination unknown: walking side-by-side without knowing the goal. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 471–478, 2014.
- [67] Sanmit Narvekar and Peter Stone. Learning curriculum policies for reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 25–33. International Foundation for Autonomous Agents and Multiagent Systems, 2019.
- [68] P. Nikdel, Rakesh Shrestha, and R. Vaughan. The hands-free push-cart: Autonomous following in front by predicting user trajectory around obstacles. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2018.
- [69] Payam Nikdel, Mohammad Mahdavian, and Mo Chen. DMMGAN: Diverse Multi Motion Prediction of 3D Human Joints using Attention-Based Generative Adversarial Network. In *IEEE International Conference on Robotics and Automation (ICRA), 2023 IEEE Int. Conf.*, 2023.
- [70] 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. IEEE, 2018.
- [71] 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. IEEE, 2021.
- [72] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.
- [73] John M. Pierre. End-to-end deep learning for robotic following. In *ICMSCE 2018*, 2018.

- [74] Aleksey Postnikov, Aleksander Gamayunov, and Gonzalo Ferrer. Transformer based Trajectory Prediction. *arXiv preprint arXiv:2112.04350*, 2021.
- [75] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *IEEE International Conference on Robotics and Automation (ICRA) workshop on open source software*, volume 3, page 5. Kobe, 2009.
- [76] Khandakar M Rashid and Amir H Behzadan. Enhancing motion trajectory prediction for site safety by incorporating attitude toward risk. *Computing in Civil Engineering 2017*, pages 425–433, 2017.
- [77] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. *arXiv preprint arXiv:1612.08242*, 2016.
- [78] Ely Repiso, Gonzalo Ferrer, and Alberto Sanfeliu. On-line adaptive side-by-side human robot companion in dynamic urban environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 872–877, 2017.
- [79] Leonid Sigal, Michael Isard, Horst Haussecker, and Michael J Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 98(1):15–48, 2012.
- [80] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- [81] Diksha Singh, Esha Trivedi, Yukti Sharma, and Vandana Niranjana. TurtleBot: Design and Hardware Component Selection. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 805–809. IEEE, 2018.
- [82] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019.
- [83] Pengxiang Su, Zhenguang Liu, Shuang Wu, Lei Zhu, Yifang Yin, and Xuanjing Shen. Motion prediction via joint dependency modeling in phase space. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 713–721, 2021.
- [84] Yue Sun, Lei Sun, and Jingtai Liu. Human comfort following behavior for service robots. In *Robotics and Biomimetics (ROBIO), 2016 IEEE Int. Conf.*, pages 649–654. IEEE, 2016.
- [85] Junya Tominaga, Kensaku Kawauchi, and Jun Rekimoto. Around me: a system with an escort robot providing a sports player’s self-images. In *Proceedings of the 5th Augmented Human Int. Conf.*, page 43. ACM, 2014.
- [86] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.

- [87] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Proceedings of the IEEE International conference on Computer Vision*, pages 3332–3341, 2017.
- [88] M. Wang, Daobilige Su, Lei Shi, Yong Liu, and J. V. Miró. Real-time 3d human tracking for mobile robots with multisensors. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5081–5087, 2017.
- [89] X. Wang, L. Zhang, Duo Wang, and X. Hu. Person detection, tracking and following using stereo camera. In *International Conference on Graphic and Image Processing*, 2018.
- [90] Yachuan Wang, Xuan Wang, Peilin Jiang, and Fei Wang. Rnn -based human motion prediction via differential sequence representation. In *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 138–143, 2019.
- [91] Ronald J. Williams and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280, 1989.
- [92] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert. Autonomous navigation for BigDog. In *IEEE International Conference on Robotics and Automation, 2010*, pages 4736–4741. IEEE, May 2010.
- [93] Jiafeng Wu, Xianghua Ma, Tongrui Peng, and Haojie Wang. An Improved Timed Elastic Band (TEB) Algorithm of Autonomous Ground Vehicle (AGV) in Complex Environment . *Sensors*, 21(24):8312, 2021.
- [94] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On Layer Normalization in the Transformer Architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.
- [95] Xinchun Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *Proceedings of the European conference on Computer Vision (ECCV)*, pages 265–281, 2018.
- [96] Ningshi Yao, Emily Anaya, Qiuyang Tao, Sungjin Cho, Hongrui Zheng, and Fumin Zhang. Monocular vision-based human following on miniature robotic blimp. In *IEEE International Conference on Robotics and Automation (ICRA), 2017 IEEE Int. Conf.*, pages 3244–3249. IEEE, 2017.
- [97] Ye Yuan and Kris Kitani. DLow: Diversifying Latent Flows for Diverse Human Motion Prediction. In *European Conference on Computer Vision*, pages 346–364. Springer, 2020.
- [98] Ye Yuan and Kris M. Kitani. Diverse Trajectory Forecasting with Determinantal Point Processes. In *ICLR*, 2020.



- [99] S. M. Zadeh, A. Yazdani, K. Sammut, and D. Powers. Online path planning for auv rendezvous in dynamic cluttered undersea environment using evolutionary algorithms. *Appl. Soft Comput.*, 70:929–945, 2018.
- [100] Zhitian Zhang, Jimin Rhim, Angelica Lim, and Mo Chen. A Multimodal and Hybrid Framework for Human Navigational Intent Inference. In *International Conference on Intelligent Robots and Systems*, 2021.