Physics-based Deep Learning for Understanding Crowd Behaviors



Jiangbei Yue University of Leeds School of computing

Submitted in accordance with the requirements for the degree of $Doctor \ of \ Philosophy$

September, 2024

Intellectual Property Statement

The candidate confirms that the work submitted is his own. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

This thesis contains material from two published papers and one paper under review in the following peer-reviewed conferences in which I am the leading author.

- Chapter 3 is published as Jiangbei Yue, Dinesh Manocha and He Wang, "Human trajectory prediction via neural social physics", European Conference on Computer Vision (ECCV), 2022.
- Chapter 4 is under review as Jiangbei Yue, Dinesh Manocha and He Wang, "Uncertainty-aware Human Trajectory Prediction", under review.
- Chapter 5 is published as Jiangbei Yue, Baiyi Li, Julien Pettré, Armin Seyfried and He Wang, "Human Motion Prediction under Unexpected Perturbation", Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

The above publications are primarily the work of the candidate.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement. © 2025 The University of Leeds and Jiangbei Yue.

Acknowledgements

First, I would like to express my gratitude to my supervisor Dr. He Wang. His conscientious guidance in research and life helps me complete my doctoral programme successfully. I have benefited significantly from his detailed research training, rigorous research attitude, timely encouragement, *etc.* It has been a true privilege to be one of his students. I sincerely hope we have the opportunity to collaborate again in the future.

I would also give my thanks to Dr. David Hogg who often cared about my study and research during my doctoral phase. I am very grateful for his help in organizing my thesis examination.

I want to thank my co-authors, Dr. Dinesh Manocha, Dr. Julien Pettré, Dr. Armin Seyfried, and Baiyi Li for their contributions to my research. I also wish to work with you again in the future.

I would like to thank my friends and colleagues, Feixiang He, Zheyan Zhang, Baiyi Li, Dody Dharma, Deshan Gong, Jialin Zhu, Sara Jones, Shaun Buckley, Alhanouf Alolyan, Mou Li, Nanzhu Chen, Miranda Horne, and Xiaoyuan Yang, for their invaluable support in my PhD life, both professionally and personally.

In addition, I would also like to extend my heartfelt thanks to my family for their unconditional love and support. Qinyi Xie, my wife, deserves my special thanks. Your understanding, patience, and constant encouragement have been my anchor through the challenges and triumphs of my PhD.

Finally, I would like to acknowledge the financial support provided by the EU project CrowdDNA, the University of Leeds, and my parents.

Abstract

Understanding crowd behaviors is crucial in many vital areas *e.g.* public safety, urban planning, autonomous vehicles, *etc.* Although numerous excellent models have been proposed for the study of crowd behaviors, challenges persist due to the complexity of human behaviors. This thesis focuses on two primary challenges: crowd dynamics modeling in low-density crowds and physical interaction modeling in high-density crowds. To this end, we propose novel methods for prediction and uncertainty analysis of crowd dynamics in low-density scenarios while we introduce and solve a new research question about full-body motion to model physical interactions.

This thesis first models crowd dynamics to predict crowd movements in low-density crowds, which is also known as human trajectory prediction. Existing methods are typically divided into model-based and model-free methods. We design a new framework *Neural Social Physics* incorporating the advantages of both methodologies based on neural differential equation models. Then we propose a novel method under the framework by combining the social force model with neural networks. Our method trains neural networks to estimate the parameters of the social force model instead of handpicking or fixing them. A deep generative model is employed to capture the stochasticity of human trajectories. Through exhaustive evaluation, our method outperforms existing methods in prediction accuracy by up to 70%. In addition, our method provides plausible explanations for pedestrian behaviors and shows strong generalizability.

This thesis further explores uncertainty modeling in human trajectory prediction to capture stochastic future trajectories. The uncertainty of human trajectories consists of the data and model uncertainty. However, existing methods either only consider the data uncertainty or mix the two uncertainties, which is coarse-grained. To overcome the challenge, we propose a new Bayesian stochastic social force model, which captures fine-grained uncertainty through a decoupling strategy. Specifically, our method captures the data and model uncertainty by using a new Bayesian neural stochastic differential equation model and a deep generative model, respectively. We designed the uncertainty-aware training scheme allowing the two models to catch corresponding uncertainties well. Extensive experiments demonstrate that our method has strong explainability and improves the state-of-the-art prediction accuracy by as much as 60.17%.

Lastly, this thesis proposes a new research question that predicts 3D full-body motions under unexpected physical perturbation on both individual and group levels to study physical interactions between individuals. However, incorporating physical interactions in motion prediction brings new challenges e.g. complex interactions and data scarcity. To this end, we propose a latent differentiable physics model based on differentiable physics and neural networks. Our model introduces a latent physics space to learn body physics. Motions in the latent physics space are estimated first and converted then back into 3D full-body motions through a deep generative model. Considering that there is no similar research, we carefully choose 11 baselines from relevant domains and adapt them to the new task. Extensive evaluation and comparison demonstrate that our method outperforms other baselines in prediction accuracy by up to 70% and has outstanding data efficiency, strong generalizability, and good explainability.

1	Introduction				
	1.1	Backg	round		
	1.2	Resear	rch Questions		
	1.3	Our M	fethods		
	1.4	Organ	ization of the Thesis		
2	\mathbf{Rel}	ated V	Vork 8		
	2.1	Crowd	Dynamics Modeling in Low-density Crowds		
		2.1.1	Human Trajectory Prediction		
		2.1.2	Uncertainty in Human Trajectory Prediction		
		2.1.3	Crowd Simulation		
	2.2	Physic	cal Interactions in High-density Crowds		
		2.2.1	Traditional Research on Physical Interactions 15		
		2.2.2	Human Motion Prediction		
	2.3	Neura	l Differential Equations and Differentiable Physics		
3	Hu	man T	rajectory via Neural Social Physics 20		
	3.1	Introd	uction		
	3.2	Relate	ed Work		
		3.2.1	Trajectory Analysis and Prediction		
		3.2.2	Crowd Simulation		
		3.2.3	Deep Learning and Differential Equations		
	3.3	Metho	dology		
		3.3.1	Neural Social Physics (NSP)		
		3.3.2	NSP-SFM		

		3.3.3	NSP vs. Deep Neural Networks	31
	3.4	Exper	iments	31
		3.4.1	Datasets	31
		3.4.2	Trajectory Prediction	32
		3.4.3	Generalization to Unseen Scenarios	34
		3.4.4	Interpretability of Prediction	36
		3.4.5	Ablation Study	37
	3.5	Conclu	usions, Limitations, and Future Work	38
4	Unc	certain	ty-aware Human Trajectory Prediction	3 9
	4.1	Introd	uction	40
	4.2	Relate	ed Work	42
	4.3	Metho	odology	43
		4.3.1	Motivation	43
		4.3.2	Bayesian Stochastic Social Force Model	44
		4.3.3	Training Scheme and Loss Function	48
	4.4	Exper	iments	49
		4.4.1	Datasets	49
		4.4.2	Fine-grained Uncertainty and Explainability	50
		4.4.3	Trajectory Forecasting	52
		4.4.4	Generalization	53
		4.4.5	Data Efficiency	55
		4.4.6	Ablation Study	56
	4.5	Conclu	usion	57
5	Hur	nan M	lotion Prediction under Unexpected Perturbation	5 8
	5.1	Introd	uction	59
	5.2	Relate	ed Work	62
	5.3	Metho	odology	63
		5.3.1	Latent Physics Space for Full-body Motions	64
		5.3.2	Skeleton Restoration Model	68
		5.3.3	Training with Auxiliary Losses	69
	5.4	Exper	iments	69
		5.4.1	Dataset and Metrics	69

		5.4.2	Baselines	'0
		5.4.3	Quantitative Results	0
		5.4.4	Qualitative Results	'2
		5.4.5	Generalization	' 4
		5.4.6	Ablation Study	'5
	5.5	Conclu	nsion	'6
6	Disc	cussion	, Conclusion and Future Work 7	7
	6.1	Discus	$sion \ldots .$	'8
	6.2	Conclu	usion	30
	6.3	Future	e Work	\$2
A	Sup	pleme	ntary Material for Chapter 3 8	4
	A.1	Additi	onal Experiments	\$5
		A.1.1	Generalization to Unseen Scenarios	\$5
		A.1.2	Interpretability of Prediction	6
		A.1.3	Ablation Experiments	37
	A.2	Details	s of NSP-SFM	8
		A.2.1	Goal Sampling Network	;9
		A.2.2	Conditional Variational Autoencoder)1
	A.3	Impler	nentation Details	12
в	Sup	pleme	ntary Material for Chapter 4 9	4
	B.1	Additi	onal Explainability Experiments	15
	B.2	Netwo	rk Details	15
	B.3	Impler	nentation Details	17
С	Sup	pleme	ntary Material for Chapter 5 9	8
	C.1	Additi	onal Experiments	19
		C.1.1	Single-person Results	9
		C.1.2	Multi-people Results)3
		C.1.3	Data Efficiency)6
	C.2	Additi	onal Experiment Details 10	18
		C.2.1	Dataset Details	18
		C.2.2	Metrics)9

	C.2.3	Baseline Adaptation	110
	C.2.4	Additional Details of Ablation Study	112
C.3	Additio	onal Details of Methodology	113
	C.3.1	Differentiable Inverted Pendulum Model	113
	C.3.2	Skeleton Restoration Model $\hdots \hdots \$	115
	C.3.3	Training	117
DC			110
Keteren	ices		119

LIST OF FIGURES

1.1	Examples of low-density crowds and high-density crowds	3
3.1	Overview of NSP-SFM.	26
3.2	Architectures of the Goal-Network and Collision-Network in NSP-SFM.	27
3.3	The architecture of the CVAE in NSP-SFM	28
3.4	The neighborhood and view field in NSP-SFM	29
3.5	Interpretability of prediction.	36
3.6	Examples for ablation study.	38
4.1	Overview of BSSFM	45
4.2	The CVAE architecture in BSSFM	47
4.3	Visualization of uncertainties	50
4.4	Visualization comparison.	51
4.5	Generalization experiments.	54
5.1	Overview of LDP	64
5.2	FZJ Push	69
5.3	Perturbations with different magnitudes in single-person. \ldots	71
5.4	Perturbations with different magnitudes in multi-people	72
5.5	Visual results in the single-person scenario.	73
5.6	Multi-people comparison.	73
5.7	Learned net forces	74
5.8	A 13-person group in a diamond formation with three people (indicated $% \mathcal{A}$	
	by orange arrows) being pushed.	74

A.1	The collision rate and the number of collisions against the number of	
	agents are shown in (a) and (b), respectively	87
A.2	The visualization results of generalization to $74, 100, 150, and 200 agents$	
	on coupa0 are shown in (a), (b), (c), and (d), respectively	88
A.3	Examples of interpretability	89
A.4	Motion randomness is captured by our model	89
A.5	Model architecture of the goal sampling network	91
A.6	The architecture of the CVAE	91
R 1	Explainability of behaviors	95
D.1 В 9	Notwork architectures of the actuation block and the neighbor block	06
D.2	The owner when the actuation block and the heighbor block	90
В.3	The CVAE architecture	96
C.1	Visual comparison on pushes with different magnitudes	99
C.2	Perturbations with different magnitudes in single-person. \ldots	100
C.3	The Generalization to an extra strong push	101
C.4	The Generalization to a multi-push scenario. $\hfill \ldots \hfill \hfill \ldots \hf$	101
C.5	Visual comparison on pushes with different magnitudes and group form-	
	ations	104
C.6	Perturbations with different magnitudes in multi-people	105
C.7	Generalization on ten people in a line	105
C.8	FZJ Push	109
C_{0}		116
0.9	The architecture of the CVAE-Lower.	110

LIST OF TABLES

3.1	Results on ETH/UCY and SDD based on standard-sampling	33
3.2	Results on ETH/UCY (left) and SDD (right) based on ultra-sampling	34
3.3	Collision rates on testing data in ETH/UCY and SDD. \ldots	35
3.4	Collision rates of the generalization experiments on ZARA2 (Z) and	
	coupa0 (C)	35
3.5	Ablation study on SDD	37
4.1	Results on SDD based on standard-sampling.	53
4.2	Results on ETH/UCY based on standard-sampling	53
4.3	Results on ETH/UCY and SDD based on ultra-sampling	54
4.4	Results on partial training data of SDD based on ultra-sampling $\ . \ . \ .$	55
4.5	Ablation study on SDD	56
5.1	Metrics in single-person.	70
5.2	Metrics in multi-people.	72
5.3	Ablation study with (1) IPM and no IPM, (2) Full body and Low-up	
	body pose reconstruction	75
A.1	Collision rates of the generalization experiments on Coupa 0	86
A.2	Ablation experiments on network architecture.	90
A.3	Ablation experiments on SDD	90
A.4	Hyperparameters for all six datasets	92
C.1	Comparison with full-body physics-based baselines	102
C.2	Metrics in complete (top) and 25% (bottom) training data for single-	
	person	106

LIST OF TABLES

C.3	Model size in single-person (left) and multi-people (right) 107
C.4	Metrics in complete (top) and 25% (bottom) training data for multi-people.107
C.5	Dataset details
C.6	Adaptation for motion prediction methods
C.7	Basic interaction force on angles

Abbreviations

SFM	Social Force Model	RNN	Recurrent Neural Network
CNN	Convolutional Neural Network	LSTM	Long Short Term Memory
VAE	Variational Autoencoder	GAN	Generative Adversarial Network
CVAE	Conditional Variational Autoen-	NSP	Neural Social Physics
	coder		
GNN	Graph Neural Network	DE	Differential Equation
GSN	Goal Sampling Network	MLP	Multi-layer Perceptron
NN	Neural Network	PDE	Partial Differential Equation
SDD	Stanford Drone Dataset	FPS	Frames Per Second
ADE	Average Displacement Error	FDE	Final Displacement Error
S-GAN	Social GAN	CF-VAE	Conditional Flow VAE
CGNS	Conditional Generative Neural	BSSFM	Bayesian Stochastic Social Force
	System		Model
SDE	Stochastic Differential Equation	DGM	Deep Generative Model
HNP	Highest Number of People	LDP	Latent Differentiable Physics
DP	Differentiable Physics	IPM	Inverted Pendulum Model
MSE	Mean Squared Error	PD	Proportional Derivative
DIM	Differential Interaction Model	MPJPE	Mean Per Joint Position Error
hipADE	Average Displacement Error at	hipFDE	Final Displacement Error at the
	the hip		hip
MBLE	Mean Bone Length Error	FSE	Foot Skating Error
TTST	Test-time Sampling Trick	Dof	Degree of Freedom
A2M	Action2Motion	ACTOR	Action-conditioned Transformer
			VAE
MDM	Motion Diffusion Model	RMDiffuse	Retrieval-augmented Motion
			Diffusion model
PhyVae	Physics-based VAE	BE/BA	Before/Back

Chapter 1

Introduction

1.1 Background

Different research fields like transportation and computer science have their specific definitions of a crowd for diverse research objectives. This thesis focuses on the general crowds that have the definition: a crowd is a group of individuals in the same physical location and at the same time [1]. Crowd research is crucial across various fields, including public safety [2, 3], urban planning [4, 5], virtual reality [6, 7], etc. As the global population grows rapidly, crowd research attracts more and more attention from academia and industry [8]. Given crowd research is a broad topic, this thesis focuses on the study of crowd behaviors which is one of the most important subtopics [9, 10]. Crowd behavior research has extremely high application value in many areas. For instance, understanding crowd behaviors in public safety contributes to developing strategies to mitigate risks from large gatherings of people like stampedes and crushes [2]. Crowd behaviors into the design of transportation systems, large buildings such as shopping malls and concert halls, etc. to enhance the overall functionality and livability of urban areas [4, 5].

We aim to understand underlying behavior patterns in crowds by machine learning in this thesis. Although there has been a lot of outstanding work across various fields in crowd behavior research [11–16], many challenges remain because of complex human behaviors [17, 18]. We concentrate on two main challenges in this thesis, which are crowd dynamics modeling in low-density crowds and physical interaction modeling in high-density crowds. High-density crowds typically refer to crowds whose densities exceed 5 people per square meter, while low-density crowds have 5 or fewer people per square meter [19]. Crowds are classified by density in crowd behavior research [20] because low-density and high-density crowds have significantly different behavior patterns. For instance, physical interactions are common in high-density scenarios but scarce in low-density scenarios. Examples of low-density and high-density crowds can be found in Fig. 1.1.

Crowd dynamics modeling in low-density crowds is crucial for crowd management, crowd control, urban planning, *etc.* Individuals tend to be regarded as mass points to model crowd dynamics in low-density crowds efficiently. The influence of bodies and poses is ignored due to the scarcity of physical interactions in low-density crowds. However, despite using the simplification, crowd dynamics remain complicated. This is

1.1 Background



Low-density Crowd

High-density Crowd

Figure 1.1: Examples of low-density crowds [21] and high-density crowds [22]. Pedestrians are marked by purple squares in the low-density crowd.

because crowd movement is influenced by various factors including individual psychology, social norms, environment, etc. [23] and has intrinsic stochasticity. Therefore, it is typically difficult to model crowd dynamics, especially in complex contexts. In summary, crowd dynamics modeling in low-density scenarios is challenging but significantly meaningful.

For high-density crowds, we focus on modeling physical interactions between individuals rather than overall crowd dynamics, where physical interactions refer to limb-to-limb contact between individuals e.g. pushing and pulling. This is because physical interactions are the indispensable factor for accurate high-density crowd dynamics modeling, but they still haven't been studied well [18]. The achievements in physical interaction modeling can facilitate the accurate modeling of complex phenomena in dense crowds such as crowd turbulence [24], density waves [25], and so on. Then this will drive the development of crowd dynamics modeling in high-density scenarios, which plays an important role in multiple domains e.g. public safety, crowd management, etc. Modeling physical interactions is crucial, yet it remains a highly challenging task. The first challenge is data. It's hard to record the direct influence e.g. forces from physical interactions. One possible way is to record the data containing physical interactions such as full-body human motions. Even so, existing data related to physical interactions are extremely scarce. Another challenge is the complexity of physical interactions. Specifically, multiple factors can influence physical interactions, including contact positions, interaction magnitude, and individual psychology. Moreover, we can't overlook the variability of individuals since different people can react diversely to the same contact force.

1.2 Research Questions

We conduct crowd behavior research in two directions: crowd dynamics modeling in low-density crowds and physical interaction modeling in high-density crowds. For the former, people and their movements are generally viewed as 2D points and trajectories consisting of a series of 2D points [11, 15, 26], respectively, because specific body shapes do not influence the crowd dynamics observably and thus can be ignored in low-density scenarios. Considering that predicting future movements is a critical capability of crowd dynamics models and is significantly valuable in many applications *e.g.* crowd management [27, 28], this thesis explores crowd dynamics modeling in low-density crowds by solving the problem of human trajectory prediction. Specifically, we pay attention to the following two research questions:

1. How to predict human trajectories accurately given their past trajectories and the environment while possessing the explainability of predictions?

2. Uncertainty analysis is vital in crowd dynamics and human trajectories. How can we model the uncertainty in human trajectory prediction?

As for the physical interaction modeling in high-density crowds, it is hard to capture the interaction data such as forces of pulls and pushes accurately. To address this problem, we use 3D full-body motion data including physical interactions since the motion data are the direct results of these interactions and are captured more easily. Then, to obtain physical interaction models influencing 3D full-body human motions accurately, we propose and solve a research question:

3. How to predict 3D full-body human motions, on both individual and group levels, under unexpected physical perturbation?

In the rest of this thesis, human trajectories denote human 2D point trajectories and human motions denote human 3D full-body motions if not specified.

1.3 Our Methods

We propose three methods, presented in Chapters 3 - 5, to solve the aforementioned corresponding research questions. Their overviews are presented as follows.

Question 1: How to predict human trajectories accurately given their past trajectories and the environment while possessing the explainability of predictions?

Human trajectory prediction is a crucial task for understanding crowd dynamics and has been studied in many research areas like computer vision, robotics, and transportation [29–31]. In addition, this task plays a vital role in many applications such as automatic driving, service robots, and transportation systems [28, 32]. However, predicting the crowd movement is highly challenging because of multiple complicated impact factors like social interactions and complex environments. Early methods [11, 33, 34] tend to explicitly model human behaviors to predict future trajectories, which possess good explainability but cannot obtain highly accurate predictions. More recently, model-free methods [15, 26, 35] based on deep learning have shown their surprising prediction accuracy and prevail in human trajectory prediction. However, the lack of explainability prevents applications of these methods in other areas such as crowd simulation. To the end, our first work designs a new framework considering explainability and prediction accuracy simultaneously based on neural differential equations [36, 37], where we model crowd dynamics as physical models. In this framework, we propose a novel trajectory prediction model that performs well on both explainability and prediction accuracy by combining the traditional social force model (SFM) [11] with neural networks.

Question 2: Uncertainty analysis is vital in crowd dynamics and human trajectories. How can we model the uncertainty in human trajectory prediction?

Uncertainty in the trajectory prediction task is that a past trajectory can correspond to multiple future trajectories. The one-to-many relationship arises because humans make movement decisions with randomness based on a complex mixture of conscious and unconscious factors. Uncertainty modeling is employed widely in applications such as safety-critical autonomous vehicles and crowd evacuation [38, 39]. However, it is challenging to model uncertainty consisting of the data and model uncertainty [38, 40] in human trajectories, as it needs to consider many factors like social interactions, the environment, and even unknown factors. Moreover, existing methods that focus on uncertainty either consider solely the data uncertainty or mix these two different uncertainties, which limits the explainability of models. To obtain better explainability, we propose a novel Bayesian stochastic social force model based on deep learning, which decouples the uncertainty in human trajectories. Our method models and estimates the data and model uncertainty individually and introduces an uncertainty-aware training scheme to ensure the effect of decoupling.

Question 3: How to predict 3D full-body human motions, on both individual and group levels, under unexpected physical perturbation?

3D Full-body human motion prediction is a well-established task and has been studied in depth [41-44], which aims to predict future motions given past movement. We propose the new research question extending the existing task by incorporating physical perturbation to understand human motions further and study physical interactions between individuals. The research of the new task can facilitate that human motion prediction techniques are applied in more new domains such as reactive motions for character animation [14, 45], crowd crush with physical interactions [3, 46], balance recovery in biomechanics [47, 48], etc. Nevertheless, new challenges arise as the introduction of physical perturbation. Specifically, the motions in the new task are much more complex than those in traditional human motion prediction, where physical interactions and their propagation need to be modeled. Moreover, the available data are scarce because of the difficulty of capturing motions with physical interactions. To overcome the aforementioned challenges, we propose the latent differentiable physics model. 3D Full-body poses are mapped onto a scalable differentiable physics model to reduce the complexity of motions incorporating physical interactions. Our model is trained to estimate motions on the level of the physics model, which requires small data due to the simplicity of the physics model. Neural networks are used to restore 3D full-body poses under the guidance of estimations based on the physics model.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows:

Chapter 2: Related Work

This chapter offers a comprehensive literature review of research fields covered in Chapters 3, 4, and 5, including human trajectory prediction, uncertainty in human trajectory prediction, crowd simulation, traditional research on physical interactions, human motion prediction, neural differential equations, and differentiable physics.

Chapter 3: Human Trajectory via Neural Social Physics

This chapter introduces a novel framework considering prediction accuracy and explainability for human trajectory prediction and proposes a new model based on SFM and neural networks under this framework. Extensive experimental results demonstrate that our model has state-of-the-art trajectory prediction accuracy, good crowd behavior explanation, and strong generalizability.

Chapter 4: Uncertainty-aware Human Trajectory Prediction

This chapter proposes a novel Bayesian stochastic social force model that captures the fine-grained uncertainty in human trajectories through a decoupling strategy. Through experiments, we demonstrate that our model possesses strong explainability while outperforming existing methods in prediction accuracy.

Chapter 5: Human Motion Prediction under Unexpected Perturbation

This chapter proposes a new task setting: predicting human motions, on both individual and group levels, under unexpected physical perturbation to study physical interactions between people. To this end, we have identified new challenges and proposed a novel deep learning model based on differentiable physics to address them. Through extensive evaluation, we demonstrate the advantages of our model in several aspects including prediction accuracy, data efficiency, generalizability, and explainability.

Chapter 6: Conclusion and Future Work

This chapter presents the research conclusions of this thesis and discusses potential future work.

Chapter 2

Related Work

This chapter provides a thorough literature review of research topics covered in Chapters 3, 4, and 5. First, we review methods related to our study on crowd dynamics modeling in low-density crowds in Section 2.1. In Section 2.2, we present literature related to our research on physical interactions in high-density crowds. Lastly, Section 2.3 reviews literature about neural differential equations and differentiable physics, which is crucial to the methods presented in Chapters 3, 4, and 5.

2.1 Crowd Dynamics Modeling in Low-density Crowds

This thesis studies human trajectory prediction and its uncertainty modeling to explore crowd dynamics in low-density crowds, covered in chapters 3 and 4, respectively. Consequently, we discuss primary approaches to predicting human trajectories in Section 2.1.1. Subsequently, mainstream uncertainty models in human trajectory prediction are reviewed in Section 2.1.2. Finally, Section 2.1.3 presents the related literature in the field of crowd simulation, to which our methods in chapters 3 and 4 also contribute.

2.1.1 Human Trajectory Prediction

Modeling human trajectories is key to understanding crowd dynamics. Therefore, a wide range of domains have been studying this problem. With the development of technology, existing approaches can generally be divided into model-based, model-free, and hybrid approaches, based on the proportion of explicitly modeling crowd behaviors in approaches. Specifically, it is the core of model-based methods to model crowd behaviors explicitly. Model-free methods focus on learning crowd behaviors from data. Hybrid methods bridge the two categories of methods to leverage the strengths of both. Our model, as detailed in Chapter 3, is among the first hybrid methods., which exploits the advantages of model-based and model-free methods.

Model-based Methods. These methods tend to be early human trajectory prediction work. They typically describe human trajectory movement through ordinary/partial differential equations, optimization problems, *etc.* based on their proposed fundamental hypotheses about human trajectories. For example, [11, 49, 50] assume that Newton's laws of motion govern human trajectory movement and offer various modeling of forces, which uses differential equations to represent and predict the movement. [21, 51] formulate human trajectory prediction as an energy minimization problem, in which energy functions are designed to describe factors influencing trajectories, including individual characteristics, social norms, and environmental conditions.

Model-free Methods. Rapid development of deep learning has facilitated the prosperity of model-free methods in human trajectory prediction. Predominant model-free methods generally are based on deep learning, leveraging its strong data-fitting ability. Therefore, this thesis focuses on model-free methods based on deep learning. With the progress of deep learning, a variety of networks have been introduced to the task, including Recurrent Neural Networks (RNNs) [52], Convolutional Neural Networks (CNNs) [53], generative networks [54–56], transformers [57], etc. Consequently, we can classify broadly existing model-free methods based on deep learning according to network architectures into RNN-based, CNN-based, generative networks-based, and transformer-based.

RNN-based Methods. RNNs were first used in human trajectory prediction because of their excellent performance in sequence prediction tasks [58, 59]. Social-LSTM [15] is the pioneering work extending an RNN to human trajectory prediction, which proposes a model based on the Long Short Term Memory (LSTM) network to learn human movement. This LSTM-based model assigns an LSTM to each pedestrian to model their dynamics and introduces a social pooling layer connecting different LSTMs to learn social interactions. The success of [15] has inspired the subsequent RNN-based methods [60– 64]. Specifically, Bartoli *et al.* [60] designed a context-aware LSTM model that not only considers social interactions but includes human-environment interactions. Tran *et al.* [63] proposed a dual-channel recurrent neural network, where the goal channel predicts destinations while the trajectory channel estimates complete future positions based on results from the goal channel.

CNN-based Methods. CNNs receive interest in modeling human trajectories because of their strong capability of capturing spatial and temporal correlation and computationally efficient parallel operations. The first end-to-end convolutional neural network proposed by [65] encodes each position of observed trajectories to corresponding embedding vectors and the concatenated vectors are fed into a CNN to extract the history features. A fully connected layer, finally, takes the history features as inputs and estimates future trajectories. Subsequently, more variants of CNN [66–69] were proposed to predict human trajectories. For instance, Social-STGCNN [66] constructs the spatio-temporal graph representation of input trajectories fed into a graph convolution neural network to extract features. Then these features are passed through the timeextrapolator convolution neural network to generate future trajectories. Shi *et al.* [67] proposed a sparse graph convolution network model to address redundant interactions, which employs the sparse directed spatial graph to adaptively capture interactions and the sparse directed temporal graph to estimate movement tendency. Then a sparse graph convolution network takes the two graphs as inputs to output history trajectory feature which is fed into a time convolution network to predict future positions.

Generative network-based Methods. With the significant progress of generative networks, capturing accurate distributions of future trajectories has become possible. Considering the intrinsic stochasticity of human trajectories, it has attracted much attention to predict the distributions rather than a single future prediction given the observed trajectory. To obtain accurate future distributions, multiple different generative networks have been used in the task, including Generative Adversarial Networks (GANs) [26, 70, 71], Conditional Variational Autoencoders (CVAEs) [72–74], diffusion models [75, 76], etc. Specifically, Gupta et al. [26] designed a new GAN to predict distributions of future trajectories with a novel pooling mechanism modeling social interactions, which is one of the earliest generative network-based methods. PECNet [73] is a goal-conditioned method that predicts distributions of future trajectories based on distributions of destinations estimated by a CVAE. Inspired by diffusion models, Gu et al. [75] modeled the motion indeterminacy diffusion to predict future trajectories, where the desired trajectory can be estimated from walkable areas by progressively removing indeterminacy.

Transformer-based Methods. Transformers utilize attention mechanisms to encode global features, which can capture global information and long-range dependencies effectively. This drives the application of transformers in human trajectory prediction [77–79]. For example, Shi *et al.* [78] designed a novel transformer to predict future trajectories, which models human movement on temporal and social dimensions simultaneously. The authors introduced a new agent-aware attention mechanism to solve the problem that the original attention mechanism ignores the agent identity. TUTR [79] employs a clustering model and a mode-level transformer encoder to capture the multimodality of future trajectories and uses a social-level transformer decoder to model social interactions. Features from the encoder and decoder, finally, are passed through two fully connected networks to predict multimodal future trajectories and their probabilities.

Hybrid Methods. These methods aim to incorporate the advantages of modelbased and model-free methods. Model-based methods tend to have strong explainability from explicit behavior modeling but perform poorly in prediction accuracy because of their limited data-fitting ability. In contrast, model-free methods based on deep learning tend to achieve outstanding performance in prediction due to the strong datafitting capability of deep learning. However, the black-box nature of these modelfree methods leads to a lack of explainability, which limits their applications in other domains such as crowd simulation and analysis. Hybrid methods [80–82] expect to obtain high prediction accuracy while retaining explainability. Our work [80] in Chapter 3 embeds crowd physics models into neural networks to combine the advantages of both model-based and model-free approaches. Subsequently, the proposed method in [81] integrates a data-driven neural ordinary differential equation and a model-based computational graph for the trajectory prediction task while [82] combines socially explainable energy maps with CVAEs to model stochasticity in trajectory data.

2.1.2 Uncertainty in Human Trajectory Prediction

An important observation is that multiple future trajectories are plausible given the history trajectory [26, 83]. Therefore, it is necessary to model this uncertainty/stochasticity in human trajectory prediction. Moreover, the modeling of the trajectory uncertainty plays a crucial role in many applications such as crowd evacuation and autonomous vehicles [38, 39]. Although the uncertainty of trajectories is often overlooked in early trajectory prediction work [11, 15, 51], recent mainstream methods [79, 84, 85] have emphasized the uncertainty and studied stochastic trajectory prediction. Existing methods focusing on modeling the trajectory uncertainty can be divided into explainable and unexplainable according to the explainability of uncertainty modeled in these methods. Explainability is a delimiter since explainable methods generally are more important than unexplainable ones, especially in safety-critical tasks like autonomous vehicles, by providing human-understood reasons for predicted trajectories [38].

The explainable methods typically model the trajectory uncertainty explicitly. For example, the Gaussian distribution and the Gaussian mixture model have been exploited in many methods [86–88] to represent uncertainty. Specifically, [88] exploits a Gaussian mixture model involving multiple predefined movement patterns to capture multimodal future trajectories explicitly and introduces an artificial potential field to improve prediction accuracy and safety. In addition, Shi *et al.* [85] proposed a treebased stochastic trajectory prediction model that uses a coarse-to-fine scheme, where the trajectory uncertainty can be explained by paths in the tree. The model initially constructs a coarse tree with acceptable complexity and enough coverage and optimizes the tree greedily to capture uncertainty. Finally, the coarse predictions from the coarse tree are refined by the teacher forcing to generate fine predictions. Moreover, Shi *et al.* [79] used a clustering model to estimate general motion modes that can represent various pedestrian behaviors *e.g.* going straight and turning and designed a transformer conditioned on the motion modes to predict multimodal future trajectories and corresponding probabilities.

On the contrary, the unexplainable methods generally model the trajectory uncertainty implicitly. For instance, [89] employs Bayesian neural networks [90] whose weights are probabilistic to model human movement and use Monte Carlo dropout [91] to estimate the uncertainty in trajectories, where uncertainty is learned from data via probabilistic network weights implicitly. Amirian *et al.* [84] utilized the GAN to generate plausible future trajectories given the observed trajectories, in which the trajectory uncertainty is captured implicitly by training the GAN. This model extends the Info-GAN [92] to stochastic trajectory prediction and proposes a new attentionbased pooling layer incorporating some hand-crafted interaction features. Similarly, many unexplainable methods employ other generative networks to model uncertainty implicitly, including CVAE-based [72, 80, 93], diffusion model-based [75, 76], *etc.*

Our work in Chapter 4 is an explainable stochastic trajectory prediction method. Furthermore, our model explores the fine-grained uncertainty structure, in contrast to existing methods with coarse-grained uncertainty. Therefore, our model performs better in explainability compared with existing methods and enhances prediction accuracy slightly.

2.1.3 Crowd Simulation

Crowd simulation aims to simulate the movement of virtual individuals given initial conditions such as initial positions and predefined destinations [94]. This field has advanced significantly and continues to develop rapidly, which has two foundations: empirical and data-driven methods.

The vast majority of crowd simulation methods are empirical methods that typically abstract crowd dynamics into deterministic systems based on rules and hypotheses predefined carefully. These methods model crowd behaviors at different scales and generally fall into microscope, microscope, and mesoscope. Microscopic methods highlight individual-level motion details and features, which include force-based models [95–97], velocity-based models [98–100], vision-based models [101–103], agent-based models [104–106], etc. By comparison, macroscopic methods tend to ignore individuallevel information and view the crowd as a unified and continuous entity to simulate crowd behaviors using global solvers. As a result, these macroscopic methods commonly aim to simulate large-scale and dense crowds. Mainstream macroscopic methods contain potential field-based models [107–109], aggregate dynamics [110], continuum models [12, 111, 112], etc. Recently, mesoscopic methods considering local and global features simultaneously attracted the attention of researchers, which involves interactive formation models [113–115], dynamic group models [116–118], social psychological models [119–121], etc.

More recently, as more and more real crowd data are captured, data-driven methods have been drawing increasing interest, which learn crowd dynamics from real-world data. For example, In [122], Charalambous and Chrysanthou proposed a real-time agent-based crowd simulation method, which introduces the perception-action graph constructed based on the input data to guide the simulation. Yao *et al.* [123] established a crowd simulation framework based on a residual network, in which the residual network predicting future movement is trained on real data and used to enhance the visual realism of simulated crowd movement. Chen *et al.* [124] proposed a social physics-based diffusion model to generate the diversity of crowd behaviors in simulation results. This trained model estimates the distribution of the next position through the reverse diffusion process based on current interaction dynamics and historical movement, which is guided by physics models.

Both of the proposed methods in Chapter 3 and 4 are novel data-driven crowd simulators. Distinct from existing methods, they construct the differential forward movement estimation based on neural differential equations and predict required parameters using neural networks to achieve good performance in many tasks including crowd simulation while our method in Chapter 4 further explores the multimodality of crowd dynamics by modeling fine-grained uncertainty.

2.2 Physical Interactions in High-density Crowds

We present traditional research on physical interactions in Section 2.2.1 across different disciplines. Deviating from traditional methods, this thesis studies physical interactions by solving the new full-body human motion prediction task incorporating perturbation. Therefore, we discuss mainstream methods in human motion prediction in Section 2.2.2.

2.2.1 Traditional Research on Physical Interactions

Physical interactions can influence individual-level behaviors and crowd-level movements. Then physical interaction research focuses on how people respond to external forces and how these forces propagate in crowds, which helps to prevent dangerous situations causing discomfort or deaths such as losing balance and crowd disasters [18, 48]. Further, related research attracts attention from multiple domains such as balance recovery [47, 48], crowd simulation [3, 46], and motion synthesis of character animation [14, 125]. In this section, we review the predominant traditional approaches on physical interaction modeling, which are from disciplines: biomechanics, dynamical systems, and computer graphics.

Biomechanics emphasizes the coordination and change of bones, muscles, and joints within the human body during/after physical interactions. For example, [48, 126] focus on biomechanical mechanisms such as trunk rotation and center-of-mass movement for balance recovery after physical interactions. [127, 128] conducted research on biomechanical responses like the change of hips to external perturbations during motions such as walking. The aforementioned work tends to be based on existing muscle and musculoskeletal models that describe how muscles, bones, and joints within bodies systematically support human movement and interaction. In addition, many researchers concentrate on developing muscle and musculoskeletal modeling [129–132] for further understanding of the human body and interaction. For instance, Ross *et al.* [129] connected the traditional Hill-type muscle model [133] with an oscillating external load to propose a novel forward dynamics framework for muscle modeling, while van Soest *et al.* [132] improved the musculoskeletal modeling and simulations by introducing cross-bridge muscle models [134].

Research on physical interactions in dynamical systems can be generally categorized as macroscopic or microscopic methods. The former [12, 13, 24, 135] describes the crowd as a continuum and focuses on the evolution of the density and mean velocity of the crowd containing physical interactions in space and time. For example, Hughes [13] proposed the "thinking fluid" incorporating human decision and interaction based on fluid dynamics. Golas et al. [24] proposed a continuum method to model the crucial phenomena in high-density crowds "crowd turbulence", where discomfort and friction in crowds determine the stress fields that represent physical interactions. Microscopic methods consider the heterogeneity of individuals and pay attention to the positions and velocities of individual entities. For instance, the social force model [11] describes pedestrians as particles in the particle system governed by Newton's laws of motion and defines social forces including driving, repulsive, and attractive forces to model interactions between individuals and govern the movement of pedestrians. Subsequently, a lot of methods [136–138] generalized and developed the social force model and enabled it to cover various natural crowd phenomena. Moreover, a large number of microscopic approaches [139–141] model crowd behavior based on cellular automata models [142], where pedestrians are regarded as automata in cells and the walkway is represented as grid cells. Distinct from the aforementioned microscopic methods ignoring the physical shapes of individuals, some approaches [3, 18] try to model more accurate physical interactions by introducing 3D body models to understand the further microscopic mechanism.

In computer graphics, researchers are mainly interested in how to simulate physical interactions through computer technologies. For instance, Arikan *et al.* [14] presented an algorithm to simulate the motion of the character being pushed by external forces via selecting and modifying motions from the recorded collection of motions of a real person being pushed. Mordatch *et al.* [125] proposed a motion synthesis framework that can simulate a broad range of human behaviors including interaction actions of two characters based on the contact-invariant optimization method capable of optimizing contact and behavior simultaneously. In addition, [111, 143] study the impact of physical interactions on crowd behaviors. For example, van Toll *et al.* [143] designed a crowd simulation model based on smoothed particle hydrodynamics, where agents interact as fluid particles.

Overall, traditional research on physical interactions either cannot learn from data or only possesses limited learning ability. As a result, it is hard to solve the new research question proposed in Chapter 5, predicting motions under physical interactions, for traditional methods. We identify challenges imposed by the research question and overcome them to solve the new task by proposing a latent differentiable physics model combining deep neural networks with differentiable physics in Chapter 5.

2.2.2 Human Motion Prediction

Unlike traditional research, we model physical interactions by proposing and solving a new human motion prediction task that considers perturbation. Therefore, we review existing human motion prediction approaches in this section, which broadly fall into traditional and deep learning methods. Traditional methods commonly predict human motions by applying hidden Markov models [144, 145], restricted Boltzmann machines [146, 147], Gaussian processes [148–150], *etc.* For example, Lehrmann *et al.* [144] proposed the Dynamic Forest Model based on autoregressive trees and the non-parametric and nonlinear Markov model to model human motions, which can infer efficiently and represent complex distributions. Taylor *et al.* [146] introduced a nonlinear autoregressive model to generate and predict human motions by using a conditional Restricted Boltzmann Machine to learn local restraints and global movement of human motions. Wang *et al.* [148] proposed the Gaussian Process Dynamical Model that employs Gaussian process regression to learn motion dynamics and observation mappings.

Deep learning methods have dominated human motion prediction recently because of the strong prediction capability of neural networks. Early work deemed human motion prediction as sequence-to-sequence prediction tasks and solved it based on recurrent neural networks [151–155]. As an example, the Dropout Autoencoder LSTM introduced by Ghosh *et al.* [154] employs a recurrent neural network (LSTM) to predict human motions and prevents error accumulation by designing an autoencoder to refine the predictions. Then, convolutional networks are employed for human motion prediction [156–160] considering their strength of capturing spatial features. For instance, Li *et al.* [157] proposed a convolutional sequence-to-sequence model, which uses a convolutional encoder to extract features from the known motions and introduces a decoder consisting of convolutional networks and fully-connected networks to predict future motions based on the features. Cui *et al.* [158] constructed the dynamic graphs to represent skeleton poses and exploit the natural connectivity of human joints explicitly and solved the task by introducing a generative model based on graph convolution networks. Subsequently, deep generative models *e.g.* GAN, CVAE, diffusion models, etc. facilitated the exploration of inherent stochasticity in human motions [161–165]. To illustrate, BiHMP-GAN [162] is a probabilistic generative model that generates and predicts human motions given input motions and a random vector from a predefined prior distribution, in which the novel bidirectional framework and recursive prediction scheme are designed to regularize the training. More recently, transformer-based methods [166–168] performed well in catching long-term spatial and temporal correlations through the attention mechanism. For example, Aksan et al. [167] proposed a spatiotemporal transformer model, introducing a self-attention mechanism decoupling space and time for identifying effective information from input sequences. Besides, multipeople motion prediction has been increasingly popular recently [16, 43, 169, 170]. In [169], a framework for multi-people motion prediction has been proposed, where individual motions and social interactions are modeled by the local and global encoders, respectively.

However, existing motion prediction methods tend to neglect the modeling of explicit physical interactions between individuals. As a result, they generally can't perform well in the new task which is to predict motions involving physical interactions. This limits the effectiveness of these methods in many application domains such as balance recovery research and humanoid robots. Our latent differentiable physics model in Chapter 5 addresses the challenging new task by combining differentiable physics with deep generative models. Different from existing methods, our model proposes a novel latent differentiable physics space to learn body physics with physical interactions, which can alleviate data scarcity and ensure high scalability, while a new differentiable interaction model is introduced to learn explicit physical interactions and their propagation.

2.3 Neural Differential Equations and Differentiable Physics

The combination of differential equations and deep learning is a rapidly developing research area [36, 171–174]. Our work within this thesis is inspired by related research including neural differential equations and differentiable physics. Neural differential equations aim to parameterize the vector fields of differential equations using neural networks, which can fill the gap between theory and observation. Therefore, they have

a wide range of applications, *e.g.* computer vision [175-177], climate science [178, 179], and time series modeling [36, 180, 181]. To be specific, Ramadhan *et al.* [178] designed a data-driven parameterization framework for climate models, where neural differential equations are employed to enhance existing base parameterizations by capturing the residual fluxes. Meng *et al.* [176] applied neural stochastic differential equations to guided image synthesis and editing. Given a user guide as an input image, their method adds Gaussian noise to the input and then denoises it through reverse neural stochastic differential equations to output desired images. Chen *et al.* [36] constructed neural ordinary differential equations to model time series data, which regards hidden states as continuous variables rather than discrete ones in traditional neural networks such as RNN and parameterize their derivatives through neural networks.

Differentiable physics applies deep learning technologies and frameworks to convert physics models usually containing differential equations into differentiable systems. It typically offers strong data-fitting capability, explainability, generalizability, etc. Consequently, differentiable physics has been used and developed by a large number of domains: robotics [172, 182, 183], computer graphics [184, 185], fluid dynamics [173, 186, 187], etc. Specifically, Dojo proposed in [172] is a differentiable simulator based on rigid body dynamics for robotics, which utilizes the implicit-function theorem to obtain smooth gradients. Gong *et al.* [185] proposed a novel differentiable fabric model to simulate various composite materials, which introduces differentiable forces to realize gradient-based learning. The model considers the physics of the single yarn and interactions between yarns simultaneously to achieve good simulation results. Nava etal. [173] established a differentiable physics model for fluid-structure interaction, which uses a physics-based neural network surrogate combined with a 2D differentiable model simulating the deformable solid structure to model hydrodynamic effects. Additionally, the method introduces a differentiable layer to build the interaction between the solid and the fluid.

Our work in Chapter 3 and 4 is inspired by neural differential equations. The method in Chapter 3 extends neural differential equations to human trajectory prediction, while our method in Chapter 4 studies the application of neural differential equations in trajectory uncertainty modeling. The work in Chapter 5 is motivated by differentiable physics, where our method extends differentiable physics further to a new research question about 3D human full-body motions.

Chapter 3

Human Trajectory via Neural Social Physics This chapter aims to study 2D human trajectory prediction, where pedestrians are represented as 2D points, and their trajectories are defined as a sequence of 2D points. Trajectory prediction has been widely pursued in many fields, and many *model-based* and *model-free* methods have been explored. The former include rule-based, geometric or optimization-based models, and the latter are mainly comprised of deep learning approaches. In this chapter, we propose a new framework (Neural Social Physics or NSP) combining both methodologies based on neural differential equations. NSP is a class of deep neural networks within which we use explicit physics models with learnable parameters. The explicit physics models serve as a strong inductive bias in modeling pedestrian behaviors, while the rest of the networks provide a strong data-fitting capability in terms of system parameter estimation and dynamics stochasticity modeling. We introduce a new model under this framework by combining the social force model with neural networks. We compare our model with 15 recent deep learning methods on 6 datasets and improve the state-of-the-art performance by 5.56%-70%. Besides, we show that our model has better generalizability in predicting plausible trajectories in drastically different scenarios where the density is 2-5 times as high as the testing data. Finally, we show that the physics model in our model can provide plausible explanations for pedestrian behaviors, as opposed to black-box deep learning. Code is available: https://github.com/realcrane/Human-Trajectory-Predictionvia-Neural-Social-Physics.

3.1 Introduction

Understanding human trajectories is key to many research areas such as physics, computer science, and social sciences. Being able to learn behaviors with non-invasive sensors is important to analyzing the natural behaviors of humans. This problem has been widely studied in computer graphics, computer vision, and machine learning [188]. Existing approaches generally fall into *model-based* and *model-free* methods. Early model-based methods tended to be empirical or rule-based methods derived via the first-principles approach: summarizing observations into rules and deterministic systems based on fundamental assumptions on human motion. In such a perspective, social interactions can be modeled as forces in a particle system [11] or an optimization problem [33], and individuals can be influenced by affective states [189]. Later, data-driven model-based methods were introduced, in which the model behavior is still dominated by the assumptions on the dynamics, *e.g.* a linear dynamical system [190], but retains sufficient flexibility so that the model can be adjusted to fit observations. More recently, model-free methods based on deep learning have also been explored, and these demonstrate surprising trajectory prediction capability [15, 26, 35, 70, 73, 74, 83, 93, 191–198].

Empirical or rule-based methods possess good explainability because they are formed as explicit geometric optimization or ordinary/partial differential equations where specific terms correspond to certain behaviors. Therefore, they have been used for not only prediction but also analysis and simulation [199]. However, they are less effective in data fitting with respect to noise and are therefore unable to predict accurately, even when the model is calibrated on data [200]. Data-driven model-based methods (*e.g.*, statistical machine learning) improve the ability of data fitting but are restricted by the specific statistical models employed which have limited capacities to learn from large amounts of data [190]. Finally, deep learning approaches excel at data fitting. They can learn from large datasets, but lack explainability and therefore have been mainly used for prediction rather than analysis and simulation [15, 73, 74].

We explore a framework that can explain pedestrian behaviors and retain good datafitting capabilities by combining model-based with model-free approaches. Inspired by recent research in neural differential equations [36, 37, 201, 202], we propose a new framework consisting of two parts based on them. The first is a deterministic model formulated using a neural differential equation. Although this equation can be arbitrary, we use a dynamical system inspired by the social force model (SFM) [11] to instantiate the framework. In contrast to the social force model and its variants, the key parameters of our deterministic model are learnable through data instead of being hand-picked and fixed. The second part of our framework captures complex uncertainty in the trajectory dynamics and observations via neural networks. In our instantiation, a conditional variational autoencoder is employed as the second component. Overall, the whole framework is a class of deep neural networks with embedded explicit models. We call this framework and the proposed instantiation *Neural Social Physics* (NSP) and NSP-SFM, respectively.

We demonstrate that our NSP-SFM model outperforms the state-of-the-art methods [26, 35, 70, 73, 74, 83, 93, 191–198] in standard trajectory prediction tasks across various benchmark datasets [21, 203, 204] and metrics. In addition, we show that NSP-SFM can generalize to unseen scenarios with higher densities and still predict plausible
motions with less collision between people, as opposed to pure black-box deep learning approaches. Finally, from the explicit model in NSP-SFM, we demonstrate that our method can provide plausible explanations for motions. Formally, (1) we propose a new framework based on neural differential equations and a novel model under this framework for trajectory prediction and analysis. (2) we propose a new mechanism to combine explicit and deterministic models with deep neural networks for crowd modeling. (3) We demonstrate the advantages of the NSP-SFM model in several aspects: prediction accuracy, generalization, and explaining behaviors.

3.2 Related Work

3.2.1 Trajectory Analysis and Prediction

Statistical machine learning has been used for trajectory analysis in computer vision [205–210]. They aim to learn individual motion dynamics [211], structured latent patterns in data [209, 212], anomalies [210, 213], etc. These methods provide a certain level of explainability but are limited in model capacity for learning from large amounts of data. Compared with these methods, our model leverages the ability of deep neural networks to handle high-dimensional and large data. More recently, deep learning has been exploited for trajectory prediction [214]. Recurrent neural networks (RNNs) [15, 60, 61] have been explored first due to their ability to learn from temporal data. Subsequently, other deep learning techniques and neural network architectures are introduced into trajectory prediction, such as Generative Adversarial Networks (GANs) [26], conditional variational autoencoders (CVAEs) [72–74] and Convolutional Neural Networks (CNNs) [66]. In order to capture the spatial features of trajectories and the interactions between pedestrians accurately, graph neural networks (GNNs) have also been used to reason and predict future trajectories [66, 67]. Compared with existing deep learning methods, our method achieves better prediction accuracy. Further, our method has an explicit model that can explain pedestrian motions and lead to better generalizability. Very recently, attempts have been made to combine physics with deep learning for trajectory prediction [215-217]. However, their methods are tied to specific physics models and are deterministic, while NSP is a general framework that aims to accommodate diverse physics models and is designed to be intrinsically stochastic to capture motion randomness.

3.2.2 Crowd Simulation

Crowd simulation aims to generate trajectories given the initial position and destination of each agent [199], which essentially aims to predict individual motions. Empirical modeling and data-driven methods have been the two foundations in simulation [110, 218]. Early research is dominated by empirical modeling or rule-based methods, where crowd motions are abstracted into mathematical equations and deterministic systems, such as flows [110], particle systems [11], and velocity and geometric optimization [33, 219]. Meanwhile, data-driven methods using statistical machine learning have also been employed, e.q., using first-person vision to guide steering behaviors [218] or using trajectories to extract features to describe motions [220, 221]. While the key parameters in these approaches are either fixed or learned from small datasets, our NSP is more general in crowd simulation. It can take existing deterministic systems as a component and provide better data-fitting capacity via deep neural networks. Compared with the aforementioned model-based methods, our NSP can be regarded as using deep learning for model calibration. Our model possesses the ability to learn from large amounts of data, which is difficult for traditional parameter estimation methods based on optimization or sampling [222]. Meanwhile, the formulation of our NSP is more general and flexible than traditional model-based methods.

3.2.3 Deep Learning and Differential Equations

Solving differential equations (DEs) with the assistance of deep learning has recently spiked strong interests [36, 202, 223, 224]. Based on the involvement depth of deep learning, the research can be categorized into deep learning assisted DE, differentiable physics, neural differential equations, and physics-informed neural networks. Deep learning assisted DE involves accelerating various steps during the DE solve, such as Finite Element mesh generation [171, 225]. The deeper involvement of neural networks is shown in differentiable physics and neural differential equations, where the former aims to make the whole simulation process differentiable [182, 184, 226], and the latter focuses on the part of the equations being parameterized by neural networks [36, 201]. Physics-informed neural networks aim to bypass the DE solve and use neural networks for prediction [174, 227]. Highly inspired by the research above, we propose a new neural differential equation framework in a new application domain for human trajectory prediction.

3.3 Methodology

3.3.1 Neural Social Physics (NSP)

At any time t, the position p_i^t of the *i*th pedestrian can be observed in a crowd. Then a trajectory can be represented as a function of time q(t), where we have discrete observations in time up to T, $\{q^0, q^1, \dots, q^T\}$. An observation or *state* of a person at time t is represented by $q^t = [p^t, \dot{p}^t]^T$ where $p, \dot{p} \in \mathbb{R}^2$ are the position and velocity. For most datasets, p is given and \dot{p} can be estimated via finite difference. Given an observation q_n^t of the *n*th person, we consider her neighborhood set Ω_n^t containing other nearby pedestrians $\{q_j^t : j \in \Omega_n^t\}$. The neighborhood is also a function of time $\Omega(t)$. Then, in NSP the dynamics of a person (agent) in a crowd can be formulated as:

$$\frac{dq}{dt}(t) = f_{\theta,\phi}(t,q(t),\Omega(t),q^T,E) + \alpha_{\phi}(t,q^{t:t-M}), \qquad (3.1)$$

where θ and ϕ are learnable parameters, E represents the environment. Functions fand α are realized by neural networks. θ contains interpretable parameters explained later and ϕ contains uninterpretable parameters (neural network weights). The agent dynamics are governed by f which depends on time t, its current state q(t), its timevarying neighborhood $\Omega(t)$ and the environment E. Similar to existing work, we assume there is dynamics stochasticity in NSP. But unlike them which assume simple forms (e.g. white noise) [190], we model time-varying stochasticity in a more general form: as a function of time, the current state and the brief history of the agent, $\alpha_{\phi}(t, q^{t:t-M})$. Then we have the following equation in NSP:

$$q^{T} = q^{0} + \int_{t=0}^{T} f_{\theta,\phi}(t,q(t),\Omega(t),q^{T},E) + \alpha_{\phi}(t,q^{t:t-M})dt, \qquad (3.2)$$

given the initial and final condition $q(0) = q^0$ and $q(T) = q^T$. If we divide the time horizon [0,T] into L smaller intervals $\Delta t = T/L$, we can derive the time-discretized version of Eq. 3.2:

$$q^{T} = q^{0} + \sum_{l=0}^{L-1} (f_{\theta,\phi}^{l \triangle t} + \alpha_{\phi}^{l \triangle t}) \triangle t$$
(3.3)

where $f_{\theta,\phi}^{l riangle t}$ and $\alpha_{\phi}^{l riangle t}$ are the values of $f_{\theta,\phi}$ and α_{ϕ} at l riangle t.

Physics models have been widely used to model crowd dynamics [11, 110]. To leverage their interpretability, we model the dynamics as a physical system in NSP. Assuming the second-order differentiability of p(t), NSP expands q(t) via Taylor's series



Figure 3.1: Overview of NSP-SFM. F_{goal} , F_{col} and F_{env} are estimated in every time step by Goal-Network, Collision-Network, and Eq. 3.8 before solving Eq. 3.5. The output is used to update the position and velocity which are then combined with the estimated noise from α for the final prediction

for a first-order approximation:

$$q(t + \Delta t) \approx q(t) + \dot{q}(t) \Delta t = \begin{pmatrix} p(t) \\ \dot{p}(t) \end{pmatrix} + \Delta t \begin{pmatrix} \dot{p}(t) + \alpha(t, q^{t:t-M}) \\ \ddot{p}(t) \end{pmatrix}, \quad (3.4)$$

where Δt is the time step. The stochasticity $\alpha(t, q^{t:t-M})$ is assumed to only influence \dot{p} . Eq. 3.4 is general and any dynamical system with second-order differentiability can be employed here. Below, we realize NSP by combining a type of physics model, the social force model [11], with neural networks. We refer to our model NSP-SFM.

3.3.2 NSP-SFM

We design the NSP-SFM by assuming each person acts as a particle in a particle system and each particle is governed by Newton's second law of motion. $\ddot{p}(t)$ is designed to be dependent on three forces: goal attraction F_{goal} , inter-agent repulsion F_{col} and environment repulsion F_{env} .

$$\ddot{p}(t) = F_{goal}(t, q^T, q^t) + F_{col}(t, q^t, \Omega^t) + F_{env}(t, q^t, E),$$
(3.5)

where E is the environment and explained later. During the deterministic forward pass, we use a semi-implicit scheme for stability: $\dot{p}^{t+1} = \dot{p}^t + \Delta t \ddot{p}^t$ and $p^{t+1} = p^t + \Delta t \dot{p}^{t+1}$. Considering stochasticity, we have $p^{t+1} = p^t + \Delta t (\dot{p}^{t+1} + \alpha^{t+1})$. In addition, unlike [11], the three forces are partially realized by neural networks. The overall model is shown



Figure 3.2: Architectures of the Goal-Network and Collision-Network in NSP-SFM. Left: Goal-Network with the input (q^t, p^T) and Right: Collision-Network with the input (q_n^t, q_j^t) . The numbers in square brackets show both the number and dimension of the layers in each component.

in Fig. 3.1. Note that, in Eq. 3.1 and Eq. 3.5, we assume p^T is given, although it is not available during prediction. Therefore, we employ a Goal Sampling Network (GSN) to sample p^T . During testing, we either first sample a p^T for prediction or require the user to input p^T . The GSN is similar to a part of Y-net [83] and is pre-trained, which is detailed in the appendix A.

Given the current state and the goal, we compute F_{goal} using the Goal-Network NN_{ϕ_1} in Eq. 3.6 (Fig. 3.2 Left), F_{col} using the Collision-Network NN_{ϕ_2} in Eq. 3.7 (Fig. 3.2 Right) and F_{env} using Eq. 3.8 directly. The Goal-Network encodes q^t then feeds it into a Long Short Term Memory (LSTM) network to capture dynamics. After a linear transformation, the LSTM output is concatenated with the embedded p^T . Finally, τ is computed by an MLP (multi-layer perceptron). In Collision-Network, the architecture is similar. Every agent q_j^t in the neighborhood Ω_n^t is encoded and concatenated with the encoded agent q_n^t . Then k_{nj} is computed. τ and k_{nj} are interpretable key parameters of F_{goal} and F_{col} . The corresponding parameter in F_{env} is k_{env} . Finally, we show our network for α for stochasticity modeling in Fig. 3.3.

Goal attraction. Pedestrians are always drawn to destinations, which can be abstracted into a goal attraction force. At time t, a pedestrian has a desired walking direction e^t determined by the goal p^T and the current position p^t : $e^t = \frac{p^T - p^t}{\|p^T - p^t\|}$. If there are no other forces, she/he will change her/his current velocity to the desired velocity $v_{des}^t = v_0^t e^t$ where v_0^t and e^t are the magnitude and direction respectively.



Figure 3.3: The architecture of the CVAE in NSP-SFM. \bar{p}^{t+1} is the intermediate prediction out of our force model and $\alpha^{t+1} = p^{t+1} - \bar{p}^{t+1}$. Encoder E_{bias} , E_{past} , E_{latent} and decoder D_{latent} are all MLP networks with dimensions indicated in the square brackets. More Details of the network can be found in the appendix A.

Instead of using a fixed v_0 as in [11], we update v_0^t at every t to mimic the change of the desired speed as the pedestrian approaches the destination: $v_0^t = \frac{\|p^T - p^t\|}{(T-t)\Delta t}$. Therefore, the desired velocity is defined as $v_{des}^t = v_0^t e^t = \frac{p^T - p^t}{(T-t)\Delta t}$. The goal attraction force F_{goal} represents the tendency of a pedestrian changing her current velocity \dot{p}^t to the desired velocity v_{des}^t within time τ :

$$F_{goal} = \frac{1}{\tau} (v_{des}^t - \dot{p}^t) \text{ where } \tau = a_1 * sigmoid(NN_{\phi_1}(q^t, p^T)) + b_1, \qquad (3.6)$$

where τ is learned through a neural network (NN) parameterized by ϕ_1 . To ensure that the learned τ value is valid, we apply a sigmoid activation function and two hyperparameters $(a_1 \text{ and } b_1)$ to constrain outputs of the network NN_{ϕ_1} as Eq. 3.6.

Inter-agent Repulsion. Pedestrians often steer to avoid potential collisions and maintain personal space when other people are in the immediate neighborhood (Fig. 3.4 a). Given an agent j in Ω_n^t of agent n and her state q_j^t , agent j repels agent n based on $r_{nj} = p_n^t - p_j^t$:

$$F_{col}^{nj} = -\nabla_{r_{nj}} \mathcal{U}_{nj} \left(\| r_{nj} \| \right), \text{ where } \mathcal{U}_{nj} \left(\| r_{nj} \| \right) = r_{col} k_{nj} e^{-\| r_{nj} \| / r_{col}}, \tag{3.7}$$

where we employ a repulsive potential field $\mathcal{U}_{nj}(||r_{nj}||)$ modeled by a monotonic decreasing function of $||r_{nj}||$. Then the repulsive force caused by agent $j \in \Omega_n^t$ to agent n is the gradient of \mathcal{U}_{nj} . Previously, simple functions such as symmetric elliptic fields were employed for \mathcal{U}_{nj} [11]. Here, we model \mathcal{U}_{nj} as a time-varying field parameterized by k_{nj} which is learned via a neural network. Similar to the learning of τ , we set $k_{nj} = a_2 * sigmoid(NN_{\phi_2}(q_n^t, q_{j,j\in\Omega_n^t}^t)) + b_2$ instead of directly learning k_{nj} , where a_2



Figure 3.4: The neighborhood and view field in NSP-SFM. (a) The neighborhood $\Omega(t)$ of a person is a sector within a circle (centered at this person with radius r_{col}) spanned by an angle ω from the current velocity vector (green arrow). (b) Each person has a view field (orange box) within which the environment repels a pedestrian. The view field is a square with dimension r_{env} based on the current velocity vector (green arrow). The current velocity is along the diagonal of the orange box. (c) The environment is segmented into walkable (red) and unwalkable (blue) areas. Within the view field of the pedestrian in (b), the yellow pixels are the environment pixels that repel the pedestrian. ω , r_{col} and r_{env} are hyperparameters.

and b_2 are hyperparameters. If we have *m* agents at time t in Ω_n^t , the net repulsive force on agent *n* is: $F_{col}^n = \sum_{j=0}^m F_{col}^{nj}$.

Environment Repulsion. Besides collisions with others, people also avoid nearby obstacles. We model the repulsion from the environment as:

$$F_{env} = \frac{k_{env}}{\|p_n^t - p_{obs}\|} \left(\frac{p_n^t - p_{obs}}{\|p_n^t - p_{obs}\|}\right),\tag{3.8}$$

where p_{obs} is the position of the obstacle and k_{env} is a learnable parameter. NSP-SFM learns k_{env} directly via back-propagation and stochastic gradient descent. Since the environment is big, we assume the agent mainly focuses on her view field (Fig. 3.4 b) within which the environment (Fig. 3.4 c) repels the pedestrian. We calculate p_{obs} as the center of the pixels that are classified as obstacles in the view field of an agent. k_{env} is shared among all obstacles. So far, we have introduced all the interpretable parameters $\theta = \{\tau, k_{nj}, k_{env}\}$ in Eq. 3.1.

Dynamics Stochasticity $\alpha(t, q^{t:t-M})$. Trajectory prediction needs to explicitly model the motion randomness caused by intrinsic motion stochasticity and observational noises [212, 228]. We employ a more general setting by assuming the noise

distribution can have arbitrary shapes and is also time varying, unlike previous formulations such as white noise [190] which is too restrictive. Generally, learning such functions requires large amounts of data, as it is unconstrained. To constrain the learning, we further assume the noise is *Normally* distributed in a latent space, rather than in the data space.

Given a prediction \bar{p}^{t+1} without dynamics stochasticity and its corresponding observation p^{t+1} , there is an error $\alpha^{t+1} = \bar{p}^{t+1} - p^{t+1}$. To model the arbitrary and time-varying shape of the distribution of α^{t+1} , we assume it depends on the brief history $p^{t:t-M}$ which implicitly considers the environment and other people. Then the conditional likelihood of α^{t+1} is: $P(\alpha^{t+1}|p^{t:t-M}) = \int P(\alpha^{t+1}|p^{t:t-M}, z)P(z)dz$, where z is a latent variable. Assuming a mapping $Q(z|\alpha^{t+1}, p^{t:t-M})$ and z being Normally distributed, minimizing the KL divergence between Q, *i.e.*, the variational posterior, and $P(z|\alpha^{t+1}, p^{t:t-M})$ leads to a conditional Variational Autoencoder [229].

Our overall loss function is defined as $L = l_{traj} + l_{cvae}$ where:

$$l_{traj} = \frac{1}{N(T-M)} \sum_{n=1}^{N} \sum_{t=M+1}^{T} \|p_n^t - \bar{p}_n^t\|_2^2,$$

$$l_{cvae} = \frac{1}{N(T-M)} \sum_{n=1}^{N} \sum_{t=M+1}^{T} \{\|\alpha_n^t - \tilde{\alpha}_n^t\|_2^2 + \lambda D_{KL}(Q(z|\alpha_n^t, p^{t:t-M}))|P(z|\alpha_n^t, p^{t:t-M}))\}.$$
(3.9)

N is the total number of samples, M is the length of the history, and T is the total length of the trajectory. l_{traj} minimizes the difference between the predicted position and the ground-truth, while l_{cvae} learns the distribution of randomness α . During training, in each iteration, we assume the first M + 1 frames of the trajectory are given and run the forward pass iteratively to predict the rest of the trajectory, then backpropagate to compute the gradient to update all parameters. We employ a progressive training scheme for the sub-nets. We first train Goal-Network with l_{traj} only, then fix Goal-Network and add Collision-Network and F_{env} for training using l_{traj} . Finally, we fix Goal-Network, Collision-Network and F_{env} , add α for training under l_{cvae} . We find this progressive training significantly improves the convergence speed. This is because we first train the deterministic part with the main forces added gradually, which converges quickly. Then the stochasticity part is trained separately to capture complex randomness. Please see the appendix A for implementation details.

3.3.3 NSP vs. Deep Neural Networks

One big difference between NSP and existing deep learning is the deterministic system embedded in NSP. Instead of learning any function mapping the input to the output (as black-box deep learning does), the deterministic system acts as a strong inductive bias and constrains the functional space within which the target mapping should lie. This is because a PDE family can be seen as a flow connecting the input and the output space [230], and the learning is essentially a process of finding the most fitting PDE within this flow. In addition to better data-fitting capability, this strong inductive bias also comes with two other advantages. First, the learned model can help explain motions because the PDE we employ is a physics system where the learnable parameters have physical meanings. Second, after learning, the PDE can be used to predict motions in drastically different scenes (*e.g.*, with higher densities) and generate more plausible trajectories (*e.g.*, fewer collisions). This is difficult for existing deep learning as it requires extrapolating significantly to unseen interactions between pedestrians.

3.4 Experiments

3.4.1 Datasets

We employ six widely used datasets in human trajectory prediction tasks: the Stanford Drone Dataset [21], ETH Hotel, ETH University [203], UCY University, Zara1, and Zara2 datasets [204]. **Stanford Drone Dataset (SDD):** SDD contains videos of a university campus with six classes of agents (pedestrians, cars, *etc.*) with rich interactions. We follow previous research [70, 73, 83] to preprocess the raw data. Specifically, raw videos recorded in FPS (Frames Per Second) = 30 are downsampled to FPS = 2.5. Then, trajectories are extracted from these videos with FPS = 2.5, yielding approximately 10,000 available human trajectories. We split all trajectories into train/test sets in a ratio of approximately 75%/25%. Eventually, we segment trajectories into 20-frame samples to train and test models, where given the first 8 frames (3.2 seconds) models need to predict the remaining 12 frames (4.8 seconds) for each trajectory.

ETH/UCY: The dataset consists of five sub-datasets (ETH Hotel, ETH University, UCY University, Zara1, and Zara2). Every sub-dataset has videos captured in FPS = 25. We also follow previous work [73, 83, 195] to preprocess the raw videos. We down-sample these videos to FPS = 2.5 and then extract human trajectories. ETH/UCY has

a total of about 30,000 human trajectories, where ETH Hotel, ETH University, UCY University, Zara1, and Zara2 approximately contain 1,000, 200, 20,000, 2,000, and 6,000 trajectories, respectively. Following previous research [73, 83, 195], we adopt the standard leave-one-out evaluation protocol, where the model is trained on four sub-datasets and evaluated on the remaining one. Similar to SDD, all human trajectories are segmented into 20-frame samples for training and testing, where the history/prediction split is 8/12 frames (3.2/4.8 seconds). Since our goal sampling network and F_{env} need to work in the pixel space, we project the world coordinates in ETH/UCY into the pixel space using the homography matrices provided in Y-net [83]. When computing the prediction error, we project the predictions in the pixel space back into the world space.

3.4.2 Trajectory Prediction

Average Displacement Error (ADE) and Final Displacement Error (FDE) are employed widely as previous research [15, 26, 73, 83]. ADE is calculated as the l_2 error between a predicted trajectory and the ground truth, averaged over the entire trajectory. FDE is calculated as the l_2 error between the predicted final point and the ground truth. Following prior works, in the presence of multiple possible future predictions, the minimal error is reported. We compare our NSP-SFM with an extensive list of baselines, including published papers and unpublished technical reports: Social GAN (S-GAN) [26], Sophie [70], Conditional Flow VAE (CF-VAE) [93], Conditional Generative Neural System (CGNS) [191], NEXT [192], P2TIRL [193], SimAug [194], PECNet [73], Traj++ [195], Multiverse [196], Y-Net [83], SIT [197], S-CSR [74], Social-DualCVAE [35] and CSCNet [198]. We divide the baselines into two groups due to their setting differences. All baseline methods except S-CSR report the minimal error out of 20 sampled trajectories. S-CSR achieved better results by predicting 20 possible states in each step, and it is the only method adopting such sampling to our best knowledge. We refer to the former as standard-sampling and the latter as ultra-sampling. We compare NSP-SFM with S-CSR and other baseline methods under their respective settings.

Standard-sampling results are shown in Tab. 3.1. On SDD, NSP-SFM outperforms the best baseline Y-Net by 16.94% and 10.46% in ADE and FDE, respectively. In ETH/UCY, the improvement on average is 5.56% and 11.11% in ADE and FDE, with the maximal ADE improvement 12.5% in UNIV and the maximal FDE improvement

Table 3.1: Results on ETH/UCY and SDD based on standard-sampling. NSP-SFM outperforms all baseline methods in both ADE and FDE. 20 samples are used in prediction and the minimal error is reported. M = 7 in all experiments. The unit is meters on ETH/UCY and pixels on SDD.

Methods	Metrics	ETH	Hotel	UNIV	ZARA1	ZARA2	AVG	SDD
	ADE	0.81	0.72	0.60	0.34	0.42	0.58	27.23
S-GAN [26]	FDE	1.52	1.61	1.26	0.69	0.84	1.18	41.44
G 1: [70]	ADE	0.70	0.76	0.54	0.30	0.38	0.54	16.27
Sopnie [70]	FDE	1.43	1.67	1.24	0.63	0.78	1.15	29.38
	ADE	N/A	N/A	N/A	N/A	N/A	N/A	12.60
CF-VAE [93]	FDE	N/A	N/A	N/A	N/A	N/A	N/A	22.30
CONG [101]	ADE	0.62	0.70	0.48	0.32	0.35	0.49	15.6
CGNS [191]	FDE	1.40	0.93	1.22	0.59	0.71	0.97	28.2
NEWD [100]	ADE	0.73	0.30	0.60	0.38	0.31	0.46	N/A
NEXT [192]	FDE	1.65	0.59	1.27	0.81	0.68	1.00	N/A
DOTUDI [100]	ADE	N/A	N/A	N/A	N/A	N/A	N/A	12.58
P2TIRL [193]	FDE	N/A	N/A	N/A	N/A	N/A	N/A	22.07
C: A [104]	ADE	N/A	N/A	N/A	N/A	N/A	N/A	10.27
SimAug [194]	FDE	N/A	N/A	N/A	N/A	N/A	N/A	19.71
DECN ([79]	ADE	0.54	0.18	0.35	0.22	0.17	0.29	9.96
PECNet [73]	FDE	0.87	0.24	0.60	0.39	0.30	0.48	15.88
	ADE	0.39	0.12	0.20	0.15	0.11	0.19	N/A
$1ra_{j}++[195]$	FDE	0.83	0.21	0.44	0.33	0.25	0.41	N/A
Maltinens [100]	ADE	N/A	N/A	N/A	N/A	N/A	N/A	14.78
Multiverse [196]	FDE	N/A	N/A	N/A	N/A	N/A	N/A	27.09
V	ADE	0.28	0.10	0.24	0.17	0.13	0.18	7.85
Y-net [83]	FDE	0.33	0.14	0.41	0.27	0.22	0.27	11.85
	ADE	0.38	0.11	0.20	0.16	0.12	0.19	N/A
511 [197]	FDE	0.88	0.21	0.46	0.37	0.27	0.44	N/A
Social	ADE	0.66	0.34	0.39	0.27	0.24	0.38	N/A
DualCVAE [35]	FDE	1.18	0.61	0.74	0.48	0.42	0.69	N/A
CCCNet [109]	ADE	0.51	0.22	0.36	0.31	0.47	0.37	14.63
USCINET [198]	FDE	1.05	0.42	0.81	0.68	1.02	0.79	26.91
NSP-SFM	ADE	0.25	0.09	0.21	0.16	0.12	0.17	6.52
(Ours)	FDE	0.24	0.13	0.38	0.27	0.20	0.24	10.61

27.27% in ETH. We also compare NSP-SFM with S-CSR in Tab. 3.2. NSP-SFM outperforms S-CSR on ETH/UCY by 70% and 62.5% on average in ADE and FDE. In SDD, the improvement is 35.74% and 0.3% (Tab. 3.2). S-CSR is stochastic and learns perstep distributions, which enables it to draw 20 samples for every step during prediction. Therefore, the min error of S-CSR is much smaller than the other baselines. Similarly, NSP-SFM also learns a per-step distribution (the α function) despite its main behavior being dictated by a deterministic system. Under the same ultra-sampling setting, NSP-SFM outperforms S-CSR.

Table 3.2: Results on ETH/UCY (left) and SDD (right) based on ultra-sampling. 20 samples per step are used for prediction and the overall minimal error is reported. NSP-SFM outperforms S-CSR on both datasets in ADE and FDE.

Methods	Metrics	ETH	Hotel	UNIV	ZARA1	ZARA2	Avg	SDD
	ADE	0.19	0.06	0.13	0.06	0.06	0.10	2.77
5-C5h [14]	FDE	0.35	0.07	0.21	0.07	0.08	0.16	3.45
NSP-SFM	ADE	0.07	0.03	0.03	0.02	0.02	0.03	1.78
	FDE	0.09	0.07	0.04	0.04	0.04	0.06	3.44

3.4.3 Generalization to Unseen Scenarios

We evaluate NSP-SFM on significantly different scenarios after training. We increase the scene density as it is a major factor in pedestrian dynamics [231]. This is through randomly sampling initial and goal positions and letting NSP-SFM simulate the trajectories. Since there is no ground truth, to evaluate the prediction plausibility, we employ collision rate because it is widely adopted [232] and parsimonious: regardless of the specific behaviors of agents, they do not penetrate each other in the real world. The collision rate is computed based on the percentage of trajectories colliding with one another. We treat each agent as a disc with the radius r = 0.2 m in ETH/UCY and r = 7.5 pixels in SDD. Once the distance between two agents falls below 2r, we count the two trajectories as in collision. Due to the tracking error and the distorted images, the ground truth r is hard to obtain. We need to estimate r. If it is too large, the collision rate will be high in all cases; otherwise, the collision rate will be too low, e.g., r = 0 will give 0% collision rate all the time. Therefore, we did a search and found that the above values are reasonable as they keep the collision rate of the ground-truth data approximately zero. All data are captured from a bird's eve view or a very similar perspective. Therefore, the radius of a human disk doesn't change significantly as the distance between the person and the camera. This means that it is reasonable to use a fixed radius for a dataset. We show two experiments. The first is the collision rate on the testing data, and the second is scenarios with higher densities. While the first is mainly to compare the plausibility of the prediction, the second is to test the model's generalizability. For comparison, we choose two state-of-the-art baseline methods: Ynet and S-CSR. Y-net is published which achieves the best performance, while S-CSR is unpublished but claims to achieve better performance.

Methods	ETH	Hotel	UNIV	ZARA1	ZARA2	Avg	SDD
Y-net	0	0	1.51%	0.82%	1.31%	0.73%	0.47%
S-CSR	0	0	1.82%	0.41%	1.31%	0.71%	0.42%
NSP-SFM	0	0	1.48%	0	0.66%	0.43%	0.42%

Table 3.3: Collision rates on testing data in ETH/UCY and SDD. NSP-SFM universally outperforms all baseline methods.

Table 3.4: Collision rates of the generalization experiments on ZARA2 (Z) and coupa0 (C). NSP-SFM shows strong generalizability in unseen high-density scenarios.

Methods	Z(1)	Z(2)	Z(3)	Z(avg)	C(1)	C(2)	C(3)	C(avg)
Y-net	1.8%	2.2%	2.0%	2.0%	2.8%	2.9%	3.8%	3.2%
S-CSR	3.2%	2.4%	1.8%	2.5%	2.5%	1.7%	1.9%	2.0%
NSP-SFM	0.2%	0.2%	0	0.1%	0.6%	0.6%	0.6%	0.6%

Tab. 3.3 shows the comparison of the collision rate. NSP-SFM outperforms the baseline methods in generating trajectories with fewer collisions. Y-net and S-CSR also perform well on the testing data because their predictions are close to the ground-truth. Nevertheless, they are still worse than NSP-SFM. Next, we test drastically different scenarios. We use ZARA2 and coupa0 (a sub-dataset from SDD) as the environment and randomly sample the initial positions and goals for 32 and 50 agents respectively. Because the highest number of people that simultaneously appear in the scene is 14 in ZARA2 and 11 in coupa0, we effectively increase the density by 2-5 times. For NSP-SFM, the initial and goal positions are sufficient. For Y-net and S-CSR which require 8 frames (3.2 seconds) as input, we use NSP-SFM to simulate the first 8 frames of each agent, then feed them into both baselines. Tab. 3.4 shows the results of three experiments. Since the density is significantly higher than the data, both Y-net and S-CSR cause much higher collision rate. While NSP-SFM's collision rate also occasionally increases (*i.e.* SDD) compared with Tab. 3.3, it is far more plausible. More results and details are in the appendix A.



Figure 3.5: Interpretability of prediction. Red dots are observed, green dots are our prediction and black dots are the ground-truth. The blue dots are neighbors. F_{goal} , F_{col} , and F_{env} are shown as yellow, light blue, and black arrows for a person. The orange areas are the view field for avoiding collisions with other people (left) and the environment (middle). They provide plausible explanations of individual behaviors such as steering. The left and middle show the major influence of different forces. Right shows motion randomness captured by our model.

3.4.4 Interpretability of Prediction

Unlike black-box deep learning methods, NSP-SFM has an embedded explainable model. While predicting a trajectory, NSP can also provide plausible explanations of the movement, by estimating the 'forces' exerted on a specific person. This potentially enables NSP-SFM to be used in applications beyond prediction, *e.g.* behavior analysis [233]. Fig. 3.5 Left shows that a person, instead of directly walking towards the goal, steered upwards (the green trajectory in the orange area). This could be explained by the strong repulsive force (the light blue arrow) which is generated by the potential collisions with the agents in front of this person, in line with existing studies [231]. Similar explanations can be made in Fig. 3.5 Middle, where all three forces are present. F_{env} (the black arrow) is the most prominent, as expected, as the person is very close to the car. The repulsive force (light blue arrow) also plays a role due to the person in front of the agent (the blue dot in the orange area).

Fig. 3.5 Right shows an example where movement randomness is captured by NSP-SFM. In this example, there is no other pedestrian and the person is not close to any obstacle. However, the trajectory still significantly deviates from a straight line, which cannot be fully explained by e.g. the principle of minimal energy expenditure [234]. The deviation could be caused by unobserved factors, e.g. the agent changing her goal

SDD	$F_{goal}(w/o)$	NSP-SFM(w/o)	NSP-SFM(w)
ADE	6.57	6.52	1.78
FDE	10.68	10.61	3.44

Table 3.5: Ablation study on SDD. (w/o) means without CVAE and (w) means with CVAE. F_{goal} is goal attraction only and NSP-SFM is all three forces.

or being distracted by something on the side. These factors do not only affect the trajectory but also the dynamics, *e.g.* sudden changes in velocity. These unobserved random factors are implicitly captured by the CVAE in NSP-SFM. More results are in the appendix A.

We emphasize that NSP-SFM merely provides plausible explanations and by no means the only possible explanations. Although explaining behaviors based on physics models has been widely used, there can be alternative explanations [235]. Visualizing the forces is merely one possible way. Theoretically, it is also possible to visualize deep neural networks, *e.g.* layer activation. However, it is unclear how or which layer to visualize to explain the motion. Overall, NSP-SFM is more explainable than black-box deep learning.

3.4.5 Ablation Study

To further investigate the roles of different components, we conduct an ablation study on SDD with three settings: $F_{goal}(w/o)$ with goal attraction only, *i.e.* omitting other components such as F_{col} , F_{env} and dynamics stochasticity; NSP-SFM (w/o) without dynamics stochasticity; and NSP-SFM (w) the full model. The results are shown in Tab. 3.5. Interestingly, $F_{goal}(w/o)$ can already achieve good results. This is understandable as it is trained first in our progressive training scheme and catches most of the dynamics. NSP-SFM (w/o) further improves the performance. The improvement seems to be small but we find the other repulsive forces are crucial for trajectories with irregular geometries such as avoiding obstacles. Further NSP-SFM (w) significantly improves the results because it enables NSP-SFM to learn the dynamics stochasticity via a per-step distribution. We show one example in Fig. 3.6 in all settings. More ablation experiments can be found in the appendix A.



Figure 3.6: Examples for ablation study. Red, green, and cyan dots are observations, prediction, and ground-truth respectively. From left to right: ground truth, $F_{goal}(w/o)$, NSP-SFM(w/o) and NSP-SFM(w).

3.5 Conclusions, Limitations, and Future Work

In this Chapter, we have proposed a new neural differential equation framework and a novel model under this framework for trajectory prediction. Through exhaustive evaluation and comparison, our model, NSP-SFM, has proven to be more accurate in trajectory prediction, generalize well in significantly different scenarios, and can provide possible explanations for predictions. The major limitation of NSP lies in the physics model, which only focuses on common low-density crowds. In addition, our method requires data to be captured under a bird's eye view or a very similar perspective. Although our method can mitigate the limitation of the capture view by learning dynamics from data, it might be useful for our method to consider camera parameters. In the future, we would like to extend the current model considering parameters related to social forces by incorporating camera parameters. Additionally, we can extend the current framework to model high-density crowds, where continuum models or reciprocal velocity obstacles need to be used to replace the components in Eq. 3.4. We would also like to incorporate learning-based collision detection techniques into this framework [236, 237].

Chapter 4

Uncertainty-aware Human Trajectory Prediction This Chapter focuses on uncertainty modeling in human trajectory prediction. Similar to Chapter 3, individuals are assumed to be 2D points and we use a series of 2D points to represent their trajectories. Human trajectory prediction is a crucial task in understanding human behaviors. The random nature of human movements, from lowerlevel steering to high-level affective states, dictates that uncertainty estimation is vital in prediction, which consists of the data and model uncertainty. However, existing deep learning methods either only capture the data uncertainty or mix the data and model uncertainty. To model these two uncertainties explicitly and investigate their relative importance in prediction, we propose a novel Bayesian stochastic social force model based on deep learning that decouples them. Our model captures the data uncertainty by a new explicit Bayesian neural stochastic differential equation model, and the model uncertainty by an implicit model. Additionally, Bayesian inference is employed to incorporate prior knowledge of human behaviors. Given that there is no clear separation between the two uncertainties in the data, we design a proper training scheme to enable both uncertainties to capture the ideal randomness. Through comprehensive evaluation, our model demonstrates strong explainability with fine-grained uncertainty modeling. In addition, it also outperforms existing methods in prediction accuracy by up to 60.17% on public datasets SDD and ETH/UCY.

4.1 Introduction

Human trajectory forecasting has wide-ranging applications, *e.g.* social robots and selfdriving vehicles [238, 239], and therefore is studied in fields such as computer science, physics, mathematics, robotics, and transportation [9]. Early methods tend to be based on deterministic prediction [15, 51, 240], which has been challenged by the observations of intrinsic stochasticity of trajectory data [26, 241]. Subsequently, stochastic prediction has become the mainstream, especially in recent deep learning work [26, 38, 70, 73, 74, 79, 80, 83, 89, 242].

Among stochastic prediction methods, explainability is a delimiter, where the stochasticity can be captured implicitly, and hence unexplainable [26, 75, 80], or explicitly and explainable [38, 83]. Recent research argues that the latter is important, especially in safety-critical applications such as autonomous vehicles, as it gives humanunderstood reasons for model predictions [38]. However, we argue that existing uncertainty modeling is coarse and needs to be refined. In human trajectories, the uncertainty can be divided into the *data* and *model* uncertainty [243]. The former is caused by intrinsic behavioral randomness such as affective states, while the latter is due to factors that are not considered in the model or not observed in data, such as random events that attract people's attention [38, 244]. However, existing methods with uncertainty modeling either only consider the data uncertainty or mix the two uncertainties together. Only modeling the data uncertainty essentially assumes the model comprehensively considers all possible factors, which is often not true. On the other hand, mixing the two uncertainties leads to ambiguity, *e.g.* which part of the motion randomness could be explained by random behaviors and which part by unknown factors.

To this end, we propose a new Bayesian stochastic social force model (BSSFM) with fine-grained uncertainty modeling for human trajectory prediction. Inspired by recent success in neural differential equation models [36, 37, 201, 202], we model the data uncertainty by a novel Bayesian neural stochastic differential equation (SDE) model based on the social force model (SFM) [11], where pedestrians are seen as a particle system. The forces on particles are stochastic, influenced by their goals and various interactions. The key parameters of the distributions of the forces are learned by deep neural networks. Moreover, we use Bayesian inference to incorporate prior knowledge of human trajectories when training the networks. This Bayesian neural SDE model enables explicit and explainable uncertainty estimation. Further, to accommodate the model uncertainty, we use a deep generative model (DGM). Unlike the data uncertainty, the model uncertainty can only be attributed to unknown factors and therefore needs to be modeled implicitly. Finally, we propose an uncertainty-aware training scheme that decouples the data uncertainty and the model uncertainty. The particle system, despite being stochastic, generates an expected prediction, and we regard its residue from the ground truth as caused by the model uncertainty. Learning two uncertainties simultaneously would lead to competition between them. Therefore, our training scheme ensures that the data uncertainty component maximizes its explainability and leaves the remaining to the model uncertainty.

Our model not only provides predictions but also plausible explanations of predictions. This explainability benefits from the separation of data and model uncertainty distributions. On top of uncertainty, our model also achieves state-of-the-art performance in various trajectory prediction tasks across multiple datasets. Additionally, our model exhibits superior generalizability to unseen scenarios compared to existing methods. Formally, our contributions are threefold:

- A novel Bayesian stochastic social force model with fine-grained uncertainty of human trajectories.
- A new way to integrate stochastic differential equations with neural networks for human trajectory prediction.
- A demonstration of the effectiveness of our model across a broad spectrum of tasks, including behavior explanation, trajectory forecasting, generalization to unseen scenarios, and data efficiency.

4.2 Related Work

Human trajectory forecasting is a task to predict future movements based on historical trajectories and the environment. Early attempts focused on deterministic models [11, 15, 51, 240]. The social force model [11] treats pedestrians as a particle system and designs social forces to generate future trajectories via Newton's laws of motion, where parameters of social forces are hand-picked. Inspired by the success of recurrent neural networks, [15] employs a Long Short-Term Memory (LSTM) for each person to learn and predict general patterns of human movement. [61] proposes a soft attention model to extract the interaction information between people and use it to predict future trajectories.

Deterministic models have been challenged because of the observation that human trajectories are random and multi-modal. Therefore, stochastic trajectory prediction arose to capture the multimodality in human trajectories. Methods based on deep learning dominate stochastic trajectory prediction with their excellent performance. Existing methods fall into one of two categories based on the explainability of the uncertainty: Explainable and unexplainable. The explainable methods tend to model the stochasticity explicitly. For example, several methods [66, 87, 245] employ the Gaussian distribution or the Gaussian Mixture Model to learn the data uncertainty. SIT [85] explores the application of the tree structure in human trajectory prediction to address the multimodality of human trajectories. TUTR [79] extracts several general motion modes from training data to explain the uncertainty. A different line of work models the stochasticity implicitly, ignoring the explainability. For instance, some approaches [89, 244] use Bayesian neural networks [90] to capture uncertainties in human

trajectories. A more popular trend is to employ deep generative models with a latent space/noise space to acquire the uncertainty implicitly. A series of methods [26, 70, 84] employs generative adversarial networks (GANs) while a lot of methods [74, 80, 93, 242] are based on variational autoencoders (VAEs). Recently, some works [75, 246] based on the diffusion model have arisen to exploit its strong generation ability. Moreover, from the view of uncertainty modeling, all existing methods are coarse-grained because they either only consider the data uncertainty or blend the data and model uncertainty. Our model focuses on stochastic trajectory prediction with explainable uncertainty. In contrast to existing methods based on deep learning, our model with fine-grained uncertainty, for the first time and to our knowledge, decouples two uncertainties and captures them individually. As a result, our model possesses better explainability than previous work while improving the prediction performance slightly.

4.3 Methodology

4.3.1 Motivation

Supposing that $\mathbf{c}_i^t \in \mathbb{R}^2$ is the 2D coordinate of the *i*th pedestrian at timestep *t*, the task requires predicting future trajectories $\mathcal{T}_{i=1:N}^f = {\mathbf{c}_i^t}_{i=1:N}^{t=t_p+1:t_p+t_f}$ for the next t_f timesteps, given the past trajectories $\mathcal{T}_{i=1:N}^p = {\mathbf{c}_i^t}_{i=1:N}^{t=1:t_p}$ for the past t_p timesteps and the environment, where *N* denotes the number of people. Our model predicts stochastic future trajectories by capturing the data uncertainty explicitly and the model uncertainty implicitly. Inspired by the traditional SFM, we capture the data uncertainty by a particle system, where stochastic forces on particles are estimated. Formally, we denote a trajectory as a function $\mathbf{c}(t)$ and model the data uncertainty via a second-order neural stochastic differential equation:

$$\begin{cases} d\boldsymbol{c}(t) = \dot{\boldsymbol{c}}(t)dt, \\ d\dot{\boldsymbol{c}}(t) = \boldsymbol{\mu}(t)dt + \boldsymbol{\sigma}(t)d\boldsymbol{W}(t), \\ \boldsymbol{c}(0) = \mathbf{c}^{0}, \dot{\boldsymbol{c}}(0) = \dot{\mathbf{c}}^{0}, \boldsymbol{c}(t_{p} + t_{f}) = \mathbf{c}^{t_{p} + t_{f}}, \end{cases}$$
(4.1)

where \mathbf{W} is a Wiener process and functions $\boldsymbol{\mu}(t)$ and $\boldsymbol{\sigma}(t)$ are parameterized by neural networks, given the boundary condition of the initial position \mathbf{c}^0 , the initial velocity $\dot{\mathbf{c}}^0$, and the destination $\mathbf{c}^{t_p+t_f}$. Properties of the Wiener process point out that $\Delta \mathbf{W} =$ $\mathbf{W}^{t+\Delta t} - \mathbf{W}^t \sim \mathcal{N}(0, \Delta t)$. We assume that the distribution of the force at any timestep t is a Gaussian distribution because human movement tends to have a main direction and the Gaussian distribution can describe this trend well. Note that modeling with forces enables our model to possess explainability naturally. In addition, we can get the following equation from Eq. 4.1:

$$\boldsymbol{c}^{t_p+t_f} - \boldsymbol{c}^0 = \int_{t=0}^{t_p+t_f} \dot{\boldsymbol{c}}(t) \mathrm{d}t = \int_{t=0}^{t_p+t_f} (\int \boldsymbol{\mu}(t) \mathrm{d}t + \int \boldsymbol{\sigma}(t) \mathrm{d}\boldsymbol{W}(t)) \mathrm{d}t.$$
(4.2)

To fit our model to discrete observations in time, we introduce the time-discretized version for Eq. 4.1 on a fixed time interval Δt :

$$\begin{cases} \mathbf{c}^{t+\Delta t} - \mathbf{c}^{t} = \dot{\mathbf{c}}^{t+\Delta t} \Delta t, \\ \dot{\mathbf{c}}^{t+\Delta t} - \dot{\mathbf{c}}^{t} = \mu^{t} \Delta t + \sigma^{t} \Delta \mathbf{W}, \\ \mathbf{c}(0) = \mathbf{c}^{0}, \dot{\mathbf{c}}(0) = \dot{\mathbf{c}}^{0}, \mathbf{c}(t_{p} + t_{f}) = \mathbf{c}^{t_{p}+t_{f}}. \end{cases}$$
(4.3)

Then, we estimate μ^t and σ^t in Eq. 4.3 using neural networks. Further, we have the forward estimation process that captures the data uncertainty:

$$\mathbf{c}^{t+\Delta t} = \mathbf{c}^t + \dot{\mathbf{c}}^t \Delta t + \mu^t \Delta t^2 + \sigma^t \Delta \mathbf{W} \Delta t = \mathbf{c}^t + \dot{\mathbf{c}}^t \Delta t + (\mu^t + \sigma^t \frac{\Delta \mathbf{W}}{\Delta t}) \Delta t^2.$$
(4.4)

The prior knowledge of human behaviors is introduced to train the neural networks in the SDE via Bayesian inference. In addition, motivated by previous work [26, 75, 80], we use a deep generative model to capture the model uncertainty implicitly. Lastly, we design a proper uncertainty-aware training scheme to ensure that the data and model uncertainty are captured accurately by corresponding models.

4.3.2 Bayesian Stochastic Social Force Model

Similar to [11, 80], we divide the force in Eq. 4.4 into three factors: the actuation factor \mathbf{D}_a^t from goals, the neighbor factor \mathbf{D}_n^t from neighbors to model the interaction between people, and the scene factor \mathbf{D}_s^t from the scene context to avoid obstacles:

$$\mu^t + \sigma^t \frac{\Delta \mathbf{W}}{\Delta t} = \mathbf{D}_a^t + \mathbf{D}_n^t + \mathbf{D}_s^t, \tag{4.5}$$

where \mathbf{D}_{a}^{t} and \mathbf{D}_{n}^{t} are time-varying Gaussians while \mathbf{D}_{s}^{t} is a static Gaussian. Our factors are stochastic and learnable, unlike previous approaches [11, 80]. We use the noise term ε to denote the captured model uncertainty. Substituting Eq. 4.5 into Eq. 4.4, we obtain the final forward estimation equation:

$$\mathbf{c}^{t+\Delta t} = \mathbf{c}^t + \dot{\mathbf{c}}^t \Delta t + (\mathbf{F}_a^t + \mathbf{F}_n^t + \mathbf{F}_s^t) \Delta t^2 + \varepsilon^t,$$

$$\mathbf{F}_a^t \sim \mathbf{D}_a^t, \ \mathbf{F}_n^t \sim \mathbf{D}_n^t, \ \mathbf{F}_s^t \sim \mathbf{D}_s^t.$$
 (4.6)



Figure 4.1: Overview of BSSFM. At each timestep, distributions \mathbf{D}_a and \mathbf{D}_n are estimated by the actuation block and the neighbor block. \mathbf{D}_s models the force from the scene and is a Gaussian distribution with learnable parameters. Then, we sample \mathbf{F}_a , \mathbf{F}_n , \mathbf{F}_s from their corresponding predicted distributions. The model uncertainty term ε is sampled from a DGM. Finally, our model predicts future trajectories via Eq. 4.6.

The overview of our BSSFM is shown in Fig. 4.1. The explicit goal distribution is predicted by the goal sampling network in [80] first. Then, given the past trajectories and sampled goals, our model predicts distributions \mathbf{D}_a^t and \mathbf{D}_n^t via the actuation block and the neighbor block and estimates the distribution \mathbf{D}_s^t from the scene context at any timestep t. Then we sample forces \mathbf{F}_a^t , \mathbf{F}_n^t , and \mathbf{F}_s^t from their corresponding predicted distributions. Simultaneously, the model uncertainty ε^t is generated by a deep generative model. Eventually, we can predict future trajectories by solving Eq. 4.6 iteratively.

Actuation Block. Following [80], we use the goal sampling network to predict the explicit goal distribution given the scene image and the past trajectory. The goal distribution is a non-parametric probability distribution map with shape [h,w,1], where h and w are the height and width of the input scene image. Each pixel value on this map is the probability that the pixel is sampled as a goal. More details including sampling can be found in [80, 83]. Given a sampled goal $\mathbf{c}^{t_p+t_f}$, the current position \mathbf{c}^t , and the current velocity $\dot{\mathbf{c}}^t$, we define the attraction $\mathbf{F}_a^t \sim \mathbf{D}_a^t$ as:

$$\mathbf{F}_{a}^{t} = \left(\frac{\mathbf{c}^{t_{p}+t_{f}} - \mathbf{c}^{t}}{(t_{p}+t_{f}-t)\Delta t} - \dot{\mathbf{c}}^{t}\right)k_{a}^{t}, \quad k_{a}^{t} \sim \mathcal{N}_{\phi_{1}}(\mu_{a}^{t}, \sigma_{a}^{t\,2}), \tag{4.7}$$

where $\left(\frac{\mathbf{c}^{t_p+t_f}-\mathbf{c}^t}{(t_p+t_f-t)\Delta t}-\dot{\mathbf{c}}^t\right)$ is the expected velocity correction on the current velocity $\dot{\mathbf{c}}^t$ towards $\mathbf{c}^{t_p+t_f}$. μ_a^t and σ_a^t are the mean and standard deviation of the Gaussian distribution at timestep t, estimated by neural networks with parameters ϕ_1 :

$$[\mu_a^t, \sigma_a^t]_{\phi_1} = D_a \left\{ \text{Concat} \left[E_a(\mathbf{c}^{t_p + t_f}), \, \text{LSTM}(\mathbf{c}^t, \mathbf{\dot{c}}^t) \right] \right\},\tag{4.8}$$

where LSTM denotes the Long Short-Term Memory network [247], Concat means concatenation, and both E_a and D_a are multi-layer perceptrons (MLPs). The network architecture is shown in the Fig. 4.1 actuation block, and details are in the appendix B.

Neighbour Block. Given the *i*th pedestrian \mathbf{c}_i^t at the timestep t and his/her set of neighbors Ω_i^t , we define the distribution of the neighbor force as follows:

$$\mathbf{D}_{n,i}^{t} = \sum_{j=0}^{m} \mathbf{D}_{n,ij}^{t}, \text{ where } j \in \Omega_{i}^{t}, \ \mathbf{F}_{n,ij}^{t} \sim \mathbf{D}_{n,ij}^{t},
\mathbf{F}_{n,ij}^{t} = -\nabla_{\mathbf{r}_{ij}^{t}} r_{n} e^{-\|\mathbf{r}_{ij}^{t}\|/r_{n}} k_{n,ij}^{t},
\mathbf{r}_{ij} = \mathbf{c}_{i}^{t} - \mathbf{c}_{j}^{t}, \ k_{n,ij}^{t} \sim \mathcal{N}_{\phi_{2}}(\mu_{n,ij}^{t}, \sigma_{n,ij}^{t\,2}),$$
(4.9)

where $\mu_{n,ij}^t$ and $\sigma_{n,ij}^t$ are the mean and standard deviation of the Gaussian distribution. Similar to [11], $r_n e^{-\|\mathbf{r}_{ij}\|/r_n}$ is a repulsive potential energy function with a radius r_n (hyperparameter), and the negative gradient w.r.t. r_{ij} gives a repulsive force. However, such repulsion between pedestrians has randomness [190, 228]. Therefore, our model considers a time-varying Gaussian with a learnable mean and variance, so the distribution of the total neighbour force $\mathbf{D}_{n,i}^t$ is a time-varying Gaussian mixture. We predict $\mu_{n,ij}$ and $\sigma_{n,ij}$ using neural networks parameterized by ϕ_2 :

$$[\mu_{n,ij}^t, \sigma_{n,ij}^t]_{\phi_2} = D_n \left\{ \text{Concat} \left[E_n(\mathbf{c}_j^t, \dot{\mathbf{c}}_j^t), \, \text{LSTM}(\mathbf{c}_i^t, \dot{\mathbf{c}}_i^t) \right] \right\}.$$
(4.10)

The network architecture is shown in the Fig. 4.1 neighbor block, and details are in the appendix B.

Scene Context. To model how people avoid obstacles in the scene, given the



Figure 4.2: The CVAE architecture in BSSFM. H denotes the number of history frames.

pedestrian \mathbf{c}^t and the scene \mathbf{S} , we define the scene force as:

$$\mathbf{D}_{s}^{t} = \sum_{obst \in \mathbf{S}} \mathbf{D}_{obst}^{t}, \ \mathbf{F}_{obst}^{t} \sim \mathbf{D}_{obst}^{t},$$

$$\mathbf{F}_{obst}^{t} = \left(\frac{\mathbf{c}^{t} - \mathbf{c}_{obst}}{\|\mathbf{c}^{t} - \mathbf{c}_{obst}\|_{2}^{2}}\right) k_{s}, \ k_{s} \sim \mathcal{N}(\mu_{s}, \sigma_{s}^{2}),$$
(4.11)

where *obst* represents the obstacle in the scene and μ_s and σ_s are the mean and standard deviation. Different from Eq. 4.7 and Eq. 4.9, k_s is assumed to be time-independent and agent-independent. This is because we observe similar influences of obstacles on different pedestrians. Therefore, unlike \mathbf{D}_a and \mathbf{D}_n , we directly learn μ_s and σ_s .

Model Uncertainty. Deep generative models demonstrate their ability to capture the model uncertainty in human trajectory prediction [26, 73, 80]. In particular, the conditional variational autoencoder (CVAE) performs well in capturing the model uncertainty in methods based on the particle system [80]. Therefore, a CVAE is used to generate the ε term in Eq. 4.6. To be specific, we train the CVAE to reconstruct the residues $\mathbf{r}_i^t = \mathbf{c}_i^t - \hat{\mathbf{c}}_i^t$, where \mathbf{c}_i^t is the ground truth and $\hat{\mathbf{c}}_i^t$ is the prediction of the trained particle system. The architecture of the CVAE is shown in Fig. 4.2. Networks E_{res} and E_{past} encode the residue \mathbf{r}_i^t and the history condition $(\mathbf{c}_i^{t-H:t-1}, \mathbf{\hat{c}}_i^t)$ to obtain features $\mathbf{f_{res}}$ and $\mathbf{f_{past}},$ respectively, where H is the number of history frames. We concatenate features \mathbf{f}_{res} and \mathbf{f}_{past} and feed them into the encoder E_{latent} to yield means and variances of the Gaussian distribution of the latent variable z. Next, we obtain the input of the decoder D_{latent} by concatenating a sampled z and the feature \mathbf{f}_{past} . Finally, the decoder D_{latent} outputs the estimation of the residue. Red connections in Fig. 4.2 are only used when training. This is because the ground truth \mathbf{r}_i^t is unavailable during testing. To reconstruct the residue when testing, the latent variable z is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_{latent}\mathbf{I})$ with a hyperparameter σ_{latent} , while the feature $\mathbf{f_{past}}$ is extracted from the history condition by the trained encoder E_{past} . Then we concatenate the sampled z and the $\mathbf{f_{past}}$ to obtain the input of the trained decoder D_{latent} . Eventually, D_{latent} offers the estimation of the residue. The E_{res} , E_{past} , E_{laten} and D_{latent} are MLPs, and more details can be found in the appendix **B**.

4.3.3 Training Scheme and Loss Function

Although separating data and model uncertainty improves explainability as shown later, there is a competition between them. This is because they can be seen as correction forces on predictions, but it is not straightforward to decide who should do more corrections. Following our modeling philosophy in maximizing the explainability, *i.e.* the data uncertainty, we train our model with an uncertainty-aware scheme to decouple the data uncertainty and the model uncertainty. First, the particle system is trained to maximize the capture of the data uncertainty, where the prior knowledge of human trajectories is incorporated by the Bayesian inference. Then, to avoid competition between two uncertainties, we use the residues of the trained particle system from the ground truth to train the CVAE and obtain the model uncertainty.

We train the particle system and the CVAE by using loss functions \mathcal{L}_{Bayes} and \mathcal{L}_{cvae} , respectively. Explainable parameters $\eta = \{k_a, k_{n,ij}, k_s\}$ control the particle system. Therefore, we aim to obtain the posterior $p(\eta|\mathcal{T}_o) \propto p(\mathcal{T}_o|\eta)p(\eta)$ to predict future trajectories, where \mathcal{T}_o denotes all observed trajectories in the training dataset. However, the integrals on $p(\mathcal{T}_o|\eta)$ and $p(\eta)$ tend to be intractable. So, we make Bayesian inference [248] based on Eq. 4.6 to optimize a variational posterior $q_{\phi}(\eta)$ to approximate the true posterior $p(\eta|\mathcal{T}_o)$. $\phi = \{\phi_1, \phi_2, \mu_e, \sigma_e\}$ is the set of learnable parameters in the particle system. Specifically, we acquire the loss \mathcal{L}_{Bayes} by calculating the KL divergence between the variational posterior $q_{\phi}(\eta)$ and the true posterior $p(\eta|\mathcal{T}_o)$:

$$\mathcal{L}_{Bayes} = D_{\mathbb{KL}}(q_{\phi}(\eta) \| p(\eta | \mathcal{T}_{o}))$$

$$= \mathbb{E}_{q_{\phi}(\eta)} \left[\log q_{\phi}(\eta) - \log \left(\frac{p(\mathcal{T}_{o} | \eta) p(\eta)}{p(\mathcal{T}_{o})} \right) \right]$$

$$= \mathbb{E}_{q_{\phi}(\eta)} \left[\log q_{\phi}(\eta) - \log p(\mathcal{T}_{o} | \eta) p(\eta) + \log p(\mathcal{T}_{o}) \right]$$

$$= \mathbb{E}_{q_{\phi}(\eta)} \left[\log q_{\phi}(\eta) - \log p(\mathcal{T}_{o} | \eta) p(\eta) \right], \qquad (4.12)$$

where $\log p(\mathcal{T}_o)$ is a constant. In the implementation, we assume that both $q_{\phi}(\eta)$ and

 $p(\eta)$ are diagonal Gaussian distributions. Then we can get:

$$\log q_{\phi}(\eta) = \sum_{k \in \eta} -\frac{(k - \mu_{\phi})^2}{2\sigma_{\phi}^2} - \log \sigma_{\phi} - \log \sqrt{2\pi},$$

$$\log p(\eta) = \sum_{k \in \eta} -\frac{(k - \mu_{prior})^2}{2\sigma_{prior}^2} - \log \sigma_{prior} - \log \sqrt{2\pi},$$
(4.13)

where μ_{ϕ} and σ_{ϕ}^2 are the means and variances predicted by our method with parameters ϕ for each $k \in \eta$. We also assume that the likelihood $p(\mathcal{T}_o|\eta)$ is a diagonal Gaussian:

$$\log p(\mathcal{T}_{o}|\eta) = \sum_{\mathcal{T}^{f} \in \mathcal{T}_{o}} -\frac{1}{2} \|\mathcal{T}^{f} - \widehat{\mathcal{T}}^{f}\|_{2}^{2} - \frac{t_{f}}{2} \log 2\pi,$$
(4.14)

where \mathfrak{T}^f is the ground-truth future trajectory, and $\widehat{\mathfrak{T}}^f$ is predicted by the particle system in our model given η and the past trajectory \mathfrak{T}^p . Theoretically, our method can incorporate various informative prior knowledge in the data uncertainty component by introducing different explicit models. For our model considering the particle system, we train the NSP [80] to obtain the prior distributions in Eq. 4.13.

After training the particle system, we train the CVAE by using a standard CVAE loss:

$$\mathcal{L}_{cvae} = \frac{1}{Nt_f} \sum_{i=1}^{N} \sum_{t=t_p+1}^{t_p+t_f} \{ \|\mathbf{r}_i^t - \tilde{\mathbf{r}}_i^t\|_2^2 + \lambda D_{\mathbb{KL}}(q(\mathbf{z}_i^t|\mathbf{x})\|\mathcal{N}(0,\mathbf{I})) \},$$
(4.15)

where N is the total number of data samples, λ is a tradeoff hyperparameter, \mathbf{r}_i^t is the residue between ground truth and the prediction from the particle system, $\tilde{\mathbf{r}}_i^t$ is the reconstructed residue, \mathbf{z}_i^t is the latent variable, and x denotes the input of the CVAE. Our overall loss function is $\mathcal{L} = \mathcal{L}_{Bayes} + \mathcal{L}_{cvae}$.

4.4 Experiments

4.4.1 Datasets

We evaluate our model and baselines using two popular public datasets for human trajectory forecasting: the Stanford Drone Dataset (SDD) [21] and ETH/UCY [203, 204]. **SDD** consists of videos across several distinct scenes. These videos are recorded from a drone flying over Stanford University, providing a bird's-eye view of human movement. There are more than 10,000 unique pedestrians with rich interactions in



Ours with Data Uncertainty NSP with Mixed Uncertainties Ours with Decoupled Uncertainties

Figure 4.3: Visualization of uncertainties. The probability distribution of prediction at each timestep is shown as heatmaps, where yellow to blue indicates the probability density from high to low.

the dataset. The dataset also contains abundant pedestrian-environment interactions due to the irregular shapes of walkable regions. In line with prior studies [73, 79, 80, 83], we extract trajectories at 0.4-second intervals, generating 20-frame samples in an 8/12 setting. This means we use the initial 8 frames to predict the next 12 frames. More details like the size of the dataset can be found in section 3.4.1. **ETH/UCY** consists of five sub-datasets (ETH, HOTEL, UNIV, ZARA1, and ZARA2), capturing a variety of human behaviors and providing world coordinates for pedestrians. We convert the world coordinates into pixel coordinates by using homography matrices since our model works in the pixel space, and project the predictions back into the world space when calculating errors for fair performance comparison. These five datasets are typically used for evaluation with a leave-one-out protocol [26, 73, 80, 83], which we also employ. In this protocol, the model is trained on four sub-sets and tested on the remaining one, rotating through all subsets. Additionally, the same extraction and processing with SDD are utilized in ETH/UCY. More details such as the size of the dataset are in section 3.4.1.

4.4.2 Fine-grained Uncertainty and Explainability

Data uncertainty can be explainable while it is hard to explain model uncertainty. However, existing methods, even when considering both, learn them as a whole, which leads to ambiguity of uncertainty sources, and competition between them when fitting data. In contrast, our model and training scheme separate them and maximize the data uncertainty. Note that the data uncertainty maximisation is crucial because it is represented by the particle forces, where their impact on motions is well studied in



Figure 4.4: Visualization comparison. Figures for NSP are from [80]. Red, blue, and green dots denote observations, neighbors, and predictions, respectively. Forces \mathbf{F}_a , \mathbf{F}_n , and \mathbf{F}_s are shown as yellow, light blue, and black arrows, respectively. The orange areas are two view fields. The explicit distributions of stochastic forces predicted by our model are shown as heatmaps. Yellow to blue in the heatmaps indicates that the probability density is from high to low.

particle systems. Therefore, the data uncertainty can be easily understood by humans. On the other hand, the model uncertainty is hard to explain because it is caused by unknown factors. Eventually, the fine-grained uncertainty has better explainability than the coarse one employed in existing methods.

In Fig. 4.3, we show our data uncertainty (Fig. 4.3 left), our total uncertainty (Fig. 4.3 right), and the NSP mixed uncertainty (Fig. 4.3 middle). Our total uncertainty is similar to the NSP uncertainty. This is understandable because both methods employ a particle system as the underlying governing model. However, the key difference is that our model has the capability of separately learning the data uncertainty in Fig. 4.3 left, which reveals the intrinsic behavioral randomness in steering caused by factors such as actuation and neighbors. Admittedly, the very definition of data uncertainty and model uncertainty depends on the specification of the underlying model. However, we argue that the explicit separation of them in learning enables better visualization and analysis of the potential reasons for predictions.

Next, we show that data uncertainty can be further decomposed for a more detailed analysis in explaining the predictions. Note this is only possible when data uncertainty is separate from the model uncertainty. To demonstrate this, we compare our model with NSP as shown in Fig. 4.4, where we only use the data uncertainty component. In Fig. 4.4 left, our model and NSP use estimated forces to explain the predictions at one timestep. The force from the scene is zero and is not visualized here. Here our model can further decompose the data uncertainty into force uncertainties. In Fig. 4.4 right, three stochastic forces are considered. Note that the scene force has an obvious narrow high probability density (yellow) region. Namely, the scene force has the most concentrated distribution with the smallest variances. This means the pedestrian pays more attention to avoiding obstacles when he/she is close to the car than other factors, and our model is most certain about the scene factor. Although uncertainty is also captured in NSP, they only employed the deterministic forces to explain the prediction. This might be because their uncertainty is captured by a CVAE, which mixes data and model uncertainty and cannot be further decomposed. More experimental results can be found in the appendix B.

The explainability in our model is from investigating the uncertainties in forces. As a result, our model can provide human-understood model predictions, which is different from traditional black-box deep learning methods [26, 73]. When people know how and why a model outputs a specific prediction, they are more likely to trust and adopt it [249]. In addition, the explainability with informative stochastic forces enables our model to provide lots of information, which is useful in relevant applications such as crowd management [250] and autonomous vehicles [251]. For instance, we can obtain the influence of the environment on pedestrians from our model. This can help us refine the scene layout to enhance safety and optimize efficiency in crowd management [250].

4.4.3 Trajectory Forecasting

To measure the accuracy of trajectory prediction, we use two established error metrics: average displacement error (ADE) and final displacement error (FDE). They are defined as:

$$ADE = \frac{1}{Mt_f} \sum_{i=1}^{M} \sum_{t=t_h+1}^{t_h+t_f} \|\mathbf{c}_i^t - \mathbf{\hat{c}}_i^t\|_2, \quad FDE = \frac{1}{M} \sum_{i=1}^{M} \|\mathbf{c}_i^{t_h+t_f} - \mathbf{\hat{c}}_i^{t_h+t_f}\|_2, \quad (4.16)$$

where M is the number of trajectories, \mathbf{c}_i^t is the true coordinate, and $\hat{\mathbf{c}}_i^t$ is the prediction. Ten state-of-the-art methods are used as baselines: Social GAN [26], Sophie [70], PECNet [73], Y-Net [83], S-CSR [74], SocialVAE [242], V²-Net [252], NSP [80], CSR [253], and TUTR [79]. Following prior research, the best ADE and FDE of multiple predictions are reported. We adopt two popular strategies to get multiple trajectories. One is standard-sampling, which allows the model to predict 20 complete future trajectories individually, and the other is ultra-sampling, which allows the model to sample 20

Table 4.1: Results on SDD based on standard-sampling. XX/XX is ADE/FDE. Reported errors are in pixels. Lower is better.

SocialGAN	Sophie	PECNet	Y-net	SocialVAE	V ² -Net	NSP	TUTR	Ours
27.23/41.44	16.27/29.38	9.96/15.88	7.85/11.85	8.10/11.72	7.12/11.39	6.52/10.61	7.76/12.69	6.46/10.49

Table 4.2: Results on ETH/UCY based on standard-sampling. XX/XX is ADE/FDE. The unit is meters. Lower is better.

	SocialGAN	Sophie	PECNet	Y-net	SocialVAE	V^2 -Net	NSP	TUTR	Ours
ETH	0.81/1.52	0.70/1.43	0.54/0.87	0.28/0.33	0.41/0.58	0.23/0.37	0.25/0.24	0.40/0.61	0.25/0.25
HOTEL	0.72/1.61	0.76/1.67	0.18/0.24	0.10/0.14	0.13/0.19	0.11/0.16	0.09/0.13	0.11/0.18	0.09/0.11
UNIV	0.60/1.26	0.54/1.24	0.35/0.60	0.24/0.41	0.21/0.36	0.21/0.35	0.21/0.38	0.23/0.42	0.20/0.38
ZARA1	0.34/0.69	0.30/0.63	0.22/0.39	0.17/0.27	0.17/0.29	0.19/0.30	0.16/0.27	0.18/0.34	0.16/0.27
ZARA2	0.42/0.84	0.38/0.78	0.17/0.30	0.13/0.22	0.13/0.22	0.14/0.24	0.12/0.20	0.13/0.25	0.12/0.19
AVG	0.58/1.18	0.54/1.15	0.29/0.48	0.18/0.27	0.21/0.33	0.18/0.28	0.17/0.24	0.21/0.36	0.16/0.24

possible coordinates at each future timestep [80].

Tab. 4.1 and Tab. 4.2 present the quantitative results based on the standard-sampling. Our model outperforms all baselines on two public datasets across ADE and FDE. We observe that our model improves the previous state-of-the-art ADE and FDE by 0.06 and 0.12, respectively, on SDD. On ETH/UCY, our model gains the maximal 15.38% improvement on the HOTEL subset in the FDE metric. On average, our model achieves the best performance on ETH/UCY with an improvement of 5.88% in ADE. Further, the quantitative results based on the ultra-sampling are demonstrated in Tab. 4.3. Our model outperforms the best baseline by 14.61% and 60.17% on SDD in ADE and FDE, respectively. Although the ADE on ETH/UCY of our model is the same as the stateof-the-art performance, our model improves the previous best FDE significantly by 50%. Overall, these quantitative results demonstrate that our model with find-grained uncertainty possesses state-of-the-art prediction accuracy.

4.4.4 Generalization

Generalization is the model's important ability to adapt to unseen data. We evaluate the generalization ability of trajectory prediction models by measuring their performance on unseen diverse scenarios. Following the setting in [80], we efficiently generate unseen scenarios by increasing the crowd density which is measured by the highest number of people (HNP) simultaneously in the scene, while adopting the collision rate as

Table 4.3: Results on ETH/UCY and SDD based on ultra-sampling. XX/XX is ADE/FDE.

	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG	SDD
S-SCR	0.19/0.35	0.06/0.07	0.13/0.21	0.06/0.07	0.06/0.08	0.10/0.16	2.77/3.45
NSP	0.07/0.09	0.03/0.07	0.03/0.04	0.02/0.04	0.02/0.04	0.03/0.06	1.78/3.44
CSR	0.28/0.53	0.07/0.08	0.24/0.35	0.07/0.09	0.05/0.09	0.14/0.23	4.87/6.32
Ours	0.05/0.04	0.03/0.02	0.03/0.03	0.02/0.02	$\boldsymbol{0.02/0.02}$	0.03/0.03	1.52/1.37



Figure 4.5: Generalization experiments. The collision rate and the number of collisions against the HNP are shown in (a) and (b) respectively. Both horizontal axes represent the HNP from 50 to 200. The vertical axes in (a) and (b) represent the collision rate and the number of collisions respectively.

a metric. We evaluate the generalization ability of our model and other three baseline methods Y-net [83], S-CSR [74] and NSP [80]. Specifically, we use the scene Coupa0 from SDD and its large space can accommodate a lot of people in theory. However, the HNP in the original Coupa0 is merely 11. We increase the HNP to 50, 100, 150 and 200 people to obtain new scenarios. We run all methods trained on the training data of SDD as simulators. We simulate the motion of agents batch by batch since the boundary of the Coupa0 can't accommodate more than 50 agents simultaneously. Following [80], three intervals during the simulation are employed to calculate the average collision rates, where we regard each agent as a disc with a radius r = 7.5 pixels. One collision is counted if the minimum distance between two trajectories falls below 2r at any time. Finally, we calculate the collision rate as $R_{col} = \frac{M}{N(N-1)/2}$, where N is the number of agents in the scene and M is the number of collisions.

The collision rates of three intervals for each method are averaged in every simulation. These results are shown in Fig. 4.5 (a). We note that our model always achieves

Table 4.4: Results on partial training data of SDD based on ultra-sampling. XX/XX is ADE/FDE. All methods are trained by using partial training data of SDD (from 25% to 100%) and are tested on the testing data of SDD.

	S-CSR	NSP	Ours
Full SDD	2.77/3.45	1.78/3.44	1.52/1.37
$50~\%~{\rm SDD}$	3.78/5.09	2.03/4.15	1.89/1.80
$25~\%~{\rm SDD}$	4.60/6.44	2.10/4.32	2.01/2.00

the best performance. In addition, we also show the number of collisions averaged over three intervals in Fig. 4.5 (b). The number of collisions increases for all models as the density becomes higher. However, our model outperforms other baselines across all simulation scenarios in terms of the collision number. Our model also possesses the lowest increase rate of the collision number when the density becomes higher. Overall, these results demonstrate that our model has the strongest generalizability to unseen high-density scenarios.

4.4.5 Data Efficiency

Collecting clean data for human trajectory prediction is both costly and time-consuming, often requiring manual labeling and verification. As a result, data efficiency is essential. We conduct experiments to evaluate the data efficiency of our model and two baseline methods S-CSR [74] and NSP [80]. We decrease the training data of SDD to 50% and 25% and train our model and two baselines while evaluating them on the testing data of SDD. The results are shown in Tab. 4.4. Our model always achieves the best performance on different proportions of training data and then has the highest data efficiency. Both NSP and our model benefit from the neural differential equations whose explicit physical models can act as regularization in learning [226] and have high data efficiency. Furthermore, our model performs better than NSP in data efficiency because of our uncertainty-aware structure and data-efficient way of capturing the data uncertainty. Specifically, unlike NSP learning the complete mixed uncertainty by neural networks, respectively, while we only need a few data to calibrate the data uncertainty in the explicit model. Also, prior knowledge of human behaviors can be incorporated

\mathbf{D}_{a}	~	✓	~	~
\mathbf{D}_n	×	✓	✓	✓
\mathbf{D}_s	×	✓	✓	✓
ε	×	×	×	✓
ultra	×	×	✓	✓
BDP	6.52/10.57	6.46/10.49	2.32/2.66	1.52/1.37

Table 4.5: Ablation study on SDD. XX/XX is ADE/FDE.

to improve further the data efficiency of learning the data uncertainty in the explicit model.

4.4.6 Ablation Study

To further understand the effectiveness of different components in our model, we conduct ablation experiments on SDD, shown in Tab. 4.5. SDD contains various trajectories and is used widely in human trajectory prediction. Previous research [80, 254, 255] demonstrates that the ablation experiments on SDD can evaluate well the contribution of different components in a model. The true and false of ultra in Tab. 4.5 correspond to the ultra-sampling and the standard-sampling. From the second column in Tab. 4.5, we note that our model can obtain good results when only considering \mathbf{D}_a because the actuation force determines the motion trend. The third column shows that our model acquires more accurate results by introducing \mathbf{D}_n and \mathbf{D}_s , which model the pedestrian-pedestrian interaction and pedestrian-scene interaction. Next, significant improvement is obtained by adopting the ultra-sampling. From the result of the fourth column in Tab. 4.5, we see that the particle system in our model can already capture the stochasticity of dynamics well. This means that distributions \mathbf{D}_a , \mathbf{D}_n , and \mathbf{D}_s captured by our model can depict the dynamics of human trajectories correctly. Finally, we gain a superior result 1.52/1.37 by adding the estimation of the model uncertainty ε . This demonstrates that the data uncertainty and the model uncertainty should be considered together. The total uncertainty is captured well by exploiting the excellent data-fitting capacity of deep learning.

4.5 Conclusion

We have proposed a novel Bayesian stochastic social force model for human trajectory prediction. Our uncertainty-aware model captures the fine-grained uncertainty via a decoupling strategy. Our model possesses strong explainability while achieving stateof-the-art prediction accuracy on popular public datasets. One limitation is that our model doesn't incorporate psychological factors such as affective states. In our future work, we will extend our model to consider important and meaningful psychological factors to enhance the explainability further. In addition, we would like to explore and apply our models in other areas such as crowd management and autonomous driving.

Chapter 5

Human Motion Prediction under Unexpected Perturbation
This chapter investigates a new task in human motion prediction, which is predicting 3D motions under unexpected physical perturbation potentially involving multiple people. We assume that the human body is represented by a matrix $X \in \mathbb{R}^{J \times 3}$, where J is the joint number. Then, a motion consists of a sequence of X. Compared with existing research, this task involves predicting less controlled, unpremeditated, and pure reactive motions in response to external impact and how such motions can propagate through people. It brings new challenges such as data scarcity and predicting complex interactions. To this end, we propose a new method capitalizing differentiable physics and deep neural networks, leading to an explicit Latent Differentiable Physics (LDP) model. Through experiments, we demonstrate that LDP has high data efficiency, outstanding prediction accuracy, strong generalizability, and good explainability. Since there is no similar research, a comprehensive comparison with 11 adapted baselines from several relevant domains is conducted, showing LDP outperforming existing research both quantitatively and qualitatively, improving prediction accuracy by as much as 70%, and demonstrating significantly stronger generalization.

5.1 Introduction

Human motion prediction aims to predict the future movements given the past motions, which has been heavily studied in computer vision [41–44]. Deviating from existing research, we are interested in a new task setting: predicting human motions, on both individual and group levels, under unexpected physical perturbation. On the individual level, physical perturbation causes reactive motions as opposed to active motions. On the group level, such perturbations can propagate through people while possibly being intensified, *e.g.* a push at the back of a line of people could be transferred all the way to the front. These motions have not been investigated. Incorporating physical perturbation potentially extends motion prediction to new application domains *e.g.* balance recovery in biomechanics [47, 48], reactive motions for character animation [14, 45], crowd crush induced by pushing [3, 46], humanoid robots [256, 257], *etc.*

Incorporating physical perturbation in prediction imposes new challenges. First, the motions are purely reactive and less controlled such that they are less smooth and less coordinated among body parts. Furthermore, this perturbation can propagate through people when they are packed and the space to recover balance is restricted, such that an attempt to recover balance relies on pushing others. Last but not least, unlike existing research, the data for motion prediction under perturbation is extremely scarce. Not only is it rare to capture full-body motions under such circumstances, but it is also difficult to record the interactions between people, *e.g.* forces of pushes.

Before deep learning, many areas have formulated this problem, which can be broadly divided into two categories. The first is physics-based where human bodies are simplified into connected rigid bodies [3, 258]. The reaction to push is solved via optimization to compute what forces are needed to recover balance [125, 259], or through carefully tuning feed-forward controllers [258, 260]. These methods, despite aiming to mimic the balance recovery of humans, do not learn from human data and therefore cannot predict human motions. Alternatively, reactions to perturbation can be learned from data via regression [261], optimization [262], or reinforcement learning [263]. Comparatively, this type of method tends to generate more human-like motions, but they are not designed for prediction.

Recently, deep learning [16, 41, 43, 264] have dominated human motion prediction, but they cannot be adapted for our problem. First, most datasets only contain single-body motions without external perturbation. Even when multiple people are captured, it is not under unexpected perturbation. To predict push propagation, one would still need to measure information e.g. contact forces between people, ground friction, muscle forces, etc., which are all absent. This data scarcity essentially rules out most deep learning methods. Furthermore, there is also little work in modeling the physical/bio-mechanical interactions that can potentially propagate through people. Current research includes motion forecasting, generation, and synthesis. Most motion forecasting methods [41, 42, 265] are for a single person, with a few recent exceptions [16, 43, 44] but not involving perturbation. Alternatively, our problem could be formulated as motion generation conditioned on external perturbation. However, current methods [266–271] again do not explicitly model close interactions among multiple people caused by perturbation. Theoretically, motion synthesis [272-275] is a possibility, which potentially can predict motions under perturbation. But they require dense control signals to guide the synthesis, or/and extensive physical simulation. Therefore, it requires manual labor or/and is difficult to scale to many people.

To address the aforementioned challenges, we need a model that has high data efficiency, strong generalizability, and can model interactions between people. In other words, this model needs to be able to learn from a small number of samples, can predict accurately in situations similar to the data, and is capable of generating plausible motions in drastically different scenarios. To this end, we propose a new deep learning model for human motion prediction under unexpected perturbation. To address the data scarcity, we propose a scalable differentiable physics (DP) model for the human body, to learn the balance strategy and interaction propagation between people, inspired by recent DP research [185, 276, 277]. However, naively following existing DP approaches means we would need to make the full-body simulation differentiable for each individual. Not only is motion intrinsically indifferentiable due to *e.g.* foot contact, but full-body physical models are too computationally expensive to scale. Therefore, we propose a latent DP space where the full-body physics is reduced into a differentiable inverted pendulum model (IPM) [278–281], and the full-body poses are mapped to and recovered from the IPM. At the low level, the IPM governs body physics and learns key forces such as ground friction and balance recovery. As the IPM is simple, the required data is small. At the high level, we use neural networks to recover the fullbody pose from the IPM, which also does not require much data as the IPM provides strong guidance. We refer to our model as the Latent Differentiable Physics (LDP) model. Note different from other latent physics models where the dimensionality reduction is implicit [282, 283], ours is explicit and physically meaningful (*i.e.* mapping from full-body to IPM).

We show LDP can learn from very limited data and perform well under many widely used metrics. Since there is no similar work to our best knowledge, we adapt a wide range of baseline methods in the most relevant areas (motion forecasting, motion generation, and motion synthesis), in single-person and multi-people scenarios, for comparison. The results demonstrate that LDP outperforms them both quantitatively and qualitatively. Notably, our model exhibits remarkable generalizability. It can accommodate unseen out-of-distribution perturbations, group sizes, and group formations, potentially extending our research beyond human motion prediction into broader areas, *e.g.* crowd simulation. Furthermore, owing to the explicit physics model, our model possesses a distinctive feature: explainability, providing plausible explanations for the predicted motion. Formally, our contributions include:

- A new task: human motion prediction under unexpected perturbation. To our best knowledge, this is the first deep learning work addressing this problem.
- A novel differentiable physics model in human motion prediction that explicitly

considers physical interactions.

- A new differentiable IPM model that learns body physics under complex interactions.
- A novel differentiable interaction model that can learn interactions and interaction propagation.

5.2 Related Work

Human Motion Prediction. Compared with traditional statistical machine learning [144, 148], deep learning has dominated human motion prediction recently. It can be formulated as a sequence-to-sequence task modeled by Recurrent Neural Networks [151, 152, 155]. Also, human skeletons can be seen as graphs so that spatiotemporal graph convolutions can be employed [159, 160, 265, 284, 285]. Transformerbased methods [166, 167] use the attention mechanism to capture spatial and temporal correlations. Recently, there has been a surge of interest in multi-people motion prediction [16, 43, 44, 169]. MRT [169] models the social interactions between humans via a global encoder. JRFormer [43] exploits the joint relation representation for modeling the interactions where physical interactions are considered implicitly. However, existing methods share a common limitation - they do not consider unexpected perturbations, restricting their applications in predicting actively planned/controlled motions. Additionally, explicit physical interactions between people have often been overlooked in these methods. Our model extends the research to a more challenging scenario involving unexpected perturbation and perturbation propagation. The explicit physics knowledge in our model enables it to achieve better prediction, generalizability, and explainability.

Traditional Research on Balance Recovery Relevant research has been conducted in other fields where traditional methods mainly focus on modeling balance recovery strategies in response to perturbations [46–48, 256, 257, 259]. Brodie *et al.* [48] analyzed the biomechanical mechanisms in the balance recovery following an unexpected perturbation such as trips and slips. Chen *et al.* [46] studied the dynamics of individuals under pushing in crowds. A new controller was proposed to recover balance for bipedal robots under perturbation [259]. In parallel, some traditional methods aim to synthesize reactive motions to perturbation [14, 125, 261, 286, 287]. Arikan *et al.* [14] proposed an algorithm for selecting and adjusting the motions from data to synthesize the motion for animating virtual characters being pushed. [286, 287] explored how to turn the given motions under perturbation into physically valid ones. Overall, traditional methods cannot predict motions under perturbation, either because they do not learn from data or have limited learning capacity. By contrast, we incorporate DP with deep neural networks to predict such human motions.

Differentiable Physics. DP is an emerging field focusing on combining traditional physics models with deep learning techniques, to provide high data efficiency and explainability. Consequently, many domains have investigated differentiable physics such as robotics [182, 183, 288], physics [289, 290], and computer graphics [185, 291]. We propose the first explicit latent differentiable physics model for human motion prediction under unexpected physical perturbation.

5.3 Methodology

Problem Definition. Given a motion with multiple people, we denote the skeletal pose of the *n*th person at frame t as $X_t^n \in \mathbb{R}^{J \times 3}$ where J is the joint number. Unlike existing research aiming to predict p frames $\{\hat{X}_{T-p+1:T}^n\}_{n=1}^N$ given k frames $\{X_{1:k}^n\}_{n=1}^N$ history, we minimize the required history due to limited data. Given the initial frame $\{X_0^n\}_{n=1}^N$ and the input forces F^{input} , we aim to predict the following T frames, by solving an initial problem:

$$\{\hat{X}_{1:T}^n\}_{n=1}^N = \mathcal{S}_{\gamma}(\{X_0^n\}_{n=1}^N, IPM_{\eta}(\mathcal{M}(\{X_0^n\}_{n=1}^N), F^{input})),$$
(5.1)

where $\{\hat{X}_{1:T}^n\}_{n=1}^N$ is the predicted T frames. \mathcal{M} is a Skeleton-to-IPM mapping $\mathcal{M}: \mathcal{X} \to \mathcal{I}$ where \mathcal{X} and \mathcal{I} are the state space of skeleton poses (represented by joint positions) and the IPM respectively. IPM_η is a Differentiable IPM with learnable parameters η . Finally, S_γ is the inverse mapping, *i.e.* Skeleton Restoration Model, $S_\gamma: \mathcal{I} \to \mathcal{X}$, reconstructing full-body skeleton poses from IPM states, with learnable parameters γ . An overview of our model is shown in Fig. 5.1. Given a motion, we map the full-body poses into their corresponding states of an IPM [279–281] as $\{I_0^n\}_{n=1}^N = \mathcal{M}(\{X_0^n\}_{n=1}^N)$. By simulating the IPM forward in time via IPM_η , it can learn the key parameters η . The interaction forces between people are also learned simultaneously. Meanwhile, our Skeleton Restoration Model S_γ recovers the full-body poses from the predicted IPM states from IPM_η . For training, we minimize the mean squared error (MSE) between



Figure 5.1: Overview of LDP. Given a frame X_t , it is first mapped into the IPM space via Skeleton-to-IPM to get its IPM state I_t . Then I_t is simulated for one step via Differentiable IPM to compute I_{t+1} . Lastly, the full-body frame X_{t+1} is recovered from I_{t+1} via Skeleton Restoration Model. The IPM is shown in the right figure. The fullbody state X is represented by joint positions.

the predicted $\{\hat{X}_{1:T}^n\}_{n=1}^N$ and the ground-truth poses $\{X_{1:T}^n\}_{n=1}^N$:

$$Loss = MSE(\{\hat{X}_{1:T}^n\}_{n=1}^N, \{X_{1:T}^n\}_{n=1}^N),$$
(5.2)

where we need to specify S_{γ} , IPM_{η} and \mathcal{M} in Eq. 5.1. We give key equations and model information below and refer the readers to the appendix C for details.

5.3.1 Latent Physics Space for Full-body Motions

Background and Skeleton-to-IPM Mapping

We first introduce IPM_{η} and \mathcal{M} in Eq. 5.1. Differentiable physics has shown extremely high data efficiency because physics can act as a strong inductive bias and eliminate the reliance on large amounts of training data [276, 277, 292]. For our model, a key design choice is to choose a DP model that has the right level of granularity while being scalable. Among many possible choices from full-body physics [293] to simple rods [294], we choose the Inverted Pendulum Model [279–281] as it can fully capture balance loss and recovery while being scalable.

Our IPM has a massless rod mounted to a cart with a point mass at the end of the rod (Fig. 5.1 right). Denoting its state $\mathfrak{I} \ni I_t = [x_t, y_t, \theta_t, \phi_t] \in \mathbb{R}^4$ at time step t where [x, y] is the coordinates of the cart in the xy-plane and $[\theta, \phi]$ is the rotation angles of the rod around Y_L axis and X_L axis in the local coordinate system Σ_L , respectively. Our full-body pose X is represented by 22 joint positions. \mathcal{M} in Eq. 5.1 is defined as (Fig. 5.1 left): the hip joint is mapped onto the point mass, and the midpoint between

the two ankle joints is mapped onto the center of the cart. The point mass and the cart jointly determine the two angles $[\theta, \phi]$.

Next, IPM_{η} is defined. Given the initial IPM state, we can simulate it in time by solving Eq. 5.3 repeatedly [278]:

$$M(I_t, l_t)\ddot{I}_t + C(I_t, \dot{I}_t, l_t) + G(I_t, l_t) = F_t^{net},$$
(5.3)

where $M \in \mathbb{R}^{4\times 4}$, $C \in \mathbb{R}^{4\times 1}$ and $G \in \mathbb{R}^{4\times 1}$ are the inertia matrix, the Centrifugal/Coriolis matrix, and the external force such as gravity, which are all functions of state I_t , its first-order derivative \dot{I}_t and the rod length l_t . While the standard IPM has a fixed rod length, we allow it to change as the distance between the hip and the middle of two ankles can drastically change in human motions. Therefore, we also predict l_t at each time step. Overall given the net force $F_t^{net} \in \mathbb{R}^4$ and the rod length l_t , we can solve Eq. 5.3 for the next state I_{t+1} via a semi-implicit scheme $\dot{I}_{t+1} = \dot{I}_t + \Delta t \ddot{I}_t$ and $I_{t+1} = I_t + \Delta t \dot{I}_{t+1}$, where Δt is the time step.

Finally, the learnable parameters η in IPM_{η} parameterize F_t^{net} and the rod length l_t , where the formulation differs between single-person and multi-people, and will be elaborated later. It's notable that F_t^{net} in Eq. 5.3 is the generalized force. Using the generalized force (instead of the Euler force) keeps the motion equation simple, and its entries have explicit physical meanings as shown later.

Single-Person Prediction via Differentiable IPM

Under single-person, we only consider Balance-Recovery and Friction (blue blocks in Fig. 5.1 Differentiable IPM) when predicting F^{net} . Specifically, we consider three forces:

$$F_t^{net} = F_t^{self} + f_t + F_t^{input}, ag{5.4}$$

where F_t^{self} , f_t , and F_t^{input} are the balance recovery force, the ground friction and the external perturbation. The Balance-Recovery module learns F_t^{self} which is further decomposed into $F_t^{self} = F_t^{self-pd} + F_t^{self-nn}$. This decomposition is because F_t^{self} is the muscle force at the hinge of the rod which serves two purposes. The first one is to give a feed-forward torque $F_t^{self-pd}$ to react to perturbation for balance recovery, and the second is to give a torque correction $F_t^{self-nn}$ for tracking observed motions. In generalized forces, we parameterize $F_t^{self-pd}$ by proportional derivative (PD) control:

$$F_t^{self-pd} = K_p e_t + K_d \dot{e}_t, \ e_t = s_d - s_t, \tag{5.5}$$

where e_t is the PD state error, K_p and K_d are the control parameters. Different from the IPM state, the current PD state is $s_t = [\dot{x}_t, \dot{y}_t, \theta_t, \phi_t]$ and the desired PD state s_d is [0, 0, 0, 0]. In other words, we assume people tend to recover to the upright body pose and zero linear velocity after unexpected perturbation, which is a widely accepted assumption [295–297]. However, $F_t^{self-pd}$ only captures the general balance recovery strategy. To mimic the data, we parameterize $F_t^{self-nn}$ with a Long Short Term Memory (LSTM) network:

$$F_t^{self-nn} = LSTM([\theta_t, \phi_t, \dot{x}_t, \dot{y}_t, \dot{\theta}_t, \dot{\phi}_t, M]),$$
(5.6)

where M is the mass of the person.

Ground friction f_t is the main reason for successful self-balance and therefore needs to be explicitly considered. In generalized forces, friction affects the IPM motion via damping [278]. So we parameterize $f_t = -\mu[\dot{x}_t, \dot{y}_t, 0, 0]$, where the parameter μ is a learnable positive scalar and shared by all people for simplicity. The damping force only directly influences the cart motion. Finally, to compute Eq. 5.3, we also need to predict the change of the rod length l_t , where we employ a multi-layer perception (MLP):

$$\triangle l_t = MLP([\theta_t, \phi_t, \dot{x}_t, \dot{y}_t, \dot{\theta}_t, \dot{\phi}_t, F_t^{self}, M, l_t]), \tag{5.7}$$

where l_t is the rod length at time step t. We predict the rod length at the next time step by $l_{t+1} = l_t + \Delta l_t$. Finally, after obtaining the prediction of F_t^{net} and l_t at every time step t, we can calculate the next IPM state by solving Eq. 5.3 via the semi-implicit scheme mentioned above.

Multi-people with Differentiable Interaction

When there is more than one person, the complexity increases quickly. The main reason is that the interaction propagation among people is: (1) complex, *e.g.* complicated contact positions/duration/forces, (2) hard to capture in data. Therefore, we propose to consider them as latent variables that cannot be directly observed. But again large amounts of data would be needed if we only relied on data to infer these variables. Therefore we model the interactions in the reduced IPM space, rather than the original space, so that it becomes a Differential Interaction Model (DIM).

Our DIM models a differentiable interaction force between any two IPMs and is learned in the Interaction module (the yellow block in Fig. 5.1 Differentiable IPM). The overall net force on an IPM in multi-people then becomes:

$$F_t^{net} = F_t^{self} + F_{t,n}^{inta} + f_t + F_t^{input},$$
(5.8)

where F_t^{self} , f_t and F_t^{input} are the same as Eq. 5.4. Note that all forces are learned and shared among all people, so that we can generalize to an arbitrary number of people later. $F_{t,n}^{inta} \in \mathbb{R}^4$ is the new interaction force:

$$F_{t,n}^{inta} = \sum_{j \in \Omega_{t,n}} F_{t,nj}^{inta} = \sum_{j \in \Omega_{t,n}} F_{t,nj}^{inta-bs} + F_{t,nj}^{inta-nn},$$
(5.9)

where $\Omega_{t,n}$ is the neighborhood of the person n at time t. $F_{t,nj}^{inta}$ is the interaction force applied onto person n from her/his neighbor $j \in \Omega_{t,n}$. We model two factors in $F_{t,nj}^{inta-bs}$ and $F_{t,nj}^{inta-nn}$. The first $F_{t,nj}^{inta-bs}$ represents a consistent and trackable repulsive tendency when two IPMs get close, while $F_{t,nj}^{inta-nn}$ captures the variations of the repulsion. So we expect $F_{t,nj}^{inta-bs}$ to capture most of the interaction while $F_{t,nj}^{inta-nn}$ being a supplement. To this end, we separate the dimensions of an IPM state I = $[x, y, \theta, \phi]$ into two groups [x, y] and $[\theta, \phi]$ and treat them separately as $F_{nj}^{bs-xy} \in \mathbb{R}^2$ and $F_{nj}^{bs-\theta\phi} \in \mathbb{R}^2$, such that $F_{t,nj}^{inta-bs} = [F_{nj}^{bs-xy}, F_{nj}^{bs-\theta\phi}]^{\mathbf{T}}$, where we omit the time subscript t and the superscript *inta* for simplicity.

For F_{nj}^{bs-xy} , we define a repulsive potential energy between two close IPMs which leads to a repulsive force:

$$F_{nj}^{bs-xy}(r_{nj}) = -\nabla_{r_{nj}} \mathcal{U}[b(r_{nj})], \quad \mathcal{U}[b] = ue^{-\frac{b}{\sigma}},$$

$$b = \frac{1}{2} \sqrt{(\|r_{nj}\| + \|r_{nj} - \triangle t\dot{r}_{jn}\|)^2 - \|\triangle t\dot{r}_{jn}\|^2}, \quad (5.10)$$

where $r_{nj} = r_n - r_j$ is the relative position of the carts of a person and his/her neighbor j, *i.e.* r_n is the vector [x, y] in the IPM state I_n . The $\mathcal{U}[b]$ is the repulsive potential with elliptical equipotential lines, and u and σ are hyperparameters. b is the semi-minor axis of the ellipse where $\dot{r}_{jn} = \dot{r}_j - \dot{r}_n$ is the relative velocity.

For $F_{nj}^{bs-\theta\phi}$, we treat it as a force with a constant magnitude (tunable hyperparameter) and apply it on θ and ϕ independently. Although the magnitude is constant, its directions can vary in different situations. We explain it for θ and the same principle applies to ϕ . On the high level, we need to decide the direction of $F_{nj}^{bs-\theta\phi}$ based on the states of two close IPMs. θ can be positive, zero and negative. For two IPMs, this produces a total of 9 possible states, which we detail in the appendix C. After defining $F_{t,nj}^{inta-bs}$, we explain $F_{t,nj}^{inta-nn}$ which should capture the variation of interactions. Unlike $F_{t,nj}^{inta-bs}$ where we can define an explicit form, we learn $F_{t,nj}^{inta-nn}$ via an MLP:

$$F_{t,nj}^{inta-nn} = MLP([x_{nj}, y_{nj}, \theta_n, \phi_n, \theta_j, \phi_j, \dot{x}_{nj}, \dot{y}_{nj}, \dot{\theta}_{nj}, \dot{\phi}_{nj}]),$$
(5.11)

where $x_{nj} = x_n - x_j$ and $\dot{x}_{nj} = \dot{x}_n - \dot{x}_j$. y_{nj} , \dot{y}_{nj} , $\dot{\theta}_{nj}$ and $\dot{\phi}_{nj}$ are computed in a similar fashion.

5.3.2 Skeleton Restoration Model

To predict full-body motion, we recover the full-body pose from the predicted IPM states. This is divided into two steps as shown in Fig. 5.1. We first recover the lower body from the IPM state, then recover the upper body from both the IPM state and the recovered low body. There are two reasons for this design. First, the Skeleton-to-IPM mapping dictates that the IPM has a higher correlation with the lower body than with the upper body. Also, the dynamics of the lower body and the upper body are relatively independent [266, 298], *i.e.* similar low-body motions can correspond to different upper-body motions, *e.g.* different styles in walking. Therefore, we use two models to recover the lower body and the upper body, respectively. Overall, although the Skeleton Restoration Model involves deep neural networks, the required data is small as there is strong IPM guidance.

Lower Body Restoration. We use a Conditional Variational Autoencoder (CVAE) [266, 268, 273] (CVAE-Lower in Fig. 5.1) to learn a Normal distribution of the lower body X_{t+1}^l in the latent space conditioned on X_t^l . During inference, since X_{t+1}^l is unavailable, we train a sampler (Lower Sampler) to sample the latent space to generate the next frame \hat{X}_{t+1}^l . The Lower Sampler network is an MLP. It takes as input X_t^l , I_{t+1} , and outputs a latent code of CVAE-Lower which is then decoded. Overall, CVAE-Lower takes as input the current lower body X_t^l and the predicted IPM state I_{t+1} , to predict the next lower body \hat{X}_{t+1}^l , essentially reconstructing the lower body under the IPM guidance.

Upper Body Restoration. Similarly, we also use a CVAE named CVAE-Upper, except this time we use both the lower body predicted by CVAE-Lower \hat{X}_{t+1}^l and the current upper body X_t^u as the condition. A sampler (Upper Sampler) is also used to take as input I_{t+1} , \hat{X}_{t+1}^l and X_t^u , and sample the latent space of CVAE-Upper, which is then decoded to predict the upper body at the next frame \hat{X}_{t+1}^u .



Figure 5.2: FZJ Push [18]. The blue agent was pushed by the punch bag and then he pushed other people.

5.3.3 Training with Auxiliary Losses

In summary, the learnable parameters of our model include: the LSTM (Eq. 5.6), the MLPs (Eq. 5.7, Eq. 5.11), the ground friction coefficient μ , CVAE-Lower, CVAE-Upper, Lower Sampler, and Upper Sampler. Other than the main loss in Eq. 5.2, we also use other auxiliary losses such as foot sliding, IPM state MSE, *etc.* We also pre-train some components for initialization. All details including training/prediction algorithms, implementation details, parameters, code, data, *etc.* are in the appendix C.

5.4 Experiments

5.4.1 Dataset and Metrics

Data for our problem is extremely scarce compared with other human motion prediction research. The only publicly available dataset, to our best knowledge, is a new dataset [18] named FZJ Push. The dataset includes standing individuals, groups of four, and groups of five, with one person pushed by a punching bag unexpectedly, and the push is propagated through the group. In total, the dataset includes only 45 single-person motions and 63 multi-people motions. This is considerably less than data normally used for human motion prediction. As shown later, the necessity of a model with high data efficiency is crucial. The motion is recorded at 60 FPS. Shown in Fig. 5.2 a, a hanging punch bag is operated by a person to give pushes of various magnitudes to one person in the group. Then the skeletal motions (Fig. 5.2 b) are recorded. There is a pressure sensor measuring the pushing forces on the punching bag. However, the

Method	MPJPE	hipADE	hipFDE	MBLE	FSE
A2M	0.403	0.386	0.730	0.019	0.200
ACTOR	0.362	0.338	0.591	0.020	0.434
MDM	0.500	0.424	0.686	0	2.567
RMDiffuse	0.228	0.202	0.299	0.011	0.790
PhyVae	0.260	0.249	0.460	0.009	0.170
siMLPe	0.130	0.117	0.226	0.006	0.182
EqMotion	0.296	0.270	0.543	0.064	1.552
Ours	0.097	0.086	0.171	0.002	0.131

Table 5.1: Metrics in single-person.

pushing forces between people are not recorded. We discard redundant data such as frames in waiting. More details like the size of the dataset, train/test split, and the time horizon of predicted motions are in the second section of the appendix C.

For evaluation, we adopt five widely used metrics [41, 44, 266]: Mean Per Joint Position Error (MPJPE) in meters, Average Displacement Error at the hip (hipADE) in meters, Final Displacement Error at the hip (hipFDE) in meters, Mean Bone Length Error (MBLE) in meters, and Foot Skating Error (FSE) in centimeters. Details and justifications for these metrics are in the appendix C.

5.4.2 Baselines

There is no similar work in human motion prediction to our best knowledge, so we carefully review a wide spectrum of research in motion prediction, synthesis, and generation, and choose the latest methods in each field for comparison. Specifically, we choose 11 baselines: A2M [299], ACTOR [268], MDM [269], RMDiffuse [270], PhyVae [273], siM-LPe [42] and EqMotion [41] for the single-person scenario, and MRT [169], DuMMF [44], TBIFormer [16] and JRFormer [43] for the multi-people scenario. The specific adaptation varies according to the baseline, and we give the details in the appendix C. One notable difference is our model only requires the first frame with the perturbation force during inference, while the other methods tend to require much more information such as multiple frames.

5.4.3 Quantitative Results

The single-person comparison is shown in Tab. 5.1. Despite requiring the minimal information, our model still achieves the best performance on all metrics except the



Figure 5.3: Perturbations with different magnitudes in single-person.

MBLE. MDM obtained 0 MBLE because its parameterization is joint angle based, *i.e.* no bone-length change incurred. A joint angle parameterization could also work with our model but in practice, we find a joint-position-based parameterization works better. Across different metrics, LDP outperforms the best baseline by as much as 25.38%, 26.50%, 24.34%, 66.67%, and 22.94% on MPJPE, hipADE, hipFDE, MBLE, and FSE respectively, excluding the MBLE of MDM. We tend to attribute the higher performance to the explicit physics-based inductive biases embedded in the design of LDP. Furthermore, we look into performances under perturbations with different magnitudes (weak, medium, and strong) in Fig. 5.3, where we only include the best three baselines and leave the full comparison in appendix C. Stronger pushes lead to stronger responses and tend to be harder to predict. This is especially obvious in metrics related to motion tracking, *i.e.* MPJPE, hipADE, and hipFDE, where as the push becomes stronger, the errors become larger. Comparatively, LDP consistently outperforms other baselines, demonstrating its effectiveness in strong perturbations. In addition, compared with weak and medium pushes, LDP has a slower error increment under strong pushes, in contrast to the more volatile performances of other baselines, showing better generalizability. Overall, LDP either ranks as the best or is close to the top performance across metrics and perturbation levels.

The results under the multi-people scenario are shown in Tab. 5.2. The MBLE of DuMMF is 0 because it employs joint-angle-based parameterization. Multi-people is a challenging task for all methods. On all metrics, LDP outperforms all baselines by at least 34.57%, 34.29%, 16.15%, 70%, and 70.51% on MPJPE, hipADE, hipFDE, MBLE, and FSE, respectively, (excluding the MBLE of DuMMF). Moreover, we show



Table 5.2: Metrics in multi-people.

Figure 5.4: Perturbations with different magnitudes in multi-people.

detailed analysis under perturbations with different magnitudes in Fig. 5.4, with the three best baselines. One challenge in multi-people is to predict the onset and duration of interactions. The baseline methods need to learn the interactions by purely fitting the data, while our method learns them as a latent physical process. Consequently, none of the baselines can predict well, *e.g.* they predict moving without being pushed or not moving while being pushed, while our model can learn to predict the interactions and their propagation well. Overall, our model achieves or is close to the best performance across metrics and perturbation levels.

5.4.4 Qualitative Results

We visually compare our methods with the best three baselines under single-person in Fig. 5.5. Our prediction has the highest quality and is the most similar to the ground truth. RMDiffuse severely violates bone lengths, especially around ankles, and generates jittering motions. PhyVae predicts walking but with rather small steps. siMLPe predicts only a single step. The multi-people scenario is much harder (Fig. 5.6), where both individual reactions and interactions need to be predicted. MRT and TBIFormer



Figure 5.5: Visual results in the single-person scenario.



Figure 5.6: Multi-people comparison.

suffer from serious intersections between individuals. JRFormer predicts merely subtle movements that deviate considerably from the ground truth. Our model generates the most similar prediction to the ground truth.



Figure 5.7: Learned net forces. We present the leaned net forces for the multi-people visualization results of our model in Fig. 5.6. These net forces are on the second person (from the left) person. The bar height indicates the magnitude and the sign indicates the direction, where the person moves in the positive direction of the x-axis.



Figure 5.8: A 13-person group in a diamond formation with three people (indicated by orange arrows) being pushed.

Explainability In Fig. 5.7, we show the learned net forces on the second person (from the left), to provide plausible explanations of the predicted motion. This person remains still initially under zero net force, then experiences a push from the first person, resulting in forces in x and θ , and small forces in y and ϕ . Then the third person is pushed by the second, resulting in the change of the net force on the second person from positive to negative in x and θ . Finally, the second person recovers the balance. Our model predicts the motion results from plausible forces and therefore possesses strong explainability.

5.4.5 Generalization

LDP can easily generalize to out-of-distribution scenarios, *e.g.* unseen pushes, more people, different formations, *etc.* Since there is no ground truth, we show the visual result of a challenging generalization scenario in Fig. 5.8, where 13 people stand in a

Method	MPJPE	hipADE	hipFDE	MBLE	FSE
no IPM, Full	0.217	0.195	0.341	0.007	0.196
no IPM, Low-up	0.206	0.184	0.320	0.009	0.313
IPM, Full	0.110	0.094	0.242	0.004	0.126
IPM, Low-up	0.106	0.092	0.218	0.003	0.069

Table 5.3: Ablation study with (1) IPM and no IPM, (2) Full body and Low-up body pose reconstruction.

diamond formation and 3 of them indicated by the orange arrows are pushed. Note the data only contain up to 5 people in simple formations such as one or two lines. So this 13-people formation is totally out-of-distribution. However, our model can still generate plausible motions for the entire group, given only the initial poses and the perturbation forces, demonstrating strong generalizability. More experiments can be found in the appendix C.

5.4.6 Ablation Study

The Differentiable IPM and the Skeleton Restoration Model are two key components of our model. We conduct the ablation study to assess the effectiveness of them. There are four combinations: with/without IPM, and full-body restoration or separate restoration (first lower body then upper body). When the IPM is absent, the next frame is directly predicted by either one full-body CVAE (Full) or two CVAEs with one for the lower body and the other for the upper body (Low-up). Without IPM, there are also no samplers (Lower Sampler and Upper Sampler in Fig. 5.1) so we need to directly sample in the latent space of the CVAEs. We randomly sample the latent space 3 times when predicting the next frame and average the results. In contrast, with IPM, we can train the samplers to only sample once to predict the next frame.

Results are shown in Tab. 5.3. When there is no IPM, the performance deteriorates significantly across all metrics. With the IPM guidance, all metrics are significantly improved. Further, the Low-up separation of the body improves the performance further across all metrics under the IPM guidance, especially on the FSE. However, it exhibits limited effectiveness without the IPM guidance, even resulting in a bad FSE. This is because IPM states have strong correlations with the lower body, without which the

Low-up is unable to improve the performance significantly even when the lower body is separately predicted. More details can be found in the appendix C.

5.5 Conclusion

We proposed a new task, human motion prediction under unexpected perturbation, which extends human motion prediction into new application domains. To this end, we have identified and overcome new challenges e.g. data scarcity and interaction modeling, by proposing a new class of deep learning models based on differentiable physics. Our model outperforms existing methods despite requiring far less information and shows strong generalization to unseen scenarios. One limitation is our method requires explicit modeling of the physical process, making the model not as general as black-box deep neural nets that can be plug-and-play on data. However, we argue this is mainly driven by the data scarcity. Also, it brings stronger generalizability and interpretability. In the future, we will investigate more general physics models that can potentially accommodate more diversified physical interactions between people. A big difference between other existing datasets [300, 301] and the dataset FZJ Push is the former is active motions while the latter is passive balance recovery. Another limitation of our model is its focus on passive balance recovery, which makes it less effective at handling active motions, such as dancing. We will also explore LDP on action motions in the future.

Chapter 6

Discussion, Conclusion and Future Work

6.1 Discussion

In this section, we further discuss the biases in the used data, the informative details about our methods, and the applications of these proposed models. We evaluate our methods proposed in Chapter 3 and Chapter 4 on public datasets SDD [21] and ETH/UCY [203, 204]. Although SDD and ETH/UCY contain numerous trajectories and have been employed widely, we still should pay attention to their inherent biases. Specifically, all data of SDD is captured on a university campus, which leads to three main biases: limited scenes, predominantly young pedestrians, and low-density crowds. Although ETH/UCY incorporates two urban scenes (Hotel and Zara) besides university scenes, it also suffers from the bias of limited scenes. In addition, ETH/UCY neglects other agents like cars and also focuses on low-density crowds. To address biases including limited scenes, predominantly young pedestrians, and the absence of other agents, we can combine diverse datasets to train the trajectory prediction models, and some prior knowledge such as the ages of pedestrians can be exploited to enhance the capability of these models in applications such as autonomous vehicles [28]. Given that we might confront crowds with various densities in applications like crowd management [27], it is also necessary to handle the data bias of lower-density crowds in SDD and ETH/UCY. A possible and easy solution is to combine models focusing on diverse densities and allow the combined model to estimate current crowd density. Then, the combined model can select the appropriate model to predict crowd dynamics according to the estimated density while we can use the SDD and ETH/UCY to train the model for low-density crowds. We use the public dataset FZJ Push [18] to evaluate our method on the proposed new 3D motion prediction task in Chapter 5. FZJ Push recorded pushing forces and accurate 3D motions and is the only available public dataset on our new task. However, FZJ Push is a small dataset because it is expensive to capture 3D motions under physical perturbations. Therefore, there are many data biases such as limited perturbation variation, limited group formations, the simple experimental scene, etc. in FZJ Push. We think that the most efficient way to resolve these data biases might be to collect more new data. Additionally, the introduction of biomechanical models [129, 132] might help the motion prediction model mitigate the data biases in applications like biomechanics [47] and character animation [45].

Our methods in Chapter 3 and Chapter 4 model crowd dynamics based on the social force model. However, unlike the hand-picked and fixed parameters in the so-

cial force model, the crucial parameters in our models are learnable and can adapt dynamically to the current states. Therefore, our models outperform the social force model in capturing crowd dynamics. For instance, our models can estimate appropriate parameters at each step to provide natural collision avoidance behaviors, whereas the social force model relies on predefined fixed parameters, resulting in rigid collision avoidance behaviors. In addition, different from the deterministic social force model, our methods consider the stochasticity of crowd dynamics. Our NSP-SFM in Chapter 3 models the stochasticity via a CVAE. Our BSSFM in Chapter 4 captures the data and model uncertainty by employing an explicit Bayesian neural SDE model and a deep generative network, respectively. The introduction of stochasticity enables our models to predict multiple potential future trajectories, aligning with the intrinsic stochasticity of human trajectories. Therefore, our methods can offer multiple possible collision avoidance behaviors, as opposed to the deterministic ones in the social force model.

Our trajectory prediction methods in Chapter 3 and Chapter 4 only model and predict future pedestrian trajectories. If other road users such as cars emerge in the view fields of pedestrians, our methods can regard them as obstacles. Although the treatment requires information of other road users, it can enable our methods to learn correct dynamics of pedestrians, as behaviors of pedestrians are often influenced by other road users [302, 303].

Our two trajectory prediction methods in Chapter 3 and Chapter 4 capture the stochasticity of future trajectories. However, NSP-SFM in Chapter 3 only captures the stochasticity via a CVAE, which mixes the data and model uncertainty. BSSFM in Chapter 4 explores the fine-grained stochasticity by decoupling the data and model uncertainty. Both of the two methods achieve excellent performance in stochastic trajectory prediction, which demonstrates that they can be applied in domains such as autonomous vehicles [304] by providing accurate prediction results. Further, BSSFM can explain their stochastic predicted trajectories based on the stochastic social forces. Therefore, predictions of BSSFM can provide more useful information like the source of stochasticity compared with those of NSP-SFM. This means that BSSFM might have a wider application range.

Finally, we provide the discussion about the application of models developed in this thesis. Extensive domains [28, 45, 305] can benefit from our models. For example, our trajectory prediction models can improve the performance of autonomous vehicles by

providing future movement prediction of pedestrians [28, 306]. The existing capabilities of our trajectory prediction models can achieve approximate 0.2 meters of ADE for 4.8 seconds of the prediction horizon. In many autonomous vehicles scenarios [303, 304, 307], prediction results having ADE within 1 meter can provide useful information and the minimum required prediction horizon is estimated as 1.6 seconds. Our 3D motion prediction model can estimate the complete balance recovery process with approximate 0.1 meters of MPJPE and then be applied in domains like character animation [14, 45] that expect complete motions but don't demand very high prediction accuracy. However, it is challenging to apply our models in some scenarios. Our trajectory prediction models don't consider the long-term prediction (more than 1 minute) and high-density crowds, which might hinder their application in many domains such as urban planning [4, 5]. The embedded IPM in our 3D motion prediction model might constrain its application in fields like crowd crush study [3, 46].

6.2 Conclusion

Understanding crowd behaviors is a complex and challenging research area that involves a wide range of disciplines such as computer science, physics, transportation, *etc.* However, the study of crowd behaviors always draws much interest from academia and industry because of its applications in a variety of practical tasks like crowd management, urban planning, *etc.* Though many successful methods have been proposed for crowd behavior research and significant progress has been gained [11–16], we still face many challenges in the area because of the complexity of human behaviors [17, 18]. This thesis concentrates efforts on two primary challenges, which are crowd dynamics modeling in low-density crowds and physical interaction modeling in high-density crowds. This thesis explores the former challenge by studying human trajectory prediction in Chapter 3 and uncertainty modeling in human trajectories in Chapter 4. Moreover, this thesis explores the latter challenge by solving a new research question proposed by us about 3D full-body human motions in Chapter 5.

Specifically, we propose a novel trajectory prediction framework based on neural differential equations, which incorporates the advantages of model-based and model-free methods, and introduce a new trajectory prediction model combining the social force model with neural networks under the framework to address the first research question in Chapter 3. Extensive experiments on two widely used public datasets demonstrate that our model can achieve state-of-the-art prediction accuracy and provide plausible explanations for model predictions. Moreover, we have shown that our model can generalize to unseen scenarios better than other baselines through simulation experiments. The primary limitation of our framework stems from the used physics system, which only considers low-density crowd dynamics. As a result, our framework fails to capture the dynamics of high-density crowds that the current physics system cannot depict accurately. Another limitation of our model lies in the coarse-grained environment repulsion, which views all objects in the environment as obstacles and employs the same parameters for all obstacles and pedestrians. The impact from the environment is not only the repulsion. It is also possible for some objects e.g. chairs in the environment to attract some pedestrians. Also, different pedestrians can have diverse responses even to the same object and different objects can influence a pedestrian to varying degrees.

Chapter 4 presents a novel Bayesian stochastic social force model to model the uncertainty of human trajectories, which aims to address the second research question. Our uncertainty-aware method decouples the uncertainty into the data uncertainty and the model uncertainty, estimating them through an explicit Bayesian neural SDE model and a deep generative model, respectively. Furthermore, an uncertainty-aware training scheme is introduced to enable both models to capture their desired uncertainties. Eventually, we employ Bayesian inference to incorporate prior knowledge of human behaviors. Exhaustive evaluation and comparison demonstrate that our method has strong explainability for predicted distributions of trajectories and outperforms existing methods in prediction accuracy. The major limitation is that our method ignores psychological factors such as affective states influencing human trajectories. Incorporating psychological factors in uncertainty modeling can provide more accurate uncertainty estimation and improve the explainability of models further.

In Chapter 5, we propose a new task (*i.e.* the third research question), predicting human motions under unexpected physical perturbation, to study physical interactions between individuals. Incorporating physical interactions extends motion prediction to other domains *e.g.* physics and biomechanics, but brings new challenges *e.g.* limited data and complex interactions to motion modeling. We proposed a latent differentiable physics model to overcome these new challenges, where a latent physics space and a deep generative model are introduced to learn body physics under physical interactions and restore 3D full-body motions from predictions in the latent physics space, respectively.

Comprehensive experiments show that our model performs better than existing research quantitatively and qualitatively. Moreover, the results of generalization experiments in several out-of-distribution scenarios demonstrate the strong generalizability of our model. However, our method still confronts two primary deficiencies. The first is that the simple physics model, IPM, in our method has limited capabilities in representing physical interactions. Another limitation is that our model focusing on passive balance recovery cannot handle active motions like dancing well.

6.3 Future Work

We will continue to improve and extend our work within this thesis in the future. This section presents potential directions and ideas based on the aforementioned limitations.

Given the limitations of our work in Chapter 3, we would like to extend it in two aspects in the future. First, we plan to expand our current framework to consider highdensity crowd dynamics. To achieve this goal, we can employ other physics models *e.g.* continuum models [24] to replace the existing physics system in the framework. Another potential extension concentrates on our model, where we would like to explore the fine-grained environment force. One possible solution is to train neural networks to estimate the parameters of the current environment forces from every category of objects for each pedestrian, where the total environment force is the summation of the environment forces from all objects. The inputs of the neural networks can be past trajectories and images of the surrounding environment/hand-crafted features.

The limitation of our model in Chapter 4 motivates us to extend our model by incorporating psychological factors in uncertainty modeling. One potential extension way is to introduce psychological models *e.g.* the OCEAN model [308] and the PAD model [309] that can provide analysis and classification of psychological states. Then we can estimate the psychological states of pedestrians based on their history trajectories, video data, *etc.* These estimated psychological states can be used as conditions to guide the future trajectory prediction of our model in Chapter 4 or other models [15, 80]. Additionally, we aim to explore applications of our model in autonomous vehicles, crowd management, *etc.*, as it can provide these safety-critical tasks with accurate and explainable predictions of pedestrian trajectory distributions.

Considering the limitations of our method in Chapter 5, we have two potential directions to extend our work. First, we can apply more general physics models such

as the double inverted pendulum model [310] to replace the IPM in our method and enhance the ability to represent physical interactions. Furthermore, we intend to explore comprehensive models that can predict reactive and active motions accurately. One possible idea based on our existing pipeline is to extract informative features from observed motions, which can guide the forward movement estimation in a chosen latent physics space. The latent IPM space in our method might fail and more general physics models can be necessary under this circumstance due to the complexity of considering various reactive and active motions simultaneously.

Appendix A

Supplementary Material for Chapter 3

A.1 Additional Experiments

A.1.1 Generalization to Unseen Scenarios

We use the collision rate to evaluate prediction/simulation plausibility. We first elaborate on the definition of the collision rate and then show more experimental results. Provided there are N agents in a scene, we consider their collision rates during a period of time such as 4.8 seconds which is widely used to evaluate trajectory predictions/simulation [73, 74, 83]. We count one collision if the minimum distance between two agents is smaller than 2r at any time, where r is the radius of a disc representing an agent. The maximum possible number of collisions is N(N-1)/2. The final collision rate is defined as:

$$R_{col} = \frac{M}{N(N-1)/2},\tag{A.1}$$

where M is the number of collisions.

We show more results on the scene, coupa0, with different numbers of agents. We chose this scene because it is a relatively large space and can theoretically accommodate many people. The highest number of people simultaneously in the environment in the original data is merely 11. Therefore, this is a good scene to show how different methods can generalize to higher densities when learning from low-density data.

In each experiment, the agents are randomly initialized with different initial positions, initial velocities, and goals near the boundary of the scene, which is sufficient for our method to simulate. Therefore, we use NSP-SFM to simulate trajectories of 30 seconds (t = 0 to 29) at FPS = 10 for all agents. We sample three intervals out of every trajectory, from t = 0 to 8, t = 4 to 12, and t = 8 to 16, where the density in the central area reaches the highest during t=8 to 16. For each interval (8 seconds long), we subsample at FPS = 2.5 to get 20 frames, where the first 8 frames are used as input for Y-net [83] and S-CSR [74]. The remaining 12 frames and the simulation results (12 frames) of Y-net and S-CSR are used to calculate the collision rate. Before simulation, all methods are trained on the training dataset of SDD under the same setting explained in Chapter 3.

The results are shown in Tab. A.1. We tested 50, 74, 100, 150, and 200 agents on the aforementioned three methods including ours. We can see that our method is always the best in the collision rate under different settings. Although its collision rate increases with the growth of the number of agents, our method is still the best compared

Table A.1: Collision rates of the generalization experiments on Coupa0. Results of Ynet, S-SCR, and NSP on 50, 74, 100, 150, and 200 agents are shown in corresponding tables, where (1), (2), and (3) denote three intervals for calculating the collision rate. (a) 50 Agents (b) 74 Agents

	()			
Methods	(1)	(2)	(3)	avg
Y-net	2.8%	2.9%	3.8%	3.2%
S-CSR	2.5%	1.7%	1.9%	2.0%
Ours	0.6%	0.6%	0.6%	0.6%

(6) 11 Hgombs							
Methods	(1)	(2)	(3)	avg			
Y-net	3.8%	4.8%	3.0%	3.9%			
S-CSR	0.9%	0.9%	1.5%	1.1%			
Ours	0.2%	0.3%	0.5%	0.3%			

(c)	100	Agents
-----	-----	--------

Methods	(1)	(2)	(3)	avg
Y-net	4.2%	5.2%	7.6%	5.7%
S-CSR	0.9%	1.1%	0.8%	0.9%
Ours	0.3%	0.7%	0.4%	0.5%

(d)	150	Agents

Methods	(1)	(2)	(3)	avg
Y-net	4.9%	4.0%	3.4%	4.1%
S-CSR	0.6%	1.0%	1.7%	1.1%
Ours	0.2%	0.5%	1.0%	0.6%

(e) 200 Agents							
Methods	(1)	(2)	(3)	avg			
Y-net	5.9%	4.0%	3.5%	4.5%			
S-CSR	0.6%	0.9%	2.0%	1.2%			
Ours	0.2%	0.5%	0.8%	0.5%			

with the baselines, and our simulation results are more plausible. In addition, we also plot the relation between the collision rate (and the number of collisions) and the agent number ranging from 50 to 200 in Fig. A.1. Y-net is worse than S-CSR and our model. In addition, although the trends of our model and S-CSR are similar, the number of collisions of S-CSR increases faster than our model. Finally, some visualization results can be found in Fig. A.2. Here, every green disc has a radius of 7.5 pixels. When two green discs intersect, they collide with each other. Fig. A.2 demonstrates that our model has better performance in avoiding collisions than Y-net and S-CSR.

A.1.2 Interpretability of Prediction

More examples of interpretability are shown in Fig. A.3. In Fig. A.3 (1)-(2), we show the influence of different three forces, F_{goal} , F_{col} and F_{env} , on the whole trajectory of an agent. In Fig. A.3 (3)-(4), we choose two consecutive moments of one agent for analysis. In Fig. A.3 (1), instead of directly aiming for the goal, the agent suddenly



Figure A.1: The collision rate and the number of collisions against the number of agents are shown in (a) and (b), respectively. Both horizontal axes represent the number of agents from 50 to 200. The vertical axes in (a) and (b) represent the collision rate and the number of collisions, respectively.

turns (at the intersection between red and green dots) due to the incoming agents (the three blue dots under the green dots). The result is a result of major influence from F_{goal} and F_{col} . Similarly, the agent in Fig. A.3 (2) did not need to avoid other agents but still did not directly walk towards the goal, because of F_{env} from the grass. In Fig. A.3 (3)-(4), we show the detailed analysis of forces at two consecutive time steps of the same agent, where F_{env} is from the lawn which is a 'weakly repulsive area'. More examples where randomness is captured by our model are shown in Fig. A.4.

A.1.3 Ablation Experiments

We conduct more ablation experiments to further validate our design decisions and explore the effect of components of our model. The ablation studies on the network architectures focus on the Goal-Network and the Collision-Network. The main variants are with/without LSTM to show the importance of the temporal modeling for learning τ and k_{nj} , and replacing the MLPs with simple two-layer MLPs. Tab. A.2 shows the results on SDD. We can see that the temporal modeling and the original MLPs make our model achieve the best performance. To understand the role of each component in our model, we take the social force model (SFM) as the baseline and incrementally add components from our model. The results are shown in Tab. A.3. We tried our best to

Figure A.2: The visualization results of generalization to 74, 100, 150, and 200 agents on coupa0 are shown in (a), (b), (c), and (d), respectively. For each experimental setting, visualization results of our model, Y-net, and S-CSR are at the same frame. We amplify the area of the red ellipse to boxes with yellow borders for better visualization performance.

manually find good parameter values: $\tau = 0.5$, $k_{nj}=25/50$, and $k_{env}=65$. We adopted the same way with our model to sample destinations for SFM. Then we only learn τ and k_{nj} . At last, the result of the full model without CVAE is given. The performance is better when more components are added.

A.2 Details of NSP-SFM

In this section, we elaborate on the details of the Goal Sampling Network (GSN) and the conditional Variational Autoencoder (CVAE) in our NSP-SFM.



Figure A.3: Examples of interpretability. Red dots are observed, and green dots are our predictions. Blue dots in (1), (3), and (4) are other pedestrians at time steps 7, 16, and 17 respectively. We show the influence of all forces, F_{goal} , F_{col} , and F_{env} , on the whole trajectory in (1) and (2). We display a detailed analysis of three forces at two consecutive time steps of the same agent in (3) and (4), where F_{goal} , F_{col} , and F_{env} are shown as yellow, light blue, and black arrows, respectively.

A.2.1 Goal Sampling Network

The main components of the GSN are two U-nets [311] as illustrated in Fig. A.5. We first feed the scene image I to a U-net, U_{seg} , to get its corresponding environment pixelwise segmentation with dimension of $H * W * K_c$. H and W are the height and width of I, and K_c is the number of classes for segmentation. The segmentation maps are byproducts of the GSN from [83]. NSP can use manually annotated or automatically segmented environment maps to calculate F_{env} , but using segmentation maps from the GSN is more efficient. Then the past trajectories $\{p^t\}_{t=0}^M$ are converted into M+1



Figure A.4: Motion randomness is captured by our model. Red dots are observed, green dots are our prediction and black dots are the ground truth.

Table A.2: Ablation experiments on network architecture. Goal-Network and the Collision-Network possess the same architecture under each experimental setup.

ADE	Two layers MLP	Full MLP
w/o LSTM	6.83	6.61
with LSTM	6.66	6.52

Table A.3: Ablation experiments on SDD. Different components from our model are added incrementally.

$k_{nj}=25$	hand-tuned	learned τ and k_{nj}	NSP-SFM
ADE	8.32	6.53	6.52
FDE	10.97	10.61	10.61
$k_{nj}=50$	hand-tuned	learned τ and k_{nj}	NSP-SFM
ADE	7.54	6.53	6.52
FDE	10.81	10.61	10.61

trajectory heatmaps by:

$$Hm(t, i, j) = 2 \frac{\|(i, j) - p^t\|}{\max_{(x, y) \in I} \|(x, y) - p^t\|},$$
(A.2)

where (i, j) is the pixel coordinate on the heatmap and (x, y) is the pixel coordinate on the scene image I. Then, we concatenate these trajectory heatmaps and the segmentation map to get the input with the dimension of $H * W * (K_c + M + 1)$ for the network U_{goal} . U_{goal} will output a non-parametric probability distribution map, \tilde{D}_{goal} , with dimensions H * W. Every pixel in \tilde{D}_{goal} has a corresponding probability value between 0 and 1, and their sum is equal to 1. Details of these two U-nets can be found in [83]. We train the GSN by minimizing the KL divergence between predicted \tilde{D}_{goal} and its ground truth D_{goal} . We assume that D_{goal} is a discrete Gaussian distribution with a mean at the position of the ground-truth goal and a hyperparameter variance σ_{goal} . During testing, instead of picking the position with the highest probability, we adopt the test-time sampling trick (TTST) introduced by [83] to sample goals for better performance.



Figure A.5: Model architecture of the goal sampling network. The detailed network architecture of two U-nets, U_{seg} and U_{goal} , can be found in [83].



Figure A.6: The architecture of the CVAE, where \bar{p}^{t+1} is the intermediate prediction out of our force model and $\alpha^{t+1} = p^{t+1} - \bar{p}^{t+1}$. Encoder E_{bias} , E_{past} , E_{latent} , and decoder D_{latent} are all MLP networks with dimensions indicated in the square brackets. Red connections are only used in the training phase.

A.2.2 Conditional Variational Autoencoder

We model the dynamics stochasticity for each agent individually by using a CVAE as illustrated in Fig. A.6. Red connections are only used in the training phase. Given an agent p^t and his/her destination, a deterministic prediction \bar{p}^{t+1} without dynamics stochasticity is first calculated from F_{goal} , F_{col} and F_{env} and a semi-implicit scheme. During training time, we use the corresponding ground truth p^{t+1} to calculate the error $\alpha^{t+1} = p^{t+1} - \bar{p}^{t+1}$, and feed α^{t+1} into an encoder E_{bias} to get the feature f_{bias} . The brief history $(p^{t-7}, \ldots, p^{t-1}, p^t)$ is encoded as f_{past} by using an encoder E_{past} . We concatenate f_{bias} with f_{past} and encode it using a latent encoder to yield the parameters

HyperPara	ETH	Hotel	UNIV	ZARA1	ZARA2	SDD
a_1	1	1	1	1	1	1
b_1	0.1	0.1	2.2	1.6	1.4	0.4
a_2	50	50	50	50	50	100
b_2	0	0	0	0	0	0
ω	$\pi/3$	$\pi/3$	$\pi/3$	$\pi/3$	$\pi/3$	$\pi/3$
r_{col}	75	75	75	75	75	100
r _{env}	50	50	50	75	75	50
σ_{goal}	4	4	4	4	4	4
σ_{latent}	1.3	1.3	1.3	1.3	1.3	1.3
λ_{weak}	N/A	N/A	N/A	N/A	N/A	0.2

Table A.4: Hyperparameters for all six datasets.

 (μ, σ) of the Gaussian distribution of the latent variable Z. We sample Z, concatenate it with f_{past} for history information, and decode using the decoder D_{latent} to acquire our guess for stochasticity $\tilde{\alpha}^{t+1}$. Finally, the estimated stochasticity will be added to the deterministic prediction \bar{p}^{t+1} to get our final prediction \tilde{p}^{t+1} . During testing time, the ground truth p^{t+1} is unavailable. Therefore, we sample the latent variable Z from a Gaussian distribution $N(0, \sigma_{latent}I)$ where σ_{latent} is a hyperparameter. We concatenate the sampled Z and f_{past} to decode directly using the learned decoder D_{latent} to get the estimate of stochasticity $\tilde{\alpha}^{t+1}$. We can produce the final prediction \tilde{p}^{t+1} using the same way as the training phase. Encoders E_{bias} , E_{past} , E_{latent} and the decoder D_{latent} are all multi-layer perceptrons (MLP) with dimensions indicated in the square brackets in Fig. A.6.

A.3 Implementation Details

We use ADAM as the optimizer to train the Goal-Network, Collision-Network and F_{env} with a learning rate between 3×10^{-5} and 3×10^{-4} , and to train the CVAE with a learning rate between 3×10^{-6} and 3×10^{-5} . When we train the CVAE of our model, the training data is scaled by 0.005 to balance reconstruction error and the KL divergence in l_{cvae} . The hyperparameter λ in l_{cvae} is set to 1. Concrete structures of

all sub-networks are shown in Fig. A.6.

We list all hyperparameters of our model in Tab. A.4. We segment scene images into two classes and three classes on ETH/UCY and SDD, respectively. The two classes on ETH/UCY are 'walkable area' and 'unwalkable area'. Three classes on SDD include 'walkable area', 'unwalkable area', and 'weakly repulsive area' that some people tend to avoid such as lawns. The calculation of F_{env} on ETH/UCY has been introduced in Chapter 3. On SDD, we calculate the position of the obstacle p_{obs} and the position of the weak obstacle p_{w-obs} (i.e. in the weakly repulsive area) by averaging pixels that are classified as 'unwalkable area' and 'weak repulsive area' respectively. Then, the F_{env} consists of two repulsive forces from p_{obs} and p_{w-obs} as shown in Eq. A.3, where the parameter k_{env} is shared and an additional hyperparameter λ_{weak} is introduced for p_{w-obs} :

$$F_{env} = \frac{k_{env}}{\|p_n^t - p_{obs}\|} \left(\frac{p_n^t - p_{obs}}{\|p_n^t - p_{obs}\|}\right) + \frac{\lambda_{weak}k_{env}}{\|p_n^t - p_{w-obs}\|} \left(\frac{p_n^t - p_{w-obs}}{\|p_n^t - p_{w-obs}\|}\right).$$
(A.3)

Appendix B

Supplementary Material for Chapter 4


Figure B.1: Explainability of behaviors. Red, blue, and green dots denote observations, neighbors, and predictions, respectively. Forces \mathbf{F}_a , \mathbf{F}_n and \mathbf{F}_s for the chosen positions are shown as yellow, light blue, and black arrows, respectively. The explicit distributions predicted by our model are shown as heatmaps. Yellow to blue in the heatmaps indicates that the probability density is from high to low.

B.1 Additional Explainability Experiments

We show more examples of explainability in Fig. B.1, where only the particle system of our model is employed. Our model can provide detailed explanations and analysis for the predictions. In Fig. B.1 (a), the pedestrian at the chosen position moves under a strong actuation force and two weak forces from the neighbor and the scene. Additionally, we can acquire efficiently how certain our model is for the predicted forces according to their predicted explicit distributions. We note that our model is more certain about the predicted actuation force compared with the other two predicted forces in Fig. B.1 (a) since the distribution of the actuation force has the smallest variances. Only factors \mathbf{D}_a and \mathbf{D}_n are shown in Fig. B.1 (b) because the scene factor is too weak to be visualized. Forces sampled from the distributions of these two factors determine the prediction of the next position. Further, we note that the factor \mathbf{D}_n has a more concentrated distribution compared with the factor \mathbf{D}_a , which means that our model is more certain in the estimation of the interaction between people.

B.2 Network Details

Actuation Block. Its network architecture is shown as Fig. B.2 left. $[\mathbf{c}^t, \dot{\mathbf{c}}^t]$ is fed into a Long Short Term Memory (LSTM) network to output the feature \mathbf{f}_a^t . The encoder E_a encodes the goal $\mathbf{c}^{t_p+t_f}$ into $\mathbf{f}_a^{t_p+t_f}$. The concatenated feature $[\mathbf{f}_a^t, \mathbf{f}_a^{t_p+t_f}]$ is fed into the



Figure B.2: Network architectures of the actuation block and the neighbor block.



Figure B.3: The CVAE architecture. H denotes the number of history frames.

decoder D_a to estimate the distribution parameters $[\mu_a^t, \log \sigma_a^t]$. The decoder outputs the $\log \sigma_a^t$ to ensure that each dimension of the output has the same range. E_a and D_a are multi-layer perceptrons (MLPs) and have architectures [2, 64, 256, 16] and [32, 512, 256, 512, 2], respectively. We use an LSTM with 256 dimensions. The embedding layer and output layer before and after the LSTM cell have dimensions 64 and 16, respectively.

Neighbour Block. Its network architecture is shown as Fig. B.2 right. For any neighbour $j \in \Omega_i^t$, the LSTM and the encoder E_n encode $[\mathbf{c}_i^t, \dot{\mathbf{c}}_i^t]$ and $[\mathbf{c}_j^t, \dot{\mathbf{c}}_j^t]$ into features \mathbf{f}_i^t and \mathbf{f}_j^t , respectively. The concatenated feature $[\mathbf{f}_i^t, \mathbf{f}_j^t]$ is fed into the decoder D_n to estimate distribution parameters. The neighbor block is the same as the actuation block in terms of the network architecture and dimensions except that the input layer of E_n has 4 dimensions.

Model Uncertainty. The E_{res} , E_{past} , E_{laten} and D_{latent} in the conditional variational autoencoder (CVAE) as shown in Fig. B.3 are MLPs and have architectures [2, 8, 16, 16], [18, 512, 256, 16], [32, 8, 50, 32], and [32, 1024, 512, 1024, 2], respectively. The hyperparameters H and σ_{latent} are set as 8 and 1.3, respectively.

B.3 Implementation Details

The optimizer ADAM is used for training all networks. The learning rates for training the particle system are between 3×10^{-6} and 3×10^{-5} , while the learning rates for training CVAE are between 1×10^{-7} and 1×10^{-6} . When we train the CVAE, the hyperparameter λ is 1. We empirically scale the data with a coefficient of 0.005 to balance the reconstruction error and the KL divergence. We follow tricks in [80] to determine neighbors and obstacles in our model.

Appendix C

Supplementary Material for Chapter 5 $\,$

A2M Ŕ Â Ŕ Ć ý Ű -ACTO Ŕ Ŕ ļ Ŕ R 1 MDA Å Ĥ RMI Ŕ í PhyV ĥ 1 4 siMLI EqM Y X Out GT

Figure C.1: Visual comparison on pushes with different magnitudes. Left: strong, Middle: weak, Right: medium.

Code and pre-processed data are available: https://github.com/realcrane/Human-Motion-Prediction-under-Unexpected-Perturbation.

C.1 Additional Experiments

C.1.1 Single-person Results

More Comparison

We give the full visual comparison between our methods and the 7 baseline methods in Fig. C.1. Overall, our method achieves the best results and is the closest to the ground truth. Comparatively, MDM and EqM predict visually unreasonable motions with strange poses. A2M, ACTOR, SiMLPe generate visually reasonable snapshots but low-quality motions as well as inaccurate prediction. RMD and PhyVae give more aesthetically pleasing results, but again not high-quality motions and accurate prediction.

A more detailed numerical comparison is presented in Fig. C.2. Note that the visual comparison is to some extent consistent with the numerical results. MDM and EqM



Figure C.2: Perturbations with different magnitudes in single-person.

give the worst quality metrics on MBLE and FSE (MBLE for MDM is 0 since it uses a joint-angle-based representation hence no bone-length error). Across all metrics, our model is the best.

Looking closely, at motion tracking errors, MDM and EqM are not the worst. It suggests that motion tracking metrics and quality metrics evaluate two aspects of the results. This is indeed the case. RMDDiffuse and PhyVae give good motion quality among the baselines and their quality metrics are also good but not necessarily the best. Meanwhile, their tracking metrics are also good but not necessarily the best. A2M can achieve better FSE and sometimes better MBLE than RMDDiffuse and PhyVae, but its motion quality is generally lower. This suggests that there might be a trade-off between motion quality and prediction accuracy among the baselines. But our method achieves the best on both kinds of metrics.

More Generalization

We show more generalization experiments in the single-person scenario. We mainly test out-of-distribution push forces in magnitude, timing, and duration. In magnitude, we fix the duration of the force to be the same as a strong push but use an extra

REPERSENCE REPERSENCE TO THE

Figure C.3: The Generalization to an extra strong push. There are three motions (yellow, blue, and green). Blue is the ground truth of a strong push. Yellow is our prediction on the same strong push. Green is an extra strong push.



Figure C.4: The Generalization to a multi-push scenario. Yellow is the predicted motion under a strong push as in Fig. C.3. Green is the extra strong push in Fig. C.3. Red is the three-phase push motion. The numbers indicate the frames.

stronger push that is 37.36% higher than the strongest push in the dataset. The result is shown in Fig. C.3. We can see that the motion pushed by the extra strong push is significantly different from the ground truth and the predicted motion under the strong push. The motion contains earlier foot movements since the initial push is extra strong and it generates a much larger acceleration in the beginning. Also, the upper body is stiffer and has less swing because the balance recovery under an extra strong push tends to require the body to stiffen quickly to prevent the character from falling down and recover balance ultimately.

Furthermore, we generalize the push in timing and duration. This time, we apply multiple pushes at different times, as opposed to one push in the beginning as in the data. Note there are not multiple pushes in the data at all. We first apply a weak push, then a medium push at the 15th frame, and finally a strong push at the 50th frame. We show the visual results of this three-phase push in Fig. C.4. One can see that the motion is initially slow and sluggish due to the weak initial push, then gradually intensifies as more pushes are introduced. Under the weak push, the character does not even start to make a step, then it starts to take steps after the medium push at the 15th frame. In the end, large strides need to be made, after the strong push at the 50th frame, to recover balance and counter-balance the accumulated accelerations.

In theory, our model can generalize to other scenarios like slippery surfaces as the friction is learned (Sec 5.3.1 in Chapter 5). Overall, our model can generalize to out-

Method	MPJPE	hipADE	hipFDE
PPR	0.623	0.455	0.602
PHC	0.488	0.409	0.662
Ours	0.097	0.086	0.171

Table C.1: Comparison with full-body physics-based baselines.

of-distribution physical disturbances in magnitude, timing, and duration.

Comparison with Full-body Physics-based Models

In literature, there is work that also employs body physics for motion imitation under full-body physics-based models [286, 287, 312, 313]. Although they turn fully/partially observed/user-specified motions into physically valid ones which is different from our task, they could be adapted to our new task. However, they are still intrinsically incapable of learning force interactions in multi-people. So, we could only compare the performance on single-person. To this end, we adapted the latest physics-based models PPR [313] and PHC [287] and compared them with our model in the single-person scenario. Results are shown in Tab. C.1. MBLE and FSE are not considered because these two baselines are joint-angle-based and simulation-based. Overall, Our model still performs best on all metrics. PPR and PHC can generate physically valid motions, but these motions are not necessarily accurate predictions.

Compared with the full-body physics models, the Inverted Pendulum Model (IPM) is not fine-grained but has the right granularity for our problem. IPM is a compact yet flexible representation and therefore has been widely used for articulated bodies such as bipedal/quadrupedal robots including humanoids [281], especially in balance recovery. Further, simplification is crucial for scalable interaction learning. A full-body model contains 50-100 degrees of freedom (Dofs). Learning from a 4-people scene then involves 200-400 Dofs plus Dofs for interaction forces, which will be extremely unscalable/slow as the learning requires many iterations of forward simulation (for many time steps) and backward propagation. Also, the Dofs will quickly explode in simulation when the number of people increases, *e.g.* our 13-person example. In comparison, one IPM only has 4 Dofs and is much more scalable for both learning and simulation, whose representational capacity has been proven [279, 280]. Also, even with a small model,

our model does not overfit, as evidenced by the superb testing results.

Another advantage of using IPMs instead of full-body physics models is the interaction modeling. We learn interaction forces as potential-energy-based forces between two IPMs, which is flexible and easily learnable. This is because contact information (position, duration, *etc.*) is not in the data. Therefore the physics model cannot involve accurate contact modeling even with full-body models, especially when the contact can be frequently established and destroyed in push propagation.

C.1.2 Multi-people Results

More Comparison

We provide the complete visual comparison between our model with the 4 baseline methods in Fig. C.5. Overall, our model obtains the best motion quality and is the closest to the ground truth. DuMMF cannot produce natural movements. JRFormer tends to predict merely subtle motions deviating from the ground truth. MRT and TBIFormer suffer from severe intersections between people for the group formation that is a line. MRT generates serious foot skating for the group formation where people stand in two lines, while TBIFormer performs as well as our model in this formation. Note that all baselines here are given much more information than our model. See the uploaded video on YouTube for a more intuitive comparison.

Detailed numerical comparison can be found in Fig. C.6. DuMMF employs the jointangle-based representation, resulting in a zero MBLE. Overall, our model achieves or is close to the best performance across metrics and perturbation levels. For all tracking error metrics, our model is much better than baseline methods. This is because only our model can predict the onset and duration of interaction accurately. In motion quality metrics, our model outperforms all baselines across three perturbation levels, meaning that our motion has the best quality.

More Generalisation

Other than the 13-people in a diamond formation shown in Chapter 5, we conduct further generalization experiments. We employ a formation with ten people standing closely in a line, to test whether a strong push can be propagated. Since we explicitly set the distances between people to be very small, we expect a strong push on the first

MRT								
DuMMF								MP
TBIFormer								
JRFormer								
Ours								
GT					and the second s			A
MRT		All	REL	Mil	胤訂	演員		
DuMMF			<u>& </u>	NAL L	RU II		£1£	₽↓ħ
TBIFormer		All	ART	Alt	新社			NA M
JRFormer		MALL	All	All	ALL	All	Â	All
Ours		Mall.	All	MILL	翻目	瀰儿		
GT		ARE	ANI	織门	All	M	ANA	MAX
MRT								
DuMMF					i	A	A A	
TBIFormer	•							
JRFormer								
Ours								
GT								

Figure C.5: Visual comparison on pushes with different magnitudes and group formations. Top: medium, Middle: strong, Bottom: weak.



Figure C.6: Perturbations with different magnitudes in multi-people.

Figure C.7: Generalization on ten people in a line. The first person is pushed by a strong force and we can simulate the force propagation. The number denotes which frame.

person to be propagated through people all the way to the front, like what is commonly observed in high-density crowds.

Our prediction results are shown in Fig. C.7. Note this type of scenario is totally out of distribution, in terms of the number of people and the formation (a much longer line). One can see a clear push propagation starting at the back of the line and then being carried over all the way to the front. This shows not only are the individual motions captured by the model, but the interaction as well as the interaction propagation are also predicted well.

Furthermore, looking closely at the predicted interaction forces between people, we find that the core reason for this push to be propagated, instead of dying down, is that it is intensified by interactions. This is also observed in high-density crowds where a small push can be intensified to cause the "butterfly effects" and finally even cause crushes. By turning the parameters in basic forces in the interaction module, it is possible to let the propagation dissipate more quickly. Overall, this shows the great flexibility of our model in capturing complex interactions and interaction propagation. This flexibility

Method	MPJPE	hipADE	hipFDE	MBLE	FSE
A2M	0.403	0.386	0.730	0.019	0.200
ACTOR	0.362	0.338	0.591	0.020	0.434
MDM	0.500	0.424	0.686	0	2.567
RMDiffuse	0.228	0.202	0.299	0.011	0.790
PhyVae	0.260	0.249	0.460	0.009	0.170
siMLPe	0.130	0.117	0.226	0.006	0.182
EqMotion	0.296	0.270	0.543	0.064	1.552
Ours	0.097	0.086	0.171	0.002	0.131
siMLPe_25%	0.203	0.189	0.411	0.009	0.650
Ours_25%	0.207	0.190	0.267	0.009	0.211

Table C.2: Metrics in complete (top) and 25% (bottom) training data for single-person.

could be crucial in crowd simulation in high-density crowds where potential crushes can happen.

C.1.3 Data Efficiency

One core reason for our LDP design is the lack of data. So it is essential to test the data efficiency. Although the original data is already much smaller than existing datasets for human motion prediction, we further reduce the data to 25% of its original size and repeat the training on single-person and multi-people scenarios.

As shown in Tab. C.2, our model trained on 25% training data still outperforms all baselines trained on 100% data, except for siMLPe in the single-person scenario. Therefore, we also trained siMLPe on 25% training data and evaluated it on all metrics for comparison. siMLPe achieves good performance and is slightly better than our model on MPJPE and hipADE on 25% training data, while our model performs much better on hipFDE and FSE. It's notable that we gave much more information to siMLPe.

One possible reason for the good performance of siMLPe might be its lightweight, as aimed for by its authors. So we also compare the model sizes in Tab. C.3. It is clear that the lightweight is not the only reason, as other baselines which are smaller than ours cannot achieve good results. We speculate that expressivity especially explicit physics

Table C.3: Model size in single-person (left) and multi-people (right). The unit is M (million). Ours_S means our model for single-person which excludes the differential interaction model. Ours_M is our complete model.

Method	A2M	ACTOR	MDM	RMD	PhyVae	$_{\rm siMLPe}$	EqMotion	Ours_S	MRT	DuMMF	TBIF	JRF	Ours_M
Parameters	0.45	14.78	18.10	40.96	2.72	0.02	0.64	2.67	6.98	6.54	10.26	3.70	2.94

Table C.4: Metrics in complete (top) and 25% (bottom) training data for multi-people.

Method	MPJPE	hipADE	hipFDE	MBLE	FSE
MRT	0.162	0.140	0.282	0.010	0.256
DuMMF	0.312	0.285	0.480	0	3.194
TBIFormer	0.204	0.177	0.305	0.010	0.234
JRFormer	0.181	0.152	0.260	0.012	0.932
Ours	0.106	0.092	0.218	0.003	0.069
$Ours_25\%$	0.139	0.115	0.270	0.011	0.117

is the key. Further, even siMLPe can achieve good numerical results, its predicted motions are of lower visual quality. More importantly, extending siMLPe to multipeople scenarios is challenging as it cannot learn interactions at all.

Next, we suspect that reduced training data brings more difficulty to the multipeople motion prediction. The results prove us correct, shown in Tab. C.4. Our model is still better than all baselines trained on 100% data, even though the training data for our model is reduced to 25%.

The high data efficiency of our model is mainly because the physics model embedded in our model has a low number of learnable parameters, but largely dictates the motion trend. The governing differential equation (Eq. 5.3 in Chapter 5) restricts the overall input-output mapping of the whole model and therefore it requires little data to learn. Similar phenomena have been observed in other differentiable physics research [276, 277].

Table C.5: Dataset details. There are four subjects and four group settings in singleperson and multi-people respectively in the dataset. Three push magnitudes (weak, medium, and strong) are used.

Single	Wk	Med	Str	Tot	Group	Wk	Med	Str	Tot
S1	3	4	3	10	G1	4	4	4	12
S2	5	3	4	12	G2	6	6	4	16
S3	5	4	4	13	G3	10	9	6	25
S4	3	4	3	10	G4	0	10	0	10
Tot	16	15	14	45	Tot	20	29	14	63

C.2 Additional Experiment Details

C.2.1 Dataset Details

The new dataset, FZJ Push, records human motions under expected physical perturbations. There are 45 single-person motions and 63 multi-people motions in the dataset. In both scenarios, repeated experiments were conducted on applying unexpected physical pushes with varying magnitude onto a person. In the single-person scenario, this is simply recording reactive motions to push and balance recovery; in multi-people scenario, one person is pushed and this person pushes other people to recover balance so that the push can be propagated among several people.

After discarding redundant frames such as those in waiting, we have 3104 frames and 5614 frames in the single-person and multi-people scenarios, respectively. All pushes are recorded via a pressure sensor Xsensor LX210:50.50.05 on the punching bag. The punching bag was moved manually by the same operator in all experiments. In addition, the pushes are labeled as small, medium, and strong.

In the single-person scenario, the dataset involves 4 subjects (S1-S4). Tab. C.5 left shows the number of experiments on each subject under different push magnitudes. We select randomly about 30% of the data to construct the test set for every subject, while the remaining data is used for training. Finally, the test set and train set have 13 motions and 32 motions, respectively. In the multi-people scenario, the dataset involves 4 group settings. G1 has four people standing in two lines, shown in Fig. C.8. G2 has a formation where four people are in a line. G3 and G4 contain 5 people in



Figure C.8: FZJ Push [18]. The blue agent was pushed by the punch bag and then he pushed other people.

a line. We give the number of motions in every group setting under different push magnitudes in Tab. C.5 right. We randomly select approximately 20% of the data in each group setting for the testing set, while the remaining is for training. Eventually, the test set and train set have 14 motions and 49 motions, respectively. Since our new task aims to predict remaining future motions given the initial poses and motions have different lengths, we have different numbers of prediction frames for diverse motions. The minimum and maximum numbers of prediction frames are 38 (42) and 105 (132), respectively, in the single-person (multi-people) scenario. All motions are recorded in FPS = 60. Therefore, the time horizon of predicted motions is between 0.63 (0.7) seconds and 1.75 (2.2) seconds in the single-person (multi-people) scenario.

C.2.2 Metrics

We adopt five metrics commonly used for evaluating motion prediction accuracy and quality as follows. MPJPE (Mean Per Joint Position Error), hipADE (Average Displacement Error at the hip), and hipFDE (Final Displacement Error at the hip) are metrics measuring the tracking errors. MPJPE is the most widely used metric in human motion prediction to evaluate prediction accuracy on every joint. hipADE focuses on the main motion trend, while hipFDE pays attention to the final position of the hip. Moreover, hipADE and hipFDE are strongly relevant to the hip joint which corresponds to the point mass in our IPM. In addition, another two metrics MBLE (Mean Bone Length Error) and FSE (Foot Skating Error) are used to measure the motion quality. We adopt these two metrics to check if our model can produce reasonable poses and motions. • **MPJPE**: Mean Per Joint Position Error (MPJPE) is the average l_2 distance between predicted positions of joints and their ground truth:

MPJPE =
$$\frac{1}{TNJ} \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{j=1}^{J} ||X_t^n[j] - \hat{X}_t^n[j]||_2,$$
 (C.1)

where $X_t^n[j]$ is the position of the *j*th joint of the *n*th person at frame *t* and $\hat{X}_t^n[j]$ is its prediction. This metric is used most widely to measure the 3D pose errors.

• hipADE: Average Displacement Error at the hip (hipADE) is the average l_2 distance between predicted positions of hip joints and their ground truth:

hipADE =
$$\frac{1}{TN} \sum_{t=1}^{T} \sum_{n=1}^{N} ||h_t^n - \hat{h}_t^n||_2,$$
 (C.2)

where h_t^n is the hip position of the *n*th person at frame *t* and \hat{h}_t^n is its prediction. This metric focuses on global errors.

• **hipFDE**: Final Displacement Error at the hip (hipFDE) is the average l_2 distance between predicted positions of the hip joints at the last frame in each motion sequence and their ground truth:

hipFDE =
$$\frac{1}{N} \sum_{n=1}^{N} ||h_T^n - \hat{h}_T^n||_2.$$
 (C.3)

• **MBLE**: Mean Bone Length Error (MBLE) is the average l_1 distance between lengths of predicted bones and their ground truth:

MBLE =
$$\frac{1}{TNB} \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{b=1}^{B} \left| X_t^{nb} - \hat{X}_t^{nb} \right|,$$
 (C.4)

where X_t^{nb} is the length of *b*th bone of the *n*th person at frame t and \hat{X}_t^{nb} is the corresponding prediction.

• **FSE**: Foot Skating Error (FSE) is the average of weighted foot velocities for all feet with a height h within a threshold H. The weighted velocity is $v_f(2-2^{h/H})$.

C.2.3 Baseline Adaptation

The task proposed in Chapter 5 is new, so there is no similar work to our best knowledge. For comparison, we adapted 11 state-of-the-art baseline methods in the most relevant areas: motion forecasting, motion generation, and motion synthesis. One selection criterion is the availability of the code, to ensure their original implementation is used.

Specifically, we choose A2M [299], ACTOR [268], MDM [269], RMDiffuse [270], PhyVae [273], EqMotion [41], siMLPe [42], PPR [313] and PHC [287] for the singleperson scenario, and MRT [169], DuMMF [44], TBIFormer [16] and JRFormer [43] for the multi-people scenario. We try our best to keep the best performance of these baselines when adapting. The adaptation details are as follows:

- A2M. Action2Motion (A2M) is the first work to generate human motions given an action type. We use the push magnitudes (weak, medium, and strong) as the action labels (0, 1, 2). The initial pose is applied to kick-start the generation instead of a blank pose filled with 0 in the testing phase.
- ACTOR. Action-conditioned Transformer VAE (ACTOR) is another action-tomotion method following A2M. Similar to A2M, the push magnitudes are regarded as the action labels (0, 1, 2). In addition, the initial pose is given when decoding.
- MDM. Motion Diffusion Model (MDM) is one of the first papers employing diffusion models in motion generation. This model can achieve great performance for text-to-motion and action-to-motion. We replace the text input in MDM with the input forces under the text-to-motion setting. Then, the part corresponding to the initial frame in \hat{x}_0 is overwritten at each iteration as the MDM does in its motion editing. This is to minimize the change for adaptation. MDM handles motion editing, where if we fix the first frame, the task setting is almost the same as our task. Specifically, motion editing with the initial frame fixed is equivalent to letting the model generate the whole motion given the input signal.
- **RMDiffuse.** Retrieval-augmented Motion Diffusion model (RMDiffuse) is the state-of-the-art model in motion generation. We adopt its test-to-motion setting and replace the original text input with the input force. Similar to MDM, the part corresponding to the initial frame in \hat{x}_0 is overwritten at each iteration during evaluation to ensure the information of the first frame is given.
- **PhyVae**. Physics-based VAE (PhyVae) is the state-of-the-art motion synthesis model. At each step, PhyVae predicts current action a_t given the current input signal g_t and current state s_t . Then a_t is fed into a pre-trained network (that can

be regarded as a decoder) to predict the next state s_{t+1} . The input force at each time step t is regarded as the input signal to synthesize the motion.

- siMLPe. This model is a lightweight network based on MLPs but can achieve state-of-the-art performance in single-person motion prediction. For this forecasting approach, it requires as input M frames and predicts N frames. To ensure the comparison is as fair as possible, we provide as input complete information on the input force including magnitude and duration. Specifically, we set M to the maximum duration of the input forces in the single-person scenario. Then, we keep the original ratio between the past and the future frames in the long-term setting in the paper to set N as M/2. M and N values are shown in Tab. C.6. During testing, given the first M frames, we predict autoregressively to get the complete motion.
- **PPR and PHC.** These two baselines are state-of-the-art physics-based character animation methods that deal with perturbations. PPR and PHC can synthesize physically valid motions given reference motions. However, reference motions are unavailable during prediction in our new task. Therefore, following the setup in PHC, we use the adapted MDM to generate the reference motions during the test phase. Then these two baselines can generate motions based on the reference motions generated from the adapted MDM.
- EqMotion, MRT, DuMMF, TBIFormer, JRFormer. These models fall into human motion forecasting. They have a similar adaptation to that in siM-LPe, as they have similar input/output requirements. Details of their settings of input/output frames are shown in Tab. C.6. EqMotion is the state-of-the-art motion forecasting model for single-person. MRT is a classical multi-people motion prediction method. DuMMF, TBIFormer, and JRFormer are state-of-the-art multi-people motion prediction models.

C.2.4 Additional Details of Ablation Study

Here, we provide more details of the ablation study in Chapter 5. We conducted the ablation study to evaluate the effectiveness of two important components in our model: the Differentiable IPM and the Skeleton Restoration Model. We have four

Method	siMLPe	EqMotion	MRT	DuMMF	TBIFormer	JRFormer
Original Past	50	25	15	10	15	15
Original Future	25	25	45	25	45	45
Adaptation Past	12	12	20	20	20	20
Adaptation Future	6	12	60	50	60	60

Table C.6: Adaptation for motion prediction methods. 12 and 20 are the maximum duration of the input forces in the single-person and multi-people scenarios, respectively.

combinations: with/without IPM, and Full (full-body restoration) / Low-up (first lower body then upper body).

Our complete model is with IPM and uses a Low-up setting. Without IPM, it means that we only use the Skeleton Restoration Model to predict the next frame, while the two samplers (Upper-sampler and Lower-sampler) have to be dropped as they require the IPM state as input. Therefore, to sample the latent space of the Conditional Variational Autoencoder (CVAE), we sample the latent variable from a standard Normal distribution during the evaluation phase. The Full/Low-up setting is only within the Skeleton Restoration Model. In Full, we use a CVAE to generate full-body poses directly. Using the current frame as a condition, we sample the latent variable in the latent space. Then both are fed into the decoder to generate the next frame. In Low-up, we have two CVAEs and we generate the next frame in exactly the same way as in the Full setting, except that we first generate the lower body and then the upper body.

C.3 Additional Details of Methodology

C.3.1 Differentiable Inverted Pendulum Model

Given I_0 and I_0 , we can simulate the IPM motion in time by solving Eq. C.5 repeatedly:

$$M(I_t, l_t)\ddot{I}_t + C(I_t, \dot{I}_t, l_t) + G(I_t, l_t) = F_t^{net},$$
(C.5)

where $M \in \mathbb{R}^{4 \times 4}$, $C \in \mathbb{R}^{4 \times 1}$ and $G \in \mathbb{R}^{4 \times 1}$ are the inertia matrix, the Centrifugal/Coriolis matrix, and the external force such as gravity:

$$M_{t} = \begin{bmatrix} m_{c} + m_{p} & 0 & m_{p}l_{t}c_{\theta_{t}} & 0 \\ 0 & m_{c} + m_{p} & m_{p}l_{t}s_{\theta_{t}}s_{\phi_{t}} & -m_{p}l_{t}c_{\theta_{t}}c_{\phi_{t}} \\ m_{p}l_{t}c_{\theta_{t}} & m_{p}l_{t}s_{\theta_{t}}s_{\phi_{t}} & m_{p}l_{t}^{2} & 0 \\ 0 & -m_{p}l_{t}c_{\theta_{t}}c_{\phi_{t}} & 0 & m_{p}l_{t}^{2}c_{\theta_{t}}^{2} \end{bmatrix},$$

$$C_{t} = \begin{bmatrix} -m_{p}l_{t}s_{\theta_{t}}\dot{\theta}_{t}^{2} \\ m_{p}l_{t}(2s_{\theta_{t}}c_{\phi_{t}}\dot{\theta}_{t}\dot{\phi}_{t} + c_{\theta_{t}}s_{\phi_{t}}(\dot{\theta}_{t}^{2} + \dot{\phi}_{t}^{2})) \\ m_{p}l_{t}^{2}s_{\theta_{t}}c_{\phi_{t}}\dot{\phi}_{t}^{2} \\ -2m_{p}l_{t}^{2}s_{\theta_{t}}c_{\theta_{t}}\dot{\theta}_{t}\dot{\phi}_{t} \end{bmatrix} G_{t} = \begin{bmatrix} 0 \\ 0 \\ -m_{p}gl_{t}s_{\theta_{t}}c_{\phi_{t}} \\ -m_{p}gl_{t}c_{\theta_{t}}s_{\phi_{t}} \end{bmatrix}.$$
(C.6)

Here, m_c and m_p are the mass of the cart and the pendulum, respectively. c_{θ_t} and s_{θ_t} denote $\cos \theta_t$ and $\sin \theta_t$, while c_{ϕ_t} and s_{ϕ_t} represent $\cos \phi_t$ and $\sin \phi_t$. We set m_c and m_p as 0.1*M* and 0.9*M* respectively where *M* is the total mass of a person. Unlike the standard IPM, we allow the rod length to change with time. Given the net force $F_t^{net} \in \mathbb{R}^4$ and the rod length l_t , we can solve Eq. C.5 for the next state I_{t+1} via a semi-implicit scheme:

$$\dot{I}_{t+1} = \dot{I}_t + \Delta t \ddot{I}_t, \quad I_{t+1} = I_t + \Delta t \dot{I}_{t+1},$$
 (C.7)

where Δt is the time step. We have elaborated on the prediction of F_t^{net} and l_t in Chapter 5. Then we have the following equation:

$$I_T - I_0 = \int_0^T \dot{I}_t dt = \int_0^T \int M_t^{-1} (F_t^{net} - C_t - G_t) dt dt,$$
(C.8)

given the initial condition I_0 and I_0 and the final station I_T . The prediction of F_t^{net} is based on the neural networks and other differentiable operations such as PD control and repulsive potential energy. The prediction of the rod length l_t is from a neural network. Finally, the semi-implicit scheme for updating I_t only includes simple differentiable arithmetic. Therefore, our complete IPM is differentiable for both single-person and multi-people scenarios.

Single-person Prediction. The hyper-parameters K_p and K_d in the PD control are [30, 30, 1500, 1500] and [4, 4, 200, 200], respectively. We use an LSTM with the size 256 to predict $F_t^{self-nn}$. The MLP predicting the rod length has hidden size [128, 128].

$\theta_j \\ \theta_n$ Pos	Zero	Neg	ϕ_j ϕ_n	Pos	Zero	Neg
Pos 1/-1	0/-1	0/-1	Pos	-1/1	-1/0	-1/0
Zero 1/0	0/0	0/-1	Zero	0/1	0/0	-1/0
Neg 1/0	1/0	1/-1	Neg	0/1	0/1	-1/1

Table C.7: Basic interaction force on angles. X/X is BE/BA.

Differential Interaction Model. if $|r_{t,nj}| < r_{neigh}$, the *j*th person is the neighbor of the *n*th person at time *t i.e.* $j \in \Omega_{t,n}$, where $r_{neigh} = 0.5$. We use an MLP with 2 hidden layers [512, 512] to predict $F_{t,nj}^{inta-nn}$. The hyperparameters *u* and σ in the repulsive potential energy function for calculating F_{nj}^{bs-xy} are 150 and 0.5, respectively. Then, we elaborate the $F_{nj}^{bs-\theta\phi} = [F_{nj}^{bs-\theta}, F_{nj}^{bs-\phi}]^{\mathbf{T}}$. We give details for $F_{nj}^{bs-\theta}$, where the same principle also applies to $F_{nj}^{bs-\phi}$. The magnitude of $F_{nj}^{bs-\theta}$ is a constant $k_{\theta} = 100$ $(k_{\phi} = 50)$, while its direction is based on the θ_n and θ_j of IPM states of *n* and *j*. We categorize θ into three groups: positive, zero, and negative. For two IPMs, this produces a total of 9 possible situations. Then we need to decide their relative position. Taking the *n*th person as the person in interest, if its relative position with respect to a neighbor *j* along the x-axis is positive *i.e.* $x_{nj} = x_n - x_j > 0$, we label it as BE (before), otherwise BA (back). We show the directions of the force for all 9 possible situations in Tab. C.7, where 1 denotes the interaction force is positive, 0 denotes no interaction forces, and -1 denotes the negative direction.

C.3.2 Skeleton Restoration Model

Lower Body Restoration. We follow [272] to construct the CVAE-Lower as shown in Fig. C.9. The encoder is an MLP with two hidden layers of 256 dimensions, with an ELU layer following each hidden layer. The dimension of the latent variable z is 64. A mixture-of-expert architecture is employed for the decoder, including 4 expert networks and a gating network. The input to the gating network and the expert networks are the latent variable z combined with the current lower-body pose X_t^l , while the output of the expert network is the next pose X_{t+1}^l .

Similar to the encoder, the gating network is an MLP with two 64D hidden layers followed by ELU activations. Each expert network has the same structure as the



Figure C.9: The architecture of the CVAE-Lower. During training, the current lowerbody pose X_t^l as the condition, and the next lower-body pose X_{t+1}^l are fed into the encoder to predict the distribution of the latent variable z. Then the decoder predicts the next pose \hat{X}_{t+1}^l from the sampled variable z and the condition. The red connection is only used in training. During inference, we use the lower sampler to sample the latent variable z to predict motion.

encoder except for the input layer and the output layer.

During testing, we use the Lower Sampler to sample the latent variable z given the current lower-body pose X_t^l and the predicted IPM state I_{t+1} . The Lower sampler has the same structure as that of the encoder except for the input layer.

Upper Body Restoration. The CVAE-Upper has the same architecture as the CVAE-Lower except for the condition X_t^u and \hat{X}_{t+1}^l as shown in Fig. C.10. Similarly, the upper sampler has the same structure as the encoder of CVAE-Upper except for the input layer. Although the upper body is not explicitly physically constrained, it is implicitly constrained by the IPM motion which is physically based.

State Representation. In the skeleton restoration model, we adopt pose representations [272, 314] for the full-body pose. Specifically, we use a vector including positions, rotations, and velocities to represent the pose X_t . X_t^l and X_t^u take the corresponding lower or upper part in the X_t . Furthermore, we use a 15D vector $[x_t, y_t, \theta_t, \phi_t, e_t, l_t, \dot{x}_t, \dot{y}_t, \dot{\theta}_t, \dot{\phi}_t, \dot{e}_t]$ for I_t and input the vector into the sampler, where e_t is the position of the end of the rod corresponding to the hip joint and l_t is the rod length.



Figure C.10: The architecture of the CVAE-Upper. The condition X_t^u and \hat{X}_{t+1}^l , together with the next pose X_{t+1}^u are fed into the encoder to predict the distribution of the latent variable z. Then the decoder predicts the next pose \hat{X}_{t+1}^u from the sampled variable z and the condition. The red connection is only used in training. During inference, we use the upper sampler to sample the latent variable z to predict motion.

C.3.3 Training

There are several components in our model. An end-to-end training but could lead to suboptimal local minima. Therefore, we employ pre-training to initialize individual components and also use auxiliary losses in addition to the main loss introduced in Chapter 5.

We train the IPM first. Then, we train the CVAE-Lower. Next, we train the lower sampler network based on the trained CVAE-Lower. Similarly, we train the CVAE-Upper first then the upper sampler network.

We train the differentiable IPM model by using the 0-order and 1-order information as shown in Eq. C.9, where λ is a weight parameter. We minimize the angular velocity $\dot{\phi}$ instead of penalizing its l_1 norm as we do in other dimensions. This is because the angular velocity should always be smooth when recovering balances so that smoothing leads to better results than minimizing the l_1 norm.

$$L_{ipm} = \frac{1}{T} \sum_{t=1}^{T} \{ |\hat{x}_t - x_t| + |\hat{y}_t - y_t| + |\hat{\theta}_t - \theta_t| + |\hat{\phi}_t - \phi_t| + |\hat{x}_t - \dot{x}_t| + |\hat{y}_t - \dot{y}_t| + |\hat{\theta}_t - \dot{\theta}_t| + \lambda |\hat{\phi}_t| \}.$$
(C.9)

We follow [272] to train the CVAE-Lower in our skeleton restoration model. Then we train the Lower sampler network based on the trained CVAE-Lower. The encoder of the CVAE-Lower and the lower Sampler both output the Gaussian distribution parameters $[\mu, \sigma]$ for the latent variables z. We train the lower sampler by using the loss function in Eq. C.10 to let the outputs of the Lower Sampler be close to those of the encoder, and we ensure that the restored poses have low FSE.

$$L_{skel} = \|\hat{z}_{\mu} - z_{\mu}\|^2 + \|\hat{z}_{\sigma} - z_{\sigma}\|^2 + \text{FSE}(\hat{X}_t^l, X_t^l).$$
(C.10)

We train the CVAE-Upper and the Upper sampler network in the same way except that the FSE in the loss function Eq. C.10 is ignored.

After initialization, the whole network can be trained as a whole. We use the Adam optimizer for all training. The learning rates for training the differentiable IPM and two samplers are 3e-4 and 1e-4, respectively. When training the CVAEs, a linear schedule is used to adjust the learning rate from 1e-4 to 1e-7, and we set the weight of the KL loss as 0.005 to encourage high reconstruction quality. The whole training takes about 15 hours on a single GeForce RTX 2080 Ti, but can be automated. The inference takes approximately 0.19 sec/frame in our 13-person experiment.

BIBLIOGRAPHY

- Nanda Wijermans. Understanding crowd behaviour. PhD thesis, University of Groningen, 2011.
- [2] H Gayathri, PM Aparna, and Ashish Verma. A review of studies on understanding crowd dynamics in the context of crowd safety in mass religious gatherings. *International journal of disaster risk reduction*, 25:82–91, 2017.
- [3] Chongyang Wang, Shunjiang Ni, and Wenguo Weng. Modeling human domino process based on interactions among individuals for understanding crowd disasters. *Physica A: Statistical Mechanics and its Applications*, 531:121781, 2019.
- [4] Milad Haghani and Majid Sarvi. Crowd behaviour and motion: Empirical methods. Transportation research part B: methodological, 107:253–294, 2018.
- [5] Donatella Darsena, Giacinto Gelli, Ivan Iudice, and Francesco Verde. Sensing technologies for crowd management, adaptation, and information dissemination in public transportation systems: A review. *IEEE Sensors Journal*, 23(1):68–87, 2022.
- [6] Céline Loscos, Franco Tecchia, and Yiorgos Chrysanthou. Real-time shadows for animated crowds in virtual cities. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 85–92, 2001.
- [7] Marcelo Kallmann, Jean-Sébastien Monzani, Angela Caicedo, and Daniel Thalmann. Ace: A platform for the real time simulation of virtual human agents. In Computer Animation and Simulation 2000: Proceedings of the Eurographics Workshop in Interlaken, Switzerland, August 21–22, 2000, pages 73–84. Springer, 2000.

- [8] Beibei Zhan, Dorothy N Monekosso, Paolo Remagnino, Sergio A Velastin, and Li-Qun Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19: 345–357, 2008.
- [9] Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, and Pierre-Alain Muller. Recent trends in crowd analysis: A review. *Machine Learning with Applications*, 4:100023, 2021.
- [10] Feixiang He. Advancing High-Fidelity Crowd Simulation: From Behavior to Environment Layout. PhD thesis, University of Leeds, 2024.
- [11] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.
- [12] Roger L Hughes. A continuum theory for the flow of pedestrians. Transportation Research Part B: Methodological, 36(6):507–535, 2002.
- [13] Roger L Hughes. The flow of human crowds. Annual review of fluid mechanics, 35(1):169–182, 2003.
- [14] Okan Arikan, David A Forsyth, and James F O'Brien. Pushing people around. In Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 59–66, 2005.
- [15] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.
- [16] Xiaogang Peng, Siyuan Mao, and Zizhao Wu. Trajectory-aware body interaction transformer for multi-person pose forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17121–17130, 2023.
- [17] Renhao Huang, Hao Xue, Maurice Pagnucco, Flora Salim, and Yang Song. Multimodal trajectory prediction: A survey. arXiv preprint arXiv:2302.10463, 2023.
- [18] Sina Feldmann and Juliane Adrian. Forward propagation of a push through a row of people. Safety science, 164:106173, 2023.

- [19] P. D. G. K. Still. Crowd safety and crowd risk analysis.
- [20] Mingbi Zhao, Jinghui Zhong, and Wentong Cai. A role-dependent data-driven approach for high-density crowd behavior modeling. ACM Transactions on Modeling and Computer Simulation (TOMACS), 28(4):1–25, 2018.
- [21] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In European conference on computer vision, pages 549–565. Springer, 2016.
- [22] picture. https://www.flickr.com/photos/nationaalarchief/5453358304/sizes/ o/.
- [23] Shanwen Yang, Tianrui Li, Xun Gong, Bo Peng, and Jie Hu. A review on crowd simulation and modeling. *Graphical Models*, 111:101081, 2020.
- [24] Abhinav Golas, Rahul Narain, and Ming C Lin. Continuum modeling of crowd turbulence. *Physical review E*, 90(4):042816, 2014.
- [25] Arianna Bottinelli and Jesse L Silverberg. Can high-density human collective motion be forecasted by spatiotemporal fluctuations? arXiv preprint arXiv:1809.07875, 2018.
- [26] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2255–2264, 2018.
- [27] Renhe Jiang, Xuan Song, Dou Huang, Xiaoya Song, Tianqi Xia, Zekun Cai, Zhaonan Wang, Kyoung-Sook Kim, and Ryosuke Shibasaki. Deepurbanevent: A system for predicting citywide crowd dynamics at big events. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 2114–2122, 2019.
- [28] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. The International Journal of Robotics Research, 39(8):895–935, 2020.

- [29] Tsubasa Hirakawa, Takayoshi Yamashita, Toru Tamaki, and Hironobu Fujiyoshi. Survey on vision-based path prediction. In Distributed, Ambient and Pervasive Interactions: Technologies and Contexts: 6th International Conference, DAPI 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Proceedings, Part II 6, pages 48–64. Springer, 2018.
- [30] Przemysław A Lasota, Terrence Fong, Julie A Shah, et al. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics*, 5(4): 261–349, 2017.
- [31] Yuchao Su, Jie Du, Yuanman Li, Xia Li, Rongqin Liang, Zhongyun Hua, and Jiantao Zhou. Trajectory forecasting based on prior-aware directed graph convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):16773–16785, 2022.
- [32] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7386–7400, 2021.
- [33] Jur van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE International Conference on Robotics and Automation, 2008.
- [34] Nicolas Schneider and Dariu M Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In german conference on pattern recognition, pages 174–183. Springer, 2013.
- [35] Jiashi Gao, Xinming Shi, and James JQ Yu. Social-dualcvae: Multimodal trajectory forecasting based on social interactions pattern aware and dual conditional variational auto-encoder. arXiv preprint arXiv:2202.03954, 2022.
- [36] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. Advances in neural information processing systems, 31, 2018.
- [37] Patrick Kidger. On neural differential equations. *arXiv preprint arXiv:2202.02435*, 2022.

- [38] Masha Itkina and Mykel Kochenderfer. Interpretable self-aware neural networks for robust trajectory prediction. In *Conference on Robot Learning*, pages 606–617. PMLR, 2023.
- [39] Jinghong Wang, Jinhua Sun, and Siuming Lo. Randomness in the evacuation route selection of large-scale crowds under emergencies. *Applied Mathematical Modelling*, 39(18):5693–5706, 2015.
- [40] Venkat Nemani, Luca Biggio, Xun Huan, Zhen Hu, Olga Fink, Anh Tran, Yan Wang, Xiaoge Zhang, and Chao Hu. Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial. *Mechanical Systems and Signal Processing*, 205:110796, 2023.
- [41] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Equotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1410–1420, 2023.
- [42] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to mlp: A simple baseline for human motion prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications* of Computer Vision, pages 4809–4819, 2023.
- [43] Qingyao Xu, Weibo Mao, Jingze Gong, Chenxin Xu, Siheng Chen, Weidi Xie, Ya Zhang, and Yanfeng Wang. Joint-relation transformer for multi-person motion prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9816–9826, 2023.
- [44] Sirui Xu, Yu-Xiong Wang, and Liangyan Gui. Stochastic multi-person 3d motion forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.
- [45] Thomas Geijtenbeek and Nicolas Pronost. Interactive character animation using simulated physics: A state-of-the-art review. *Computer graphics forum*, 31(8): 2492–2515, 2012.
- [46] Changkun Chen, Tong Lu, Weibing Jiao, and Congling Shi. An extended model for crowd evacuation considering crowding and stampede damage under the in-

ternal crushing. *Physica A: Statistical Mechanics and its Applications*, page 129002, 2023.

- [47] Elizabeth T Hsiao-Wecksler. Biomechanical and age-related differences in balance recovery using the tether-release method. Journal of Electromyography and Kinesiology, 18(2):179–187, 2008.
- [48] Matthew A Brodie, Yoshiro Okubo, Daina L Sturnieks, and Stephen R Lord. Optimizing successful balance recovery from unexpected trips and slips. *Journal of Biomechanical Science and Engineering*, 13(4):17–00558, 2018.
- [49] Jos Elfring, René Van De Molengraft, and Maarten Steinbuch. Learning intentions for improved human motion prediction. *Robotics and Autonomous Systems*, 62(4):591–602, 2014.
- [50] Gonzalo Ferrer and Alberto Sanfeliu. Behavior estimation for a complete framework for human motion prediction in crowded environments. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 5940–5945. IEEE, 2014.
- [51] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *CVPR 2011*, pages 1345–1352. IEEE, 2011.
- [52] Laurene Fausett. Fundamentals of neural networks: architectures, algorithms, and applications, 1994.
- [53] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems, 2, 1989.
- [54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.
- [55] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. Advances in neural information processing systems, 28, 2015.

- [56] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.
- [57] 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.
- [58] Alex Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.
- [59] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772. PMLR, 2014.
- [60] Federico Bartoli, Giuseppe Lisanti, Lamberto Ballan, and Alberto Del Bimbo. Context-aware trajectory prediction. In 2018 24th International Conference on Pattern Recognition (ICPR), pages 1941–1946. IEEE, 2018.
- [61] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In 2018 IEEE international Conference on Robotics and Automation (ICRA), pages 4601–4607. IEEE, 2018.
- [62] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Srlstm: State refinement for lstm towards pedestrian trajectory prediction. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 12085–12094, 2019.
- [63] Hung Tran, Vuong Le, and Truyen Tran. Goal-driven long-term trajectory prediction. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 796–805, 2021.
- [64] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goaldriven networks for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):2716–2723, 2022.
- [65] Nishant Nikhil and Brendan Tran Morris. Convolutional neural network for trajectory prediction. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pages 0–0, 2018.

- [66] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14424–14432, 2020.
- [67] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8994–9003, 2021.
- [68] Chengxin Wang, Shaofeng Cai, and Gary Tan. Graphten: Spatio-temporal interaction modeling for human trajectory prediction. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 3450– 3459, 2021.
- [69] Simone Zamboni, Zekarias Tilahun Kefato, Sarunas Girdzijauskas, Christoffer Norén, and Laura Dal Col. Pedestrian trajectory prediction with convolutional neural networks. *Pattern Recognition*, 121:108252, 2022.
- [70] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019.
- [71] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. Advances in neural information processing systems, 32, 2019.
- [72] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2375–2384, 2019.
- [73] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, pages 759–776. Springer, 2020.

- [74] Hao Zhou, Dongchun Ren, Xu Yang, Mingyu Fan, and Hai Huang. Sliding sequential cvae with time variant socially-aware rethinking for trajectory prediction. arXiv preprint arXiv:2110.15016, 2021.
- [75] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17113–17122, 2022.
- [76] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5517– 5526, 2023.
- [77] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28,* 2020, Proceedings, Part XII 16, pages 507–523. Springer, 2020.
- [78] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agentaware transformers for socio-temporal multi-agent forecasting. In *Proceedings of* the IEEE/CVF International Conference on Computer Vision, pages 9813–9823, 2021.
- [79] Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. Trajectory unified transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9675–9684, 2023.
- [80] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *European conference on computer vision*, pages 376–394. Springer, 2022.
- [81] Zhaobin Mo, Yongjie Fu, and Xuan Di. Pi-neugode: Physics-informed graph neural ordinary differential equations for spatiotemporal trajectory prediction. In Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, pages 1418–1426, 2024.

- [82] Wei Xiang, YIN Haoteng, He Wang, and Xiaogang Jin. Socialcvae: Predicting pedestrian trajectory via interaction conditioned latents. In *Proceedings of the* AAAI Conference on Artificial Intelligence, volume 38, pages 6216–6224, 2024.
- [83] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 15233–15242, 2021.
- [84] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.
- [85] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Nanning Zheng, and Gang Hua. Social interpretable tree for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2235–2243, 2022.
- [86] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7144–7153, 2019.
- [87] Jinghai Duan, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Liushuai Shi, and Gang Hua. Complementary attention gated network for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 542–550, 2022.
- [88] Haonan Li, Xiaolan Wang, Xiao Su, and Yansong Wang. Improved gaussian mixture probabilistic model for pedestrian trajectory prediction of autonomous vehicle. *Recent Patents on Mechanical Engineering*, 17(1):65–75, 2024.
- [89] Anshul Nayak, Azim Eskandarian, and Zachary Doerzaph. Uncertainty estimation of pedestrian future trajectory using bayesian approximation. *IEEE Open* Journal of Intelligent Transportation Systems, 3:617–630, 2022.

- [90] Radford M Neal. Bayesian learning for neural networks, volume 118. Springer Science & Business Media, 2012.
- [91] Lingxue Zhu and Nikolay Laptev. Deep and confident prediction for time series at uber. In 2017 IEEE International Conference on Data Mining Workshops (ICDMW), pages 103–110. IEEE, 2017.
- [92] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. Advances in neural information processing systems, 29, 2016.
- [93] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. In 4th Workshop on Bayesian Deep Learning. bayesiandeeplearning. org, 2019.
- [94] Wouter Van Toll and Julien Pettré. Algorithms for microscopic crowd simulation: Advancements in the 2010s. In *Computer Graphics Forum*, volume 40, pages 731– 754. Wiley Online Library, 2021.
- [95] Dirk Helbing, Illés Farkas, and Tamas Vicsek. Simulating dynamical features of escape panic. *Nature*, 407(6803):487–490, 2000.
- [96] Dirk Helbing, Lubos Buzna, Anders Johansson, and Torsten Werner. Selforganized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation science*, 39(1):1–24, 2005.
- [97] Nuria Pelechano, Jan M Allbeck, and Norman I Badler. Controlling individual agents in high-density crowd simulation. In Symposium on Computer animation, pages 99–108. San Diego, 2007.
- [98] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. The international journal of robotics research, 17:760–772, 1998.
- [99] Jur Van den Berg, Ming Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE international conference on robotics and automation, pages 1928–1935. Ieee, 2008.

- [100] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics Research: The 14th International Symposium ISRR*, pages 3–19. Springer, 2011.
- [101] Jan Ondřej, Julien Pettré, Anne-Hélène Olivier, and Stéphane Donikian. A synthetic-vision based steering approach for crowd simulation. ACM Transactions on Graphics (TOG), 29:1–9, 2010.
- [102] Rowan Hughes, Jan Ondřej, and John Dingliana. Davis: density-adaptive synthetic-vision based steering for virtual crowds. In *Proceedings of the 8th ACM* SIGGRAPH Conference on Motion in Games, pages 79–84, 2015.
- [103] Teófilo Bezerra Dutra, Ricardo Marques, Joaquim B Cavalcante-Neto, Creto Augusto Vidal, and Julien Pettré. Gradient-based steering for vision-based crowd simulation algorithms. In *Computer graphics forum*, volume 36, pages 337–348. Wiley Online Library, 2017.
- [104] Ioannis Karamouzas, Nick Sohre, Rahul Narain, and Stephen J Guy. Implicit crowds: Optimization integrator for robust crowd simulation. ACM Transactions on Graphics (TOG), 36:1–13, 2017.
- [105] Tomer Weiss, Alan Litteneker, Chenfanfu Jiang, and Demetri Terzopoulos. Position-based multi-agent dynamics for real-time crowd simulation. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 1–2, 2017.
- [106] Tomer Weiss, Alan Litteneker, Chenfanfu Jiang, and Demetri Terzopoulos. Position-based real-time simulation of large crowds. *Computers & Graphics*, 78: 12–22, 2019.
- [107] Sachin Patil, Jur Van Den Berg, Sean Curtis, Ming C Lin, and Dinesh Manocha. Directing crowd simulations using navigation fields. *IEEE transactions on visu*alization and computer graphics, 17(2):244–254, 2010.
- [108] Adam Barnett, Hubert PH Shum, and Taku Komura. Coordinated crowd simulation with topological scene analysis. In *Computer Graphics Forum*, volume 35, pages 120–132. Wiley Online Library, 2016.
- [109] Guanghui Lu, Leiting Chen, and Weiping Luo. Real-time crowd simulation integrating potential fields and agent method. ACM Transactions on Modeling and Computer Simulation (TOMACS), 26(4):1–16, 2016.
- [110] Rahul Narain, Abhinav Golas, Sean Curtis, and Ming C Lin. Aggregate dynamics for dense crowd simulation. In ACM SIGGRAPH Asia 2009 papers, pages 1–8. 2009.
- [111] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. ACM transactions on graphics (TOG), 25(3):1160–1168, 2006.
- [112] Hao Jiang, Wenbin Xu, Tianlu Mao, Chunpeng Li, Shihong Xia, and Zhaoqi Wang. Continuum crowd simulation in complex environments. *Computers & Graphics*, 34(5):537–544, 2010.
- [113] Branislav Ulicny, Pablo de Heras Ciechomski, and Daniel Thalmann. Crowdbrush: interactive authoring of real-time crowd scenes. In *Proceedings of the* 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 243–252, 2004.
- [114] Kang Hoon Lee, Myung Geol Choi, and Jehee Lee. Motion patches: building blocks for virtual environments annotated with motion data. In ACM SIG-GRAPH 2006 Papers, pages 898–906. ACM New York, USA, 2006.
- [115] Joseph Henry, Hubert PH Shum, and Taku Komura. Interactive formation control in complex environments. *IEEE transactions on visualization and computer* graphics, 20(2):211–222, 2013.
- [116] Ioannis Karamouzas and Mark Overmars. Simulating and evaluating the local behavior of small pedestrian groups. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):394–406, 2011.
- [117] Liang He, Jia Pan, Sahil Narang, Wenping Wang, and Dinesh Manocha. Dynamic group behaviors for interactive crowd simulation. arXiv preprint arXiv:1602.03623, 2016.
- [118] Zhiguo Ren, Panayiotis Charalambous, Julien Bruneau, Qunsheng Peng, and Julien Pettré. Group modeling: A unified velocity-based approach. In *Computer Graphics Forum*, volume 36, pages 45–56. Wiley Online Library, 2017.

- [119] Takeshi Sakuma, Tomohiko Mukai, and Shigeru Kuriyama. Psychological model for animating crowded pedestrians. *Computer Animation and Virtual Worlds*, 16 (3-4):343–351, 2005.
- [120] Tibor Bosse, Rob Duell, Zulfiqar A Memon, Jan Treur, and C Natalie Van der Wal. Agent-based modeling of emotion contagion in groups. *Cognitive Computation*, 7:111–136, 2015.
- [121] Funda Durupmar, Uğur Güdükbay, Aytek Aman, and Norman I Badler. Psychological parameters for crowd simulation: From audiences to mobs. *IEEE transactions on visualization and computer graphics*, 22(9):2145–2159, 2015.
- [122] Panayiotis Charalambous and Yiorgos Chrysanthou. The pag crowd: A graph based approach for efficient data-driven crowd simulation. In *Computer Graphics Forum*, volume 33, pages 95–108. Wiley Online Library, 2014.
- [123] Zhenzhen Yao, Guijuan Zhang, Dianjie Lu, and Hong Liu. Learning crowd behavior from real data: A residual network method for crowd simulation. *Neuro*computing, 404:173–185, 2020.
- [124] Hongyi Chen, Jingtao Ding, Yong Li, Yue Wang, and Xiao-Ping Zhang. Social physics informed diffusion model for crowd simulation. In *Proceedings of the* AAAI Conference on Artificial Intelligence, volume 38, pages 474–482, 2024.
- [125] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. ACM Transactions on Graphics (ToG), 31(4):1–8, 2012.
- [126] Ting-Yun Wang, Tanvi Bhatt, Feng Yang, and Yi-Chung Pai. Adaptive control reduces trip-induced forward gait instability among young adults. *Journal of biomechanics*, 45(7):1169–1175, 2012.
- [127] Ingram A Murray and Garth R Johnson. A study of the external forces and moments at the shoulder and elbow while performing every day tasks. *Clinical biomechanics*, 19(6):586–594, 2004.
- [128] Lydia G Brough, Glenn K Klute, and Richard R Neptune. Biomechanical response to mediolateral foot-placement perturbations during walking. *Journal of Biomechanics*, 116:110213, 2021.

- [129] Stephanie A Ross, Nilima Nigam, and James M Wakeling. A modelling approach for exploring muscle dynamics during cyclic contractions. *PLoS computational biology*, 14(4):e1006123, 2018.
- [130] Michael Günther, Oliver Röhrle, Daniel FB Haeufle, and Syn Schmitt. Spreading out muscle mass within a hill-type model: A computer simulation study. *Computational and mathematical methods in medicine*, 2012(1):848630, 2012.
- [131] Friedl De Groote, Allison L Kinney, Anil V Rao, and Benjamin J Fregly. Evaluation of direct collocation optimal control problem formulations for solving the muscle redundancy problem. Annals of biomedical engineering, 44:2922–2936, 2016.
- [132] Lemaire K.K van Soest A.J., Casius L.J.R. Huxley-type cross-bridge models in largeish-scale musculoskeletal models; an evaluation of computational cost. *Journal of biomechanics*, 83:43–48, 2019.
- [133] Archibald Vivian Hill. The heat of shortening and the dynamic constants of muscle. Proceedings of the Royal Society of London. Series B-Biological Sciences, 126(843):136–195, 1938.
- [134] Andrew F Huxley. Muscle structure and theories of contraction. Progress in biophysics and biophysical chemistry, 7:255–318, 1957.
- [135] Leroy F Henderson. The statistics of crowd fluids. nature, 229(5284):381–383, 1971.
- [136] Dirk Helbing and Tamás Vicsek. Optimal self-organization. New journal of physics, 1(1):13, 1999.
- [137] Mohcine Chraibi, Armin Seyfried, and Andreas Schadschneider. Generalized centrifugal-force model for pedestrian dynamics. *Physical Review Eâ€"Statistical*, *Nonlinear, and Soft Matter Physics*, 82(4):046111, 2010.
- [138] Daniel R Parisi, Marcelo Gilman, and Herman Moldovan. A modification of the social force model can reproduce experimental data of pedestrian flows in normal conditions. *Physica A: Statistical Mechanics and its Applications*, 388(17):3600– 3608, 2009.

- [139] Minoru Fukui and Yoshihiro Ishibashi. Self-organized phase transitions in cellular automaton models for pedestrians. *Journal of the physical society of Japan*, 68 (8):2861–2863, 1999.
- [140] Masakuni Muramatsu, Tunemasa Irie, and Takashi Nagatani. Jamming transition in pedestrian counter flow. *Physica A: Statistical Mechanics and its Applications*, 267(3-4):487–498, 1999.
- [141] Yang Li, Maoyin Chen, Zhan Dou, Xiaoping Zheng, Yuan Cheng, and Ahmed Mebarki. A review of cellular automata models for crowd evacuation. *Physica A: Statistical Mechanics and its Applications*, 526:120752, 2019.
- [142] John Von Neumann. The general and logical theory of automata. In Systems research for behavioral science, pages 97–107. Routledge, 2017.
- [143] Wouter van Toll, Thomas Chatagnon, Cédric Braga, Barbara Solenthaler, and Julien Pettré. Sph crowds: Agent-based crowd simulation up to extreme densities using fluid dynamics. *Computers & Graphics*, 98:306–321, 2021.
- [144] 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.
- [145] Shuang Lu, Joseph Picone, and Seong Kong. Fingerspelling alphabet recognition using a two-level hidden markov model. In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), page 1, 2013.
- [146] Graham W Taylor, Geoffrey E Hinton, and Sam Roweis. Modeling human motion using binary latent variables. Advances in neural information processing systems, 19, 2006.
- [147] Eunsuk Chong and Frank Chongwoo Park. Movement prediction for a lower limb exoskeleton using a conditional restricted boltzmann machine. *Robotica*, 35(11): 2177–2200, 2017.
- [148] Jack Wang, Aaron Hertzmann, and David J Fleet. Gaussian process dynamical models. Advances in neural information processing systems, 18, 2005.

- [149] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007.
- [150] Xin Jin and Jia Guo. Regressive gaussian process latent variable model for fewframe human motion prediction. *IEICE TRANSACTIONS on Information and* Systems, 106(10):1621–1626, 2023.
- [151] 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.
- [152] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structuralrnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 5308–5317, 2016.
- [153] 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.
- [154] 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. IEEE, 2017.
- [155] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. arXiv preprint arXiv:1805.06485, 2018.
- [156] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom. Deep representation learning for human motion prediction and classification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6158–6166, 2017.
- [157] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *Proceedings of the IEEE conference* on computer vision and pattern recognition, pages 5226–5234, 2018.
- [158] 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.

- [159] 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.
- [160] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In Proceedings of the IEEE/CVF international conference on computer vision, pages 11467–11476, 2021.
- [161] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 1418–1427, 2018.
- [162] 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.
- [163] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16, pages 346–364. Springer, 2020.
- [164] German Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2317–2327, 2023.
- [165] Dong Wei, Huaijiang Sun, Bin Li, Jianfeng Lu, Weiqing Li, Xiaoning Sun, and Shengxiang Hu. Human joint kinematics diffusion-refinement for stochastic motion prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6110–6118, 2023.
- [166] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning progressive joint propagation for human motion prediction. In *Computer Vision–ECCV 2020:* 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16, pages 226–242. Springer, 2020.

- [167] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatiotemporal transformer for 3d human motion prediction. In 2021 International Conference on 3D Vision (3DV), pages 565–574. IEEE, 2021.
- [168] Hua Yu, Xuanzhe Fan, Yaqing Hou, Wenbin Pei, Hongwei Ge, Xin Yang, Dongsheng Zhou, Qiang Zhang, and Mengjie Zhang. Toward realistic 3d human motion prediction with a spatio-temporal cross-transformer approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(10):5707–5720, 2023.
- [169] Jiashun Wang, Huazhe Xu, Medhini Narasimhan, and Xiaolong Wang. Multiperson 3d motion prediction with multi-range transformers. Advances in Neural Information Processing Systems, 34:6036–6049, 2021.
- [170] Sirui Xu, Yu-Xiong Wang, and Liangyan Gui. Stochastic multi-person 3d motion forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.
- [171] Zheyan Zhang, Peter K. Jimack, and He Wang. MeshingNet3D: Efficient generation of adapted tetrahedral meshes for computational mechanics. Advances in Engineering Software, 157-158, July 2021.
- [172] Taylor A Howell, Simon Le Cleacâ€TMh, J Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable simulator for robotics. arXiv preprint arXiv:2203.00806, 9(2):4, 2022.
- [173] Elvis Nava, John Z Zhang, Mike Yan Michelis, Tao Du, Pingchuan Ma, Benjamin F Grewe, Wojciech Matusik, and Robert Kevin Katzschmann. Fast aquatic swimmer optimization with differentiable projective dynamics and neural network hydrodynamic models. In *International Conference on Machine Learning*, pages 16413–16427. PMLR, 2022.
- [174] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, pages 1–12, 2022.
- [175] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. Advances in neural information processing systems, 34:8780–8794, 2021.

- [176] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073, 2021.
- [177] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. Journal of Machine Learning Research, 23:1–33, 2022.
- [178] Ali Ramadhan, John Marshall, Andre Souza, Xin Kai Lee, Ulyana Piterbarg, Adeline Hillier, Gregory LeClaire Wagner, Christopher Rackauckas, Chris Hill, Jean-Michel Campin, et al. Capturing missing physics in climate model parameterizations using neural differential equations. arXiv preprint arXiv:2010.12559, 2020.
- [179] Jeehyun Hwang, Jeongwhan Choi, Hwangyong Choi, Kookjin Lee, Dongeun Lee, and Noseong Park. Climate modeling with neural diffusion equations. In 2021 IEEE International Conference on Data Mining (ICDM), pages 230–239. IEEE, 2021.
- [180] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. Advances in Neural Information Processing Systems, 33:6696–6707, 2020.
- [181] Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate time series forecasting with dynamic graph neural odes. *IEEE Transac*tions on Knowledge and Data Engineering, 35:9168–9180, 2022.
- [182] Keenon Werling, Dalton Omens, Jeongseok Lee, Ioannis Exarchos, and C. Karen Liu. Fast and feature-complete differentiable physics for articulated rigid bodies with contact. CoRR, abs/2103.16021, 2021.
- [183] Simon Le Cleac'h, Hong-Xing Yu, Michelle Guo, Taylor Howell, Ruohan Gao, Jiajun Wu, Zachary Manchester, and Mac Schwager. Differentiable physics simulation of dynamics-augmented neural objects. *IEEE Robotics and Automation Letters*, 8(5):2780–2787, 2023.
- [184] Junbang Liang, Ming Lin, and Vladlen Koltun. Differentiable cloth simulation

for inverse problems. Advances in Neural Information Processing Systems, 32, 2019.

- [185] Deshan Gong, Zhanxing Zhu, Andrew J Bulpitt, and He Wang. Fine-grained differentiable physics: A yarn-level model for fabrics. In *International Conference* on Learning Representations, 2022.
- [186] Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In international conference on machine learning, pages 2402–2411. PMLR, 2020.
- [187] Pedro M Milani, Julia Ling, and John K Eaton. Turbulent scalar flux in inclined jets in crossflow: counter gradient transport and deep learning modelling. *Journal* of Fluid Mechanics, 906:A27, 2021.
- [188] Mounir Bendali-Braham, Jonathan Weber, Germain Forestier, Lhassane Idoumghar, and Pierre-Alain Muller. Recent trends in crowd analysis: A review. *Machine Learning with Applications*, 4:100023, 2021.
- [189] Linbo Luo, Suiping Zhou, Wentong Cai, Malcolm Yoke Hean Low, Feng Tian, Yongwei Wang, Xiantao Xiao, and Dan Chen. Agent-based human behavior modeling for crowd simulation. *Computer Animation and Virtual Worlds*, 19, 2008.
- [190] Feixiang He, Yuanhang Xia, Xio Zhao, and He Wang. Informative scene decomposition for crowd analysis, comparison and simulation guidance. ACM Transaction on Graphics (TOG), 4(39), 2020.
- [191] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 6150–6156. IEEE, 2019.
- [192] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5725–5734, 2019.

- [193] Nachiket Deo and Mohan M Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv preprint arXiv:2001.00735, 2020.
- [194] Junwei Liang, Lu Jiang, and Alexander Hauptmann. Simaug: Learning robust representations from 3d simulation for pedestrian trajectory prediction in unseen cameras. arXiv preprint arXiv:2004.02022, 2, 2020.
- [195] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In European Conference on Computer Vision, pages 683–700. Springer, 2020.
- [196] Junwei Liang, Lu Jiang, Kevin Murphy, Ting Yu, and Alexander Hauptmann. The garden of forking paths: Towards multi-future trajectory prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10508–10518, 2020.
- [197] Tong Su, Yu Meng, and Yan Xu. Pedestrian trajectory prediction via spatial interaction transformer network. In 2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops), pages 154–159. IEEE, 2021.
- [198] Beihao Xia, Conghao Wong, Qinmu Peng, Wei Yuan, and Xinge You. Cscnet: Contextual semantic consistency network for trajectory prediction in crowded spaces. *Pattern Recognition*, page 108552, 2022.
- [199] Wouter Van Toll and Julien Pettré. Algorithms for Microscopic Crowd Simulation: Advancements in the 2010s. Computer Graphics Forum, 40(2), 2021.
- [200] D. Wolinski, S. J. Guy, A.-H. Olivier, M. Lin, D. Manocha, and J. PettrA©. Parameter estimation and comparative evaluation of crowd simulations. *Computer Graphics Forum*, 33(2):303–312, 2014.
- [201] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385, 2020.
- [202] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. arXiv preprint arXiv:1909.12077, 2019.

- [203] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *European conference* on computer vision, pages 452–465. Springer, 2010.
- [204] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.
- [205] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern* analysis and machine intelligence, 22(8):831–843, 2000.
- [206] David Ellis, Eric Sommerlade, and Ian Reid. Modelling pedestrian trajectory patterns with gaussian processes. In 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, pages 1229–1234. IEEE, 2009.
- [207] Xiaogang Wang, Keng Teck Ma, Gee-Wah Ng, and W Eric L Grimson. Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models. *International journal of computer vision*, 95(3):287–312, 2011.
- [208] Sujeong Kim, Aniket Bera, and Dinesh Manocha. Interactive crowd content generation and analysis using trajectory-level behavior learning. In 2015 IEEE International Symposium on Multimedia (ISM), pages 21–26. IEEE, 2015.
- [209] He Wang and Carol Oâ€TMSullivan. Globally continuous and non-markovian crowd activity analysis from videos. In European conference on computer vision, pages 527–544. Springer, 2016.
- [210] Rima Chaker, Zaher Al Aghbari, and Imran N Junejo. Social network model for crowd anomaly detection and localization. *Pattern Recognition*, 61:266–281, 2017.
- [211] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Random field topic model for semantic region analysis in crowded scenes from tracklets. In CVPR 2011, pages 3441–3448. IEEE, 2011.
- [212] He Wang, Jan Ondřej, and Carol O'Sullivan. Trending paths: A new semanticlevel metric for comparing simulated and real crowd data. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016.

- [213] Panayiotis Charalambous, Ioannis Karamouzas, Stephen J Guy, and Yiorgos Chrysanthou. A data-driven framework for visual crowd analysis. In *Computer Graphics Forum*, volume 33, pages 41–50. Wiley Online Library, 2014.
- [214] Bogdan Ilie Sighencea, RareÈ[™] Ion Stanciu, and Cătălin Daniel Căleanu. A review of deep learning-based methods for pedestrian trajectory prediction. Sensors, 21(22):7543, 2021.
- [215] Alessandro Antonucci, Gastone Pietro Rosati Papini, Luigi Palopoli, and Daniele Fontanelli. Generating reliable and efficient predictions of human motion: A promising encounter between physics and neural networks. arXiv preprint arXiv:2006.08429, 2020.
- [216] Sven Kreiss. Deep social force. arXiv preprint arXiv:2109.12081, 2021.
- [217] Sakif Hossain, Fatema T Johora, Jörg P Müller, Sven Hartmann, and Andreas Reinhardt. Sfmgnet: A physics-based neural network to predict pedestrian trajectories. arXiv, 2022.
- [218] Axel López, François Chaumette, Eric Marchand, and Julien Pettré. Character navigation in dynamic environments based on optical flow. In *Computer Graphics Forum*, volume 38, pages 181–192. Wiley Online Library, 2019.
- [219] Yijun Shen, Joseph Henry, He Wang, Edmond S. L. Ho, Taku Komura, and Hubert P. H. Shum. Data-Driven Crowd Motion Control With Multi-Touch Gestures. *Computer Graphics Forum*, 2018. ISSN 1467-8659. doi: 10.1111/cgf.13333.
- [220] Ioannis Karamouzas, Nick Sohre, Ran Hu, and Stephen J Guy. Crowd space: a predictive crowd analysis technique. ACM Transactions on Graphics (TOG), 37 (6):1–14, 2018.
- [221] Juan Wei, Wenjie Fan, Zhongyu Li, Yangyong Guo, Yuanyuan Fang, and Jierui Wang. Simulating crowd evacuation in a social force model with iterative extended state observer. *Journal of advanced transportation*, 2020, 2020.
- [222] Zhiqiang Wan, Xuemin Hu, Haibo He, and Yi Guo. A learning based approach for social force model parameter estimation. In *IJCNN*, pages 4058–4064. IEEE, 2017.

- [223] Kirill Zubov, Zoe McCarthy, Yingbo Ma, Francesco Calisto, Valerio Pagliarino, Simone Azeglio, Luca Bottero, Emmanuel Luján, Valentin Sulzer, Ashutosh Bharambe, Nand Vinchhi, Kaushik Balakrishnan, Devesh Upadhyay, and Christopher Rackauckas. Neuralpde: Automating physics-informed neural networks (pinns) with error approximations. CoRR, abs/2107.09443, 2021.
- [224] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Phys*ics, 3:422–440, 2021.
- [225] Zheyan Zhang, Yongxing Wang, Peter K. Jimack, and He Wang. Meshingnet: A new mesh generation method based on deep learning. In *Computational Science* - *ICCS 2020*, pages 186–198, Cham, 2020. Springer International Publishing.
- [226] Deshan Gong, Zhanxing Zhu, Bulpitt. Andrew, and He Wang. Fine-grained differentiable physics: a yarn-level model for fabrics. In *International Conference* on Learning Representations, 2022.
- [227] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [228] He Wang, Jan Ondřej, and Carol O'Sullivan. Path patterns: Analyzing and comparing real and simulated crowds. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2016, pages 49–57, 2016.
- [229] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. Advances in neural information processing systems, 28, 2015.
- [230] Luis Miguel Alvarez León, Julio Esclarín Monreal, Martin Lefébure, and Javier Sánchez. A pde model for computing the optical flow. 1999.
- [231] Sahil Narang, Andrew Best, Sean Curtis, and Dinesh Manocha. Generating pedestrian trajectories consistent with the fundamental diagram based on physiological and psychological factors. *PLoS one*, 10(4):e0117856, 2015.

- [232] Yuejiang Liu, Qi Yan, and Alexandre Alahi. Social nce: Contrastive learning of socially-aware motion representations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 15118–15129, 2021.
- [233] Weiliang Zeng, Peng Chen, Hideki Nakamura, and Miho Iryo-Asano. Application of social force model to pedestrian behavior analysis at signalized crosswalk. *Transportation research part C: emerging technologies*, 40:143–159, 2014.
- [234] Ari Virtanen. Energy-based pedestrian navigation. In Proc. 20th ITS World Congr., pages 1–9, 2013.
- [235] Peng Wang. Understanding social-force model in psychological principles of collective behavior. arXiv preprint arXiv:1605.05146, 2016.
- [236] Qingyang Tan, Zherong Pan, and Dinesh Manocha. Lcollision: Fast generation of collision-free human poses using learned non-penetration constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3913–3921, 2021.
- [237] Qingyang Tan, Zherong Pan, Breannan Smith, Takaaki Shiratori, and Dinesh Manocha. N-penetrate: Active learning of neural collision handler for complex 3d mesh deformations. In *International Conference on Machine Learning*, pages 21037–21049. PMLR, 2022.
- [238] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun. Learning motion patterns of persons for mobile service robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 3601–3606. IEEE, 2002.
- [239] S Thrun, W Burgard, and D Fox. Probabilistic robotics (intelligent robotics and autonomous agents series), ser. intelligent robotics and autonomous agents, 2005.
- [240] Peter Trautman, Jeremy Ma, Richard M Murray, and Andreas Krause. Robot navigation in dense human crowds: the case for cooperation. In 2013 IEEE international conference on robotics and automation, pages 2153–2160. IEEE, 2013.

- [241] Patrick Dendorfer, Aljosa Osep, and Laura Leal-Taixé. Goal-gan: Multimodal trajectory prediction based on goal position estimation. In Proceedings of the Asian Conference on Computer Vision, 2020.
- [242] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In European Conference on Computer Vision, pages 511–528. Springer, 2022.
- [243] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems, 30, 2017.
- [244] Mario Canche, Ranganath Krishnan, and Jean-Bernard Hayet. Epistemic uncertainty quantification in human trajectory prediction. In *ICT Applications for Smart Cities*, pages 37–56. Springer, 2022.
- [245] Jürgen Wiest, Matthias Höffken, Ulrich Kreßel, and Klaus Dietmayer. Probabilistic trajectory prediction with gaussian mixture models. In 2012 IEEE Intelligent vehicles symposium, pages 141–146. IEEE, 2012.
- [246] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9644–9653, 2023.
- [247] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
- [248] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.
- [249] Donghee Shin. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. International journal of humancomputer studies, 146:102551, 2021.
- [250] Xiao-Cheng Liao, Wei-Neng Chen, Xiao-Qi Guo, Jinghui Zhong, and Xiao-Min Hu. Crowd management through optimal layout of fences: An ant colony ap-

proach based on crowd simulation. *IEEE Transactions on Intelligent Transportation Systems*, 24(9):9137–9149, 2023.

- [251] Manon Prédhumeau, Lyuba Mancheva, Julie Dugdale, and Anne Spalanzani. Agent-based modeling for predicting pedestrian trajectories around an autonomous vehicle. Journal of Artificial Intelligence Research, 73:1385–1433, 2022.
- [252] Conghao Wong, Beihao Xia, Ziming Hong, Qinmu Peng, Wei Yuan, Qiong Cao, Yibo Yang, and Xinge You. View vertically: A hierarchical network for trajectory prediction via fourier spectrums. In *European Conference on Computer Vision*, pages 682–700. Springer, 2022.
- [253] Hao Zhou, Dongchun Ren, Xu Yang, Mingyu Fan, and Hai Huang. Csr: cascade conditional variational auto encoder with socially-aware regression for pedestrian trajectory prediction. *Pattern Recognition*, 133:109030, 2023.
- [254] Ke Guo, Wenxi Liu, and Jia Pan. End-to-end trajectory distribution prediction based on occupancy grid maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2242–2251, 2022.
- [255] Xiaotong Lin, Tianming Liang, Jianhuang Lai, and Jian-Fang Hu. Progressive pretext task learning for human trajectory prediction. arXiv preprint arXiv:2407.11588, 2024.
- [256] Duško Katić and Miomir Vukobratović. Survey of intelligent control techniques for humanoid robots. Journal of Intelligent and Robotic Systems, 37:117–141, 2003.
- [257] Cengiz Kahraman, Muhammet Deveci, Eda Boltürk, and Seda Türk. Fuzzy controlled humanoid robots: A literature review. *Robotics and Autonomous Systems*, 134:103643, 2020.
- [258] Libin Liu, KangKang Yin, Michiel Van de Panne, Tianjia Shao, and Weiwei Xu. Sampling-based contact-rich motion control. In ACM SIGGRAPH 2010 papers, pages 1–10. 2010.
- [259] Christian Ott, Maximo A Roa, and Gerd Hirzinger. Posture and balance control for biped robots based on contact force optimization. In 2011 11th IEEE-RAS International Conference on Humanoid Robots, pages 26–33. IEEE, 2011.

- [260] Libin Liu, KangKang Yin, and Baining Guo. Improving sampling-based motion control. In *Computer Graphics Forum*, volume 34, pages 415–423. Wiley Online Library, 2015.
- [261] Xiaolin Wei, Jianyuan Min, and Jinxiang Chai. Physically valid statistical models for human motion generation. ACM Transactions on Graphics (TOG), 30(3):1– 10, 2011.
- [262] Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. Physics-based human motion estimation and synthesis from videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11532–11541, 2021.
- [263] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. ACM Transactions on Graphics (TOG), 39(4):33–1, 2020.
- [264] He Wang, Edmond SL Ho, Hubert PH Shum, and Zhanxing Zhu. Spatio-temporal manifold learning for human motions via long-horizon modeling. *IEEE Transac*tions on Visualization and Computer Graphics (TVCG), 2019.
- [265] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 214–223, 2020.
- [266] Xiangjun Tang, He Wang, Bo Hu, Xu Gong, Ruifan Yi, Qilong Kou, and Xiaogang Jin. Real-time controllable motion transition for characters. ACM Transactions on Graphics (TOG), 41(4):1–10, 2022.
- [267] Xiangjun Tang, Linjun Wu, He Wang, Bo Hu, Xu Gong, Yuchen Liao, Songnan Li, Qilong Kou, and Xiaogang Jin. Rsmt: Real-time stylized motion transition for characters. In ACM SIGGRAPH, pages 1–10, 2023.
- [268] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10985–10995, 2021.

- [269] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. arXiv preprint arXiv:2209.14916, 2022.
- [270] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. arXiv preprint arXiv:2304.01116, 2023.
- [271] Wenheng Chen, He Wang, Yi Yuan, Tianjia Shao, and Kun Zhou. Dynamic future net: Diversified human motion generation. In ACM Multimedia (ACM MM), pages 2131–2139, 2020.
- [272] 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.
- [273] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. ACM Transactions on Graphics (TOG), 41(4): 1–12, 2022.
- [274] He Wang, Edmond SL Ho, and Taku Komura. An energy-driven motion planning method for two distant postures. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2014.
- [275] He Wang, Kirill A Sidorov, Peter Sandilands, and Taku Komura. Harmonic parameterization by electrostatics. ACM Transactions on Graphics (TOG), 2013.
- [276] Kun Wang, Mridul Aanjaneya, and Kostas Bekris. Sim2sim evaluation of a novel data-efficient differentiable physics engine for tensegrity robots. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1694–1701. IEEE, 2021.
- [277] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory forecasting with explainable behavioral uncertainty. arXiv preprint arXiv:2307.01817, 2023.
- [278] Reza Olfati-Saber. Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles. PhD thesis, Massachusetts Institute of Technology, 2001.

- [279] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180), volume 1, pages 239–246. IEEE, 2001.
- [280] Taesoo Kwon and Jessica K Hodgins. Momentum-mapped inverted pendulum models for controlling dynamic human motions. ACM Transactions on Graphics (TOG), 36(1):1–14, 2017.
- [281] Jaepyung Hwang, Jongmin Kim, Il Hong Suh, and Taesoo Kwon. Real-time locomotion controller using an inverted-pendulum-based abstract model. In *Computer Graphics Forum*, volume 37, pages 287–296. Wiley Online Library, 2018.
- [282] Siyuan Shen, Yin Yang, Tianjia Shao, He Wang, Chenfanfu Jiang, Lei Lan, and Kun Zhou. High-order differentiable autoencoder for nonlinear model reduction. ACM Transactions on Graphics (TOG), pages 1–15, 2021.
- [283] Steffen Wiewel, Moritz Becher, and Nils Thuerey. Latent space physics: Towards learning the temporal evolution of fluid flow. In *Computer graphics forum*, volume 38, pages 71–82. Wiley Online Library, 2019.
- [284] 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.
- [285] Chuanqi Zang, Mingtao Pei, and Yu Kong. Few-shot human motion prediction via learning novel motion dynamics. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pages 846–852, 2021.
- [286] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Transactions On Graphics (TOG), 37(4):1–14, 2018.
- [287] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid

control for real-time simulated avatars. In *Proceedings of the IEEE/CVF Inter*national Conference on Computer Vision, pages 10895–10904, 2023.

- [288] Siwei Chen, Xiao Ma, and Zhongwen Xu. Imitation learning via differentiable physics. arXiv preprint arXiv:2206.04873, 2022.
- [289] A. Iollo, M. Ferlauto, and L. Zannetti. An aerodynamic optimization method based on the inverse problem adjoint equations. *Journal of Computational Physics*, 173(1):87–115, 2001. ISSN 0021-9991. doi: https://doi.org/10. 1006/jcph.2001.6845. URL https://www.sciencedirect.com/science/article/ pii/S0021999101968457.
- [290] Y. Jarny, M.N. Ozisik, and J.P. Bardon. A general optimization method using adjoint equation for solving multidimensional inverse heat conduction. International Journal of Heat and Mass Transfer, 34(11):2911-2919, 1991. ISSN 0017-9310. doi: https://doi.org/10.1016/0017-9310(91)90251-9. URL https://www.sciencedirect.com/science/article/pii/0017931091902519.
- [291] Junbang Liang, Ming Lin, and Vladlen Koltun. Differentiable cloth simulation for inverse problems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/ 28f0b864598a1291557bed248a998d4e-Paper.pdf.
- [292] Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Josh Tenenbaum, and Chuang Gan. Dynamic visual reasoning by learning differentiable physics models from video and language. Advances in Neural Information Processing Systems, 34: 887–899, 2021.
- [293] Mazen Al Borno, Martin De Lasa, and Aaron Hertzmann. Trajectory optimization for full-body movements with complex contacts. *IEEE transactions on* visualization and computer graphics, 19(8):1405–1414, 2012.
- [294] Yujiang Xiang, Hyun-Joon Chung, Joo H Kim, Rajankumar Bhatt, Salam Rahmatalla, Jingzhou Yang, Timothy Marler, Jasbir S Arora, and Karim Abdel-Malek. Predictive dynamics: an optimization-based novel approach for human

motion simulation. *Structural and Multidisciplinary Optimization*, 41:465–479, 2010.

- [295] Arthur D Kuo. An optimal control model for analyzing human postural balance. IEEE transactions on biomedical engineering, 42(1):87–101, 1995.
- [296] Jacek Stodolka, Marian Golema, and Juliusz Migasiewicz. Balance maintenance in the upright body position: Analysis of autocorrelation. *Journal of Human Kinetics*, 50:45–52, 2016.
- [297] Dustin Li and Heike Vallery. Gyroscopic assistance for human balance. In 2012 12th IEEE International Workshop on Advanced Motion Control (AMC), pages 1–6. IEEE, 2012.
- [298] Petar Kormushev, Dragomir N Nenchev, Sylvain Calinon, and Darwin G Caldwell. Upper-body kinesthetic teaching of a free-standing humanoid robot. In 2011 IEEE International Conference on Robotics and Automation, pages 3970–3975. IEEE, 2011.
- [299] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In Proceedings of the 28th ACM International Conference on Multimedia, pages 2021–2029, 2020.
- [300] 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.
- [301] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European conference on computer* vision (ECCV), pages 601–617, 2018.
- [302] Mahsa Golchoubian, Moojan Ghafurian, Kerstin Dautenhahn, and Nasser Lashgarian Azad. Pedestrian trajectory prediction in pedestrian-vehicle mixed environments: A systematic review. *IEEE Transactions on Intelligent Transport*ation Systems, 2023.

- [303] Manuel Muñoz Sánchez, Chris van der Ploeg, Robin Smit, Jos Elfring, Emilia Silvas, and René van de Molengraft. Prediction horizon requirements for automated driving: Optimizing safety, comfort, and efficiency. arXiv preprint arXiv:2402.03893, 2024.
- [304] Haicheng Liao, Zhenning Li, Huanming Shen, Wenxuan Zeng, Dongping Liao, Guofa Li, and Chengzhong Xu. Bat: Behavior-aware human-like trajectory prediction for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10332–10340, 2024.
- [305] Somil Bansal, Andrea Bajcsy, Ellis Ratner, Anca D Dragan, and Claire J Tomlin. A hamilton-jacobi reachability-based framework for predicting and analyzing human motion for safe planning. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 7149–7155. IEEE, 2020.
- [306] Julian FP Kooij, Fabian Flohr, Ewoud AI Pool, and Dariu M Gavrila. Contextbased path prediction for targets with switching dynamics. *International Journal* of Computer Vision, 127(3):239–262, 2019.
- [307] Xiaobo Chen, Huanjia Zhang, Feng Zhao, Yu Hu, Chenkai Tan, and Jian Yang. Intention-aware vehicle trajectory prediction based on spatial-temporal dynamic attention network for internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):19471–19483, 2022.
- [308] Jerry S Wiggins. The five-factor model of personality: Theoretical perspectives. Guilford Press, 1996.
- [309] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14:261–292, 1996.
- [310] K Furuta, T Okutani, and H Sone. Computer control of a double inverted pendulum. Computers & Electrical Engineering, 5(1):67–84, 1978.
- [311] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015.

- [312] Erik Gärtner, Mykhaylo Andriluka, Erwin Coumans, and Cristian Sminchisescu. Differentiable dynamics for articulated 3d human motion reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13190–13200, 2022.
- [313] Gengshan Yang, Shuo Yang, John Z Zhang, Zachary Manchester, and Deva Ramanan. Ppr: Physically plausible reconstruction from monocular videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 3914–3924, 2023.
- [314] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. ACM Transactions on Graphics (TOG), 37(4):1–11, 2018.