Human motion trajectory prediction for robot navigation

Javad Amirian

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Informatique

Javad AMIRIAN

Human Motion Trajectory Prediction for Robot Navigation

Thèse présentée et soutenue à Rennes, le 8 Juillet 2021
Rainbow, Inria Rennes - Bretagne Atlantique
Thèse N° :

Rapporteurs avant soutenance :

Alexandre ALAHI	Assistant Professor : EPFL
Dinesh MANOCHA	Professor : University of Maryland

Composition du Jury :

Examinateurs :	Alexandre ALAHI	Assistant Professor : EPFL
	Dinesh MANOCHA	Professor : University of Maryland
	Nuria PELECHANO	Associate Professor : Polytechnic University of Catalonia
	Frédéric LERASLE	Professor : Paul Sabatier University (Toulouse III)
	Eric MARCHAND	Professor : University of Rennes 1
	Jean-Bernard HAYET	Professor : CIMAT
Directeur de thèse :	Julien PETTRÉ	Research Scientist : INRIA

In memory of my mother, ***Rangineh****, and my father,* ***Omid****, who did all their best to train me in the right path, and also my sister,* ***Sedigheh****, who was taken from me by Coronavirus. You are gone but your belief in me has made this journey possible.*

ACKNOWLEDGEMENT

This thesis would not have been completed without the support of many persons to whom I am extremely grateful. First and foremost, I thank my supervisor Dr. **Julien Pettré** for giving me the opportunity for a PhD. at Inria, Rainbow Team. Throughout my PhD he provided helpful ideas and encouraging support, and fueled self-motivation and ambition by creating a vastly positive and enthusiastic working atmosphere.

I am also indebted to my co-supervisor, Dr. **Jean-Bernard Hayet**, professor at CIMAT institute in Mexico, whose insightful expertise and meticulous guidance was crucial for this thesis. I will not forget the sleepless nights we were working together before the deadlines.

I would like to thank my committee members, the reviewers Dr. **Alexandre Alahi** and Dr. **Dinesh Manocha**, the examiners, Dr. **Nuria Pelechano** and Dr. **Frédéric Lerasle** and also the chairman, Dr. **Eric Marchand** for their precious time, shared positive insight and guidance. I am grateful to all of those with whom I have had the pleasure to work in Crowdbot project: Fabien, Arturo, Bonolo, David, Dan, Daniel, Diego, Ferran, George, Jen Jen, Sabarina, Solenne, Walter and Pat. It was a pleasure to work and exchange with you. Special thanks go to Ceilidh, the former project manager of the team. I always appreciate discussing with her, and I will never forget her and her endless help. I would like to thank all the people that I met or collaborated with in Rainbow group: Wouter, Alberto, Axel, Cedric, Fabrice, Florian, Julien B. and Ramana. I would also like to thank the Linkmedia team for all the discussions and fun times we had over the past three years: Hanwei, Cheikh, Suresh, Cyrielle, Oriane, Yann, Mateusz. I hope our friendship continues. Thanks to Suresh for taking the time to read and review my manuscript. I thank the students in Cimat institute, in Cimat: Francisco, Juan Jose, and Juan for their wonderful collaboration.

Most importantly, I have to admit that none of my achievements would ever have been possible without the unwavering support of my beloved family. I would like to thank my lovely wife Marzieh, from the bottom of my heart for being my dearest companion through the ups and downs of this long journey. You are too valuable to me to be thanked enough. I will not forget the enormous and heroic sacrifices you have made for me.

Many thanks go to my sisters: Sakineh and Mozhgan whose sacrificial care for me made it possible to get to here, and my brothers Ali and Reza, you have shown all the supports for me during my academic life to not be distracted by anything and to focus on my education. I can not find words to thank you enough. Many thanks also go to my family-in-law: my father- and mother-in-law Nadali and Fatemeh, for all their kind supports, and also my sibling-in-laws.

RÉSUMÉ EN FRANÇAIS

Motivations

Les robots, autrefois limités au travail en usine en tant que manipulateurs, étendent leur territoire aux espaces publics et aux maisons privées [Wal08]. S'appuyant sur l'intelligence artificielle, la théorie du contrôle, la mécatronique, les technologies des capteurs, etc., ils deviennent aptes à accomplir de plus en plus de tâches. Et même si leur rôle dans nos vies est encore marginal, certaines prévisions identifient la robotique comme l'une des "huit technologies essentielles" qui révolutionneront nos sociétés et nos entreprises [Lik20]. Les robots font leur apparition dans divers lieux intérieurs et extérieurs, où ils partagent le même espace que les êtres humains. Ils sont développés pour transporter des marchandises et des colis dans les zones urbaines, pour acheminer des médicaments et d'autres fournitures dans les hôpitaux, ou pour tenir compagnie aux personnes âgées ou handicapées.

Un coup d'œil aux chiffres révèle que le marché des robots de service et autres dispositifs de mobilité autonomes connaît une croissance rapide. Selon une étude de marché, le marché mondial des robots autonomes de livraison du dernier kilomètre devrait être multiplié par sept au cours des dix prochaines années [PS20]. D'ici là, les véhicules autonomes remplaceront les véhicules traditionnels et les robotaxis desserviront les zones urbaines peuplées. Selon certaines prévisions, un véhicule sur dix sera autonome d'ici 2030 [Sta19]. Les fauteuils roulants intelligents sont également de plus en plus répandus parmi les personnes handicapées, ce qui n'a pas de prix.

Et ce, alors que l'un des plus grands défis à venir pour ces industries est la sécurité humaine. Tous ces dispositifs de mobilité, qu'ils soient entièrement ou semi-autonomes, doivent être capables de percevoir, d'analyser et de prédire le comportement des personnes qui les entourent et d'effectuer des actions qui sont à la fois sûres et socialement acceptables.

Il existe également des scénarios spécifiques et cruciaux dans lesquels le robot doit opérer dans une foule à forte densité et à proximité immédiate des humains. Comme le risque de contact augmente dans ces scénarios, des niveaux plus élevés de précision de prédiction sont nécessaires. Par exemple, on peut imaginer les deux exemples suivants:

(a) un robot de livraison transporte un colis dans une zone commerciale mais se retrouve piégé dans une foule se déplaçant dans un flux unidirectionnel, car un bâtiment doit être évacué.

(b) un fauteuil roulant semi-autonome transportant une personne handicapée doit naviguer dans un flux bidirectionnel de piétons se déplaçant dans le couloir d'une gare.

L'inefficacité du robot à faire une bonne prédiction du mouvement des personnes peut conduire à l'un de ces deux problèmes:

(a) Le robot effectue des actions à haut risque, établit un contact indésirable avec les personnes ou les met mal à l'aise.

(b) Il prend l'action prudente la plus triviale, qui est 'aucune action'!

Le second problème est connu sous le nom de 'problème du robot gelé' et se produit lorsque le robot ne peut pas trouver une trajectoire qu'il pense pouvoir exécuter en toute sécurité [TK10]. La présence d'un module humain de prédiction de trajectoire semble alors critique pour le robot.

Énoncé du problème

Dans cette thèse, nous abordons le problème de la prédiction de trajectoire humaine (à court terme) dans des scénarios de foule: "Étant donné les trajectoires de mouvement d'un ou plusieurs piétons dans une scène, entourant un robot, celui-ci doit prédire, en temps réel, l'état spatio-temporel des piétons, et inférer leurs intentions (destinations) à court ou moyen terme."

Dans certains cas, d'autres informations auxiliaires, telles que la vitesse des piétons ou les propriétés du contexte de la scène, peuvent être utilisées pour améliorer la qualité de la prédiction. L'horizon de prédiction que nous considérons dans cette thèse est de l'ordre de quelques secondes (normalement jusqu'à cinq secondes). Un intervalle de temps de trois à cinq secondes est également utilisé pour capturer les trajectoires historiques des piétons.

La prédiction de la trajectoire est basée sur l'historique de l'emplacement bidimensionnel des pédales dans le système de coordonnées du monde ou du robot, qui est supposé être fourni par un système de "perception". Un défi notable dans notre problème est que la perception est faite à bord du robot, ce qui entraîne un certain nombre de limitations. Tout d'abord, la caméra du robot, qu'il s'agisse d'une caméra RVB ou d'un capteur de profondeur, est généralement installée à une hauteur inférieure à la taille humaine moyenne. Par conséquent, une partie des piétons environnants peut être partiellement ou totalement occultée. De plus, comme le robot est en mouvement, le calibrage géométrique estimé du robot peut être bruité. En raison de ce bruit, les estimations de la vitesse des piétons cibles peuvent être affectées de manière significative, et les trajectoires deviennent saccadées. En outre, en raison de la capacité limitée du processeur embarqué, la précision de la détection peut parfois être compromise. Par conséquent, le système de prédiction de trajectoire doit être capable de gérer des trajectoires d'entrée bruyantes, avec des changements d'identité, ainsi que des trajectoires courtes.

Il est important de noter la différence entre les définitions d'un 'chemin' et d'une 'trajectoire'. Un chemin est une séquence purement géométrique d'emplacements, tandis qu'une trajectoire est un chemin paramétré dans le temps avec une séquence d'emplacements horodatés. Le robot

utilise les trajectoires prédites pour estimer la probabilité d'une collision à un moment donné et pour planifier une trajectoire sûre et sans collision. Pour arriver à une conclusion sur la solution de ce problème, nous aimerions examiner la nature du mouvement humain et les défis du problème de prédiction.

Indices individuels

Le mouvement d'un piéton est influencé par de multiples *facteurs physiologiques*, tels que l'âge, le poids, la taille, les handicaps, mais aussi par des caractéristiques non physiques comme le type de personnalité et l'humeur de la personne. Ces variables peuvent influencer la vitesse des personnes, leur façon de se déplacer dans une foule et la distance sociale qu'elles gardent avec les autres.

Certaines études suggèrent que les *facteurs culturels* ont également un impact important sur la dynamique des piétons. L'espace personnel, la vitesse de marche, le côté d'évitement (le côté de dépasser d'autres piétons dans les situations d'évitement de collision) et les formations de groupe pourraient être différents selon les cultures et les régions [Kam11]. Chattaraj et al. [CSC09] ont réalisé des expériences en laboratoire dans des pays où les piétons sont plus nombreux. Inde et l'Allemagne et a trouvé des différences significatives dans les comportements de foule des personnes testées. L'étude suggère, par exemple, que "les Indiens sont moins sensibles à l'augmentation de la densité que les Allemands", et que "l'espace personnel minimum pour le groupe allemand était supérieur à celui du groupe indien."

Contexte de la scène (facteurs environnementaux)

Le chemin qu'emprunte un piéton dépend également du contexte de la scène. La présence de murs, de clôtures et d'autres obstacles peut rendre certaines zones inaccessibles, certaines surfaces peuvent ne pas être praticables et il peut même y avoir des préférences pour marcher sur d'autres types de surfaces, comme le trottoir ou l'herbe. En outre, les conditions météorologiques et l'éclairage peuvent avoir un impact sur les préférences des piétons. La différence de température entre une zone ombragée et une zone ensoleillée peut avoir une incidence sur les cheminements. En outre, le *type d'environnement* peut avoir un impact sur les décisions des piétons. Une (petite) étude sociologique menée en France [MBG+13] indique que la décision d'un piéton de traverser illégalement une rue dépend de sa perception de l'agrément et de la sécurité des espaces publics. Par exemple, les piétons ont tendance à se sentir plus en sécurité dans les centres-villes que dans les campagnes et les banlieues, ce qui peut avoir un impact considérable sur leur comportement.

Interactions sociales

Les mouvements des piétons qui partagent un espace commun dépendent d'une série d'interactions différentes. La *prévention des collisions* peut être considérée comme le type d'interaction le plus courant et le plus important entre des humains en mouvement. Mais il existe également d'autres comportements collectifs qui sont essentiels dans la modélisation des activités de la foule. Le comportement de regroupement est très fréquent dans les foules. Une étude (limitée) de Moussaid et al. [MPG+10] a observé que 70% des piétons dans une rue marchent avec d'autres personnes. Le *comportement de suivi* est plus fréquent dans les foules denses ou les passages étroits, où il n'y a pas de place pour dépasser d'autres piétons. Les *flux de foule* peuvent également apparaître dans des situations où le nombre de piétons est élevé.

Cette variété de types d'interactions sociales fait de la prédiction du mouvement un problème complexe, où chaque type d'interaction peut nécessiter un modèle différent et un ensemble différent de paramètres, ou peut représenter une dimension différente d'une grande variable latente.

Incertitude et multi-modalité

Les mouvements humains sont de nature multimodale. Cela signifie que, pour un même ensemble d'ob servations, il peut y avoir plusieurs chemins plausibles pour chaque piéton. La multi-modalité peut être liée à l'objectif du piéton. Mais aussi en raison de l'existence d'options multiples lors de l'interaction avec d'autres personnes. Par exemple, on peut passer du côté gauche ou droit d'un piéton, ou même, prendre l'une des multiples trajectoires plausibles lors du passage d'un groupe de personnes. Néanmoins, un système de prédiction de trajectoire bien conçu devrait faire face à cette multi-modalité et envisager de renvoyer un ensemble de trajectoires plausibles plutôt qu'une réponse unique.

Formulation mathématique

D'un point de vue mathématique, le problème de la prédiction de la trajectoire humaine peut être considéré érigé comme un problème de prévision de séquence. En supposant que la variable de prédiction (généralement, la positiond'un piéton) que l'on désigne par $\mathbf{x}$, la séquence passée d'observations $\mathbf{x}_{-\tau:0}$ est utilisée pour prédire la séquence future $\mathbf{x}_{1:T}$, où $[-\tau, 0]$ et $[1, T]$ sont les intervalles de temps de la séquence observée (passée) et de la séquence prédite (future).

Il est intéressant de faire une analogie avec d'autres problèmes de prévision de séquences. Un exemple pertinent peut être la prévision des marchés boursiers à l'aide des méthodes d'analyse technique [Mur99], où les changements et les modèles des prix récents du marché sont utilisés pour prévoir les prix futurs. Dans certaines prédictions de séquences, la fonction peut même

prendre des informations auxiliaires pour améliorer la précision de la prédiction. C'est le cas de la "prévision de la température de l'air", où, en plus des valeurs de température passées (c'est-à-dire $\mathbf{x}$), d'autres variables telles que la latitude du lieu, la vitesse du vent, l'humidité, etc. sont prises en compte dans le modèle de prévision.

Revenons à notre problème, ici la variable de prédiction est la position des piétons. En plus de la position passée du piéton d'intérêt (ou POI) $\mathbf{x}^i_{-\tau:0}$, trois ensembles d'informations peuvent être donnés à la fonction de prédiction:

1. l'emplacement des autres piétons dans la scène $\{\mathbf{X}^{\neg i}_{-\tau:0}\}$, qui peut être utilisé pour calculer les caractéristiques d'interaction sociale,
2. les attributs personnels du piéton du POI tels que la vitesse, l'orientation de la tête, la pose du corps ou même l'âge et le sexe estimés de la personne,
3. les propriétés du contexte de la scène telles que la géométrie de l'environnement et les obstacles ou les informations sémantiques telles que les passages piétons, les trottoirs, les feux de circulation, etc.

Nous considérons ces deux dernières informations, en général comme des variables auxiliaires et les désignons par A. Par conséquent, nous formulons le problème de prédiction de la trajectoire humaine avec la fonction abstraite suivante :

$$\begin{aligned} \hat{\mathbf{x}}^i_{1:T} &= f(\mathbf{x}^i_{-\tau:0}, \mathbf{X}^{\neg i}_{-\tau:0}, \mathbf{A}_{-\tau:0}|\theta) && (1)\\ &= f(\mathbf{X}_{-\tau:0}, \mathbf{A}_{-\tau:0}|\theta) && (2) \end{aligned}$$

qui renvoie $\hat{\mathbf{x}}^i_{1:T}$ comme les emplacements prédits du i ème agent pour les T prochains pas de temps, où θ est les paramètres de la fonction de prédiction. Également une méthode probabiliste, estime la distribution de $\hat{\mathbf{x}}^i_{1:T}$ avec la fonction de densité suivante:

$$p(\hat{\mathbf{x}}^i_{1:T}|\mathbf{X}_{-\tau:0}, \mathbf{A}_{-\tau:0}; \theta) \quad (3)$$

Dans le chapitre suivant, nous passons en revue différentes fonctions déterministes et probabilistes pour la prédiction de la trajectoire humaine.

Applications

La prédiction des mouvements de la trajectoire humaine a de multiples applications. Cela inclut les exemples que nous avons présentés dans les sections précédentes, mais aussi d'autres applications critiques qui nécessitent différents niveaux de prédiction des trajectoires humaines.

Mobilité autonome

Les systèmes mobiles autonomes, de quelque type que ce soit, qui travaillent dans un espace partagé avec des humains doivent prévoir les mouvements et les trajectoires de ces derniers.

Les **robots de service** qui travaillent dans les maisons, les restaurants, les centres commerciaux, les hôtels, les hôpitaux et les centres de soins doivent être conscients de la présence des personnes et assurer leur sécurité et leur confort. Pepper [Rob14], un robot semi-humanoïde développé par Softbank Robotics (anciennement Aldebaran Robotics) est capable d'accueillir les clients dans les magasins. REEM [Rob05], un robot de service humanoïde à roues, est placé dans les centres commerciaux et les expositions pour offrir un service et divertir les gens. Un autre robot de service mobile, LoweBot [Lab16], développé par Lowe's Innovation Labs et Fellow Robots, peut amener les clients à l'endroit où se trouvent les produits demandés dans une quincaillerie.

Évidemment, selon le type d'environnement, les caractéristiques physiques du robot (poids, taille, matériau, vitesse, etc.), ainsi que la vitesse, la densité et l'activité des personnes ou de la foule, la prédiction peut être plus ou moins critique. En raison de la hauteur des capteurs sur les robots de service et du champ de vision limité, il peut y avoir des occlusions importantes qui compliquent la détection et la prédiction des personnes. Les robots mobiles ne disposent généralement pas d'une grande puissance de traitement, ce qui complique encore plus l'exécution des algorithmes de détection, de prédiction et de navigation en temps réel.

Les **fauteuils roulants autonomes**, qui sont une version plus avancée des fauteuils roulants électriques, possèdent un certain niveau d'intelligence et d'autonomie pour aider les personnes handicapées. Le système de contrôle partagé de ces appareils est chargé d'exécuter les ordres de l'utilisateur du fauteuil roulant, tout en gérant les tâches de bas niveau telles que l'évitement des collisions et la fluidité du mouvement. La prédiction des personnes environnantes est alors nécessaire pour atteindre ces objectifs.

Les **véhicules autonomes** doivent également prédire le mouvement des usagers vulnérables de la route (VRU), c'est-à-dire les piétons, les cyclistes, les conducteurs de deux-roues motorisés et leurs passagers. Étant donné que les véhicules se déplacent à des vitesses plus élevées (par rapport aux robots de service) et peuvent causer des dommages plus graves aux usagers de la route, la prédiction de trajectoire joue un rôle encore plus important. Une question essentielle lorsqu'un véhicule voit dans la rue est la suivante : "Le piéton va-t-il traverser?" [KG14].

La *Society of Automotive Engineers* (SAE) [int16] a proposé une classification des véhicules avec six niveaux, du niveau zéro (aucune automatisation) à l'autonomie complète (cinquième niveau). La prédiction de trajectoire peut être utile à n'importe lequel de ces niveaux, bien qu'au fur et à mesure que l'autonomie augmente, le système aurait davantage de responsabilités pour

assurer la sécurité des passagers et des URV, en améliorant sa prédiction de l'environnement. Récemment, cette technologie a été utilisée dans les *motocyclettes* également [LC18].

Systèmes de surveillance

Dans les lieux présentant un intérêt pour la sécurité, tels que les aéroports, les gares et les centres commerciaux, le système de surveillance doit suivre les personnes ou reconnaître les activités pour l'analyse des ventes ou le contrôle des foules. Le système de suivi peut échouer à suivre les individus en raison de l'occlusion des piétons, des changements d'éclairage ou de l'apparence des piétons, des angles morts de plusieurs caméras, etc. Le système de suivi peut alors tirer parti de la prédiction des mouvements de la foule pour associer les personnes détectées aux pistes, en particulier lorsque la densité de la foule augmente. Cela peut aider à réduire les commutations d'identification et à augmenter la précision du suivi. Le système de prédiction peut également être utilisé pour détecter les comportements anormaux et déclencher des alarmes.

Simulation de foule

La simulation réaliste du mouvement d'une foule humaine est utile dans de nombreux domaines, tels que les logiciels de simulation, les jeux vidéo et les expériences de réalité virtuelle (RV). En général, l'objectif d'un simulateur de foule est de peupler une scène virtuelle avec une foule qui présente un comportement visuellement convaincant. Un système de prédiction de trajectoire humaine entraîné avec des données réelles est également capable de simuler le mouvement d'agents. Nous en discutons en détail dans le chapitre 6.

Notre application cible

Dans cette thèse, nous nous concentrons sur le problème de la navigation des robots dans des scénarios de moyenne et haute densité de population. Nous nous intéressons principalement aux robots de service (tels que le robot Pepper) et aux chaises roulantes intelligentes, qui partagent l'espace avec les humains. Les dispositifs de mobilité de ce type se déplacent à une vitesse proche de celle de la marche ou du jogging humain. Comme nous l'avons vu précédemment, dans notre application, en raison du mouvement du robot et de l'occlusion des piétons environnants, les entrées du système de prédiction peuvent être imparfaites et il doit être capable de traiter des trajectoires d'entrée bruitées, des changements d'identité et des pistes de courte durée.

Contributions

Les principales contributions scientifiques dans le cadre de la thèse sont couvertes dans cette section. Ces contributions qui sont publiées dans des conférences et des journaux évalués par

les pairs sont réalisées par le biais de collaborations avec d'autres chercheurs et co-auteurs qui sont mentionnés dans chaque chapitre. Cette thèse est axée sur le problème de la prédiction du mouvement des piétons pour la navigation des robots dans les scènes de foule.

• Évaluation de la complexité des jeux de données de prédiction de trajectoire humaine : avant de proposer notre modèle de prédiction, nous abordons la question de l'évaluation de la complexité d'un jeu de données de trajectoire humaine donné par rapport au problème de prédiction. Pour évaluer la complexité d'un jeu de données, nous définissons une série d'indicateurs autour de trois concepts: La prédictibilité de la trajectoire; La régularité de la trajectoire; La complexité du contexte. Nous comparons les jeux de données les plus courants utilisés dans le cadre de la HTP à la lumière de ces indicateurs et discutons de ce que cela peut impliquer sur l'évaluation comparative des algorithmes HTP.

• Un modèle de prédiction de trajectoire humaine multimodale basé sur un réseau adversarial génératif : nous proposons une nouvelle approche pour prédire la trajectoire des piétons en interaction avec d'autres personnes. Elle utilise un réseau adversarial génératif (GAN) pour échantillonner des prédictions plausibles pour tout agent dans la scène. Nous avons conçu un jeu de données d'exemples de trajectoires qui peut être utilisé pour évaluer les performances des différentes méthodes en préservant les modes de distribution des prédictions.

• Nous considérons la navigation de robots mobiles dans des environnements encombrés, pour lesquels la détection embarquée de la foule est typiquement limitée par des occlusions, et pour cela nous abordons le problème de l'inférence de l'occupation humaine dans l'espace autour du robot, dans les angles morts, au-delà de la portée de ses capacités de détection. Nous proposons une solution pour échantillonner les prédictions de présence humaine possible en se basant sur l'état d'un ensemble réduit de personnes détectées autour du robot ainsi que sur les observations précédentes de l'activité de la foule.

• Une nouvelle méthode de simulation de foule basée sur des données qui peut imiter le trafic observé de piétons dans un environnement donné Nous présentons une nouvelle méthode de simulation de foule basée sur des données qui peut imiter le trafic observé de piétons dans un environnement donné. Étant donné un ensemble de trajectoires observées, nous utilisons une forme récente de réseaux neuronaux, les réseaux adversariaux génératifs (GAN), pour apprendre les propriétés de cet ensemble et générer de nouvelles trajectoires avec des propriétés similaires. Nous définissons un moyen pour les piétons simulés (agents) de suivre une telle trajectoire tout en gérant l'évitement des collisions locales. Ainsi, le système peut générer une foule qui se comporte de manière similaire aux observations, tout en permettant des interactions en temps réel entre les agents. Par le biais d'expériences avec des données du monde réel, nous montrons que nos trajectoires simulées préservent les propriétés statistiques de leur entrée.

• Analyse du comportement de la foule en présence d'un robot : nous donnons des explications

sur une expérience de foule-robot, menée en collaboration avec l'University College London (UK) pour comprendre si et comment la dynamique de la foule des piétons sera modifiée en présence d'un robot. Cela peut donner un aperçu de la conception de systèmes de prédiction dans des scénarios de forte affluence.

Contenu du manuscrit

Cette thèse est structurée comme suit. Dans le chapitre 2 nous étudions l'état de l'art des solutions proposées pour ce problème. Ceci inclut la proposition d'une taxonomie de modèles pour mieux voir la relation entre les différentes approches. A travers les chapitres 3 - 7 nous présentons les contributions énumérées dans la section précédente : Dans le chapitre 3 nous abordons "l'évaluation de la complexité de la prédiction dans les ensembles de données sur les trajectoires humaines". Ce chapitre est désigné sous le nom d'*OpenTraj*. Dans le chapitre 4, "*Social-Ways*: Modèle de prédiction de trajectoire piétonne multimodale basé sur un GAN", est présenté. Le chapitre 5 est consacré à la nouvelle approche présentée pour "l'imputation de structures de foule occultes à partir de la détection de robots". Ensuite, le chapitre 6 présente la "simulation de foule basée sur les données du GAN". Et dans le chapitre 7 nous décrivons l'expérience foule-robot menée au laboratoire PAMELA de l'UCL ainsi que certaines leçons apprises qui peuvent être utilisées pour concevoir des systèmes de prédiction dans des scénarios de foule à haute densité. Enfin, dans le chapitre 8, nous discutons des résultats de cette thèse, des limites et des travaux futurs. Et, à la fin, nous présentons la conclusion pour résumer les contributions.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Robots that once were limited to work in factories as manipulators are expanding their territories to public spaces and private homes [Wal08]. Standing on the shoulders of artificial intelligence, control theory, mechatronics, sensor technologies, and etc., they are becoming qualified for more and more tasks. And even though their role in our lives may still be marginal, some forecasts identify Robotics as one of the 'Essential-Eight' technologies that will revolutionize our societies and businesses [Lik20]. Robots are appearing in various indoor and outdoor locations, where they share the same space with human beings. They are being developed to carry goods and parcels in urban areas, to ferry medications and other supplies in hospitals, or to give company to elderly or disabled people.

A look at the numbers reveals that the market for service robots and other autonomous mobility devices is growing rapidly. According to a market research report, the global market for autonomous last-mile delivery robots is expected to grow as much as seven-fold in the next ten years [PS20]. Until then, autonomous vehicles will replace traditional vehicles and Robotaxis will serve populated urban areas. There are predictions that one in every ten vehicles will be self-driving by 2030 [Sta19]. Smart wheelchairs are also becoming more widespread among disabled people, which is priceless.

This is while one of the biggest challenges ahead for these industries is human safety. All of these mobility devices, regardless of being fully- or semi-autonomous, should be able to perceive, analyze, and predict the behavior of its surrounding people and perform actions that are both safe and socially-acceptable.

There are also specific crucial scenarios where the robot should operate in a high-density crowd and closer proximity to humans. As the risk of contact increases in these scenarios, higher levels of prediction accuracy are required. For example, you can imagine the following two examples:

(a) a delivery robot is carrying a package in a commercial area but finds itself trapped in a crowd moving in a unidirectional flow, as a building needs to be evacuated,

(*b*) a semi-autonomous wheelchair transporting a person with a disability navigates a bidirectional flow of pedestrians moving in a train station's corridor.

In both scenarios, the inefficiency of the robot to make a good prediction of the motion of people may lead to one of these two problems:

(*a*) The robot performs high-risk actions, makes undesirable contact with people, or makes them feel uncomfortable,

(*b*) or it takes the most trivial cautious action, which is 'no action'!

The second problem is known as the "freezing robot problem" and occurs when the robot cannot find a trajectory that it believes is safe to execute [TK10]. Thus, to avoid the problem, it is important to forecast the future trajectories of surrounding pedestrians and to take safe actions.

1.2 Problem Statement

In this thesis, we address the problem of (short-term) human trajectory prediction (HTP) in crowded scenarios: "Given the motion trajectories of one or multiple pedestrians in a scene, surrounding a robot, it needs to predict, in real-time, the spatio-temporal state of the pedestrians, and infer their short or mid-term intentions (destinations)".

In some cases, other auxiliary information, such as velocity of pedestrians or scene context properties, can be used to enhance the prediction quality. The prediction horizon that we consider in this dissertation is in order of few seconds (normally up to five seconds). A three- to five-second time interval is also used to capture the historical trajectories of pedestrians.

The trajectory prediction is based on the history of the two-dimensional location of pedestrians within the world- or robot- coordinate system, which is assumed to be provided by a 'perception' system. A notable challenge in our problem is that perception is done on board of the robot, causing a number of limitations. First of all, the robot's camera, or image sensor, is typically installed at a lower height than average human height. As a result, part of surrounding pedestrians may be partially or fully occluded. Also, since the robot is moving, the estimated geometric calibration of the robot might be noisy. Due to this noise, estimations of the velocity of target pedestrians may be affected significantly, and the tracks become jerky. Further, owing to the limited capacity of the onboard processor, the sensing accuracy can be compromised sometimes. Therefore, the trajectory prediction system should be able to handle noisy input trajectories, ID-switchings, and short tracks.

It is important to note the difference between the definitions of a 'path' and a 'trajectory'. A path is a purely geometric sequence of locations, while a trajectory is a time-parameterized path with a sequence of time-stamped locations. The robot uses the predicted trajectories to estimate

the probability of a collision at a certain future time and to plan a safe and collision-free motion trajectory. To come to a conclusion about the solution to this problem, we would like to look at the nature of human motion and the challenges in the prediction problem.

1.2.1 Individual Cues

The motion of a pedestrian is impacted by multiple individual cues. They include *physiological factors*, such as age, weight, height, disabilities. These variables can affect the person's speed and how he or she moves in a group as well as the social distance that they keep with others. Non-physical factors such as personality type, mood, level of consciousness, etc., can also influence human motion. According to a study by Bera et al., different *personality traits* of pedestrians, such as being shy, aggressive, tense, etc., can be associated with different trajectory patterns in crowds [BRM17].

Some studies suggest that *cultural factors* also substantially impact pedestrian dynamics. The personal space, walking speed, avoidance side (the side of passing other pedestrians in situations of collision avoidance), and group formations might be different among different cultures and regions [Kam11]. Chattaraj et al. [CSC09] performed some experiments under laboratory conditions in India and Germany and found meaningful differences in the crowd behaviors of tested persons. The study suggests, for instance, that "participants India have been less sensitive to increase in density compared to the German participants" and that "the minimum personal space for the first group was less than that for the second group."

1.2.2 Scene Context (Environment Factors)

The path a pedestrian takes also depends on the scene context. The presence of walls, fence and other obstacles can make some areas inaccessible, some surfaces might not be walkable and there might even be preferences to walk on other types of surfaces, such as the pavement versus grass area. Additionally, weather conditions and lighting can impact the preferences of pedestrians. The temperature difference between a shaded area and a sunny area can affect the pathways.

Moreover, the *type of environment* can impact the decisions of pedestrians. A (small) sociological study conducted in France [MBG+13] indicates that the decision of a pedestrian to *illegally* crossing a street depends on their perceptions of the pleasantness and safety of public spaces. For example, the pedestrians tend to feel safer in city center environments rather than countryside and outskirts, which may substantially impact their behavior.

1.2.3 Social Interactions

The motions of pedestrians sharing a common space are dependent on a range of different interactions. *Collision avoidance* might be considered as the most common and also the most important type of interaction between moving humans. But there are also other collective behaviors that are essential in modeling crowd activities. *Grouping behavior* is very frequent in crowds. A (limited) study by Moussaid et al. [MPG+10] observed that 70% of pedestrians in a street walk with other persons. The *Leader-Follower behavior* is more common in dense crowds or narrow passages, where there is no room to overtake other pedestrians. *Crowd flows* can also emerge in situations with a large number of pedestrians.

This variety in social interactions makes motion prediction a complex problem. Each type of interaction may require a different model or set of parameters or may represent a different dimension of a large latent variable.

1.2.4 Multi-Modality

Human motions are multi-modal in nature. It means that given the same set of observations, there might be multiple plausible distinct paths for each pedestrian. This multi-modality can be attributed to interactions with other agents. For example, when two pedestrians walk towards each other, several modes of behavior develop, such as moving to the left or moving to the right. Likewise, pedestrians can have many choices of paths at intersections. Thus, a well-designed trajectory prediction system should cope with this multi-modality and consider returning a set of plausible paths rather than a single answer.

1.3 Mathematical Formulation

From a mathematical point of view, the human trajectory prediction problem can be considered as a sequence forecasting problem. Assuming the prediction variable (typically, the position of a pedestrian) to be denoted by $\mathbf{x}$, the past sequence of observations $\mathbf{x}_{-\tau:0}$ is used to predict the future sequence $\mathbf{x}_{1:T}$, where $[-\tau, 0]$ and $[1, T]$ are the time interval of the observed (past) sequence and predicted (future) sequence.

It is worth making an analogy to other sequence forecasting problems. A relevant example can be the prediction of stock markets using Technical Analysis methods [Mur99], where the changes and patterns in recent market prices are used to predict future prices. In some sequence predictions, the function can even take *Auxiliary Information* to enhance the prediction accuracy. This is the case of 'air temperature forecasting', where in addition to the past temperature values (i.e., $\mathbf{x}$), other variables such as the latitude of the place, wind speed, humidity, etc., are considered in the prediction model.

Returning to our problem, here, the prediction variable is the *position of pedestrians*. In addition to the past location of the pedestrian of interest (aka POI) $\mathbf{x}^i_{-\tau:0}$, three sets of information can be given to the prediction function:

1. location of other pedestrians in the scene $\{\mathbf{X}^{\neg i}_{-\tau:0}\}$, that can be used to compute the social interaction features,
2. personal attributes of the pedestrians such as velocity, head orientation, body pose or even estimated age and gender of the person,
3. scene context properties such as environment geometry and obstacles or semantic information such as crosswalks, sidewalks, traffic lights, and etc.

We consider the two latter information, in general, as auxiliary variables and denote them by $\mathbf{A}$. Hence we formulate the human trajectory prediction problem with the following abstract function:

$$\hat{\mathbf{x}}^i_{1:T} = f(\mathbf{x}^i_{-\tau:0}, \mathbf{X}^{\neg i}_{-\tau:0}, \mathbf{A}_{-\tau:0}|\theta) \tag{1.1}$$

$$= f(\mathbf{X}_{-\tau:0}, \mathbf{A}_{-\tau:0}|\theta) \tag{1.2}$$

that returns $\hat{\mathbf{x}}^i_{1:T}$ as the predicted locations of the ith agent for the next T time-steps, where θ is the parameters of the prediction function. Also a probabilistic method, estimates the distribution of $\hat{\mathbf{x}}^i_{1:T}$ with the following density function:

$$p(\hat{\mathbf{x}}^i_{1:T}|\mathbf{X}_{-\tau:0}, \mathbf{A}_{-\tau:0}; \theta) \tag{1.3}$$

In the next chapter, we review different deterministic and probabilistic functions for human trajectory prediction.

1.4 Applications

The prediction of human trajectory motions has multiple applications. This includes the examples that we have presented in the previous sections and also other critical applications that need different levels of prediction of human trajectories.

1.4.1 Autonomous Mobility

Mobile autonomous systems of any type that work in a shared space with humans need to predict people's motion and trajectories.

Service Robots that work in homes, restaurants, shopping malls, hotels, hospitals, and healthcare centers, should be aware of the presence of people and ensure their safety and comfort.

Pepper [Rob14], a semi-humanoid robot developed by Softbank Robotics (formerly Aldebaran Robotics), is able to welcome customers in shops. *REEM* [Rob05], a wheeled humanoid service robot is placed in shopping malls and exhibitions to give service and entertain people. Another mobile service robot, *LoweBot* [Lab16], developed by Lowe's Innovation Labs and Fellow Robots, can bring customers to the location of requested products in a hardware store. *Gita* [For20], a new service robot, is programmed with pedestrian etiquette and is able to pair with and follow a person (its owner) and carries up to 40 pounds of cargo.

There is no doubt that the prediction can be less or more critical depending on the type of environment, the robot's physical characteristics (weight, size, material, speed, etc.), and also the speed, density, and activity of the surrounding people. Due to the height of the sensors on service robots, and the limited field-of-view (FoV), there might be significant occlusion that complicates the detection and prediction of the people. The mobile robots usually do not come with high processing power, making it even more complicated to perform the detection, prediction and navigation algorithms in real-time. In this context, more accurate prediction of the near-future evolution of the environment helps the robot to reduce its re-planning effort [PSS+16].

Autonomous Wheelchairs are smart power wheel-chairs (PWC) with some level of intelligence and autonomy for helping disabled people. The semi-autonomous version comes with a shared-control system that is responsible for executing the orders of the wheelchair user while handling low-level tasks such as collision avoidance and the smoothness of motion. The prediction of surrounding people is then needed to ensure the safety of the user and the people around them.

Self-Driving Vehicles (SDV) also need to predict the motion of Vulnerable Road Users (VRU), that include pedestrians, cyclists, and riders of motorized two-wheeler and their passengers. Given that the vehicles move at higher speeds (compared to service robots) and can cause more severe harm to road users, trajectory prediction plays an even more important role. An essential question when a vehicle sees around the street is: "Will the pedestrian cross?" [KG14]

The Society of Automotive Engineers (SAE) [int16] has proposed a classification of autonomous vehicles with six levels, starting from zero-level: no automation to level five: full autonomy. The motion prediction system could be useful at different levels of autonomy, and as autonomy increases, the system would be required to improve its prediction of the surrounding environment to achieve the safety of passengers and VRUs. Additionally, recent *motorcycles* have incorporated motion prediction technology [LC18].

1.4.2 Surveillance systems

In places of security interests, such as airports, train stations, and shopping malls, the surveillance system needs to track people or make activity recognition for retail analytics or crowd

control. The tracking system might fail to track individuals due to occlusion of pedestrians, changes in lighting or pedestrian appearance, blind spots of multiple cameras, and etc. Then the tracking system can leverage the crowd motion prediction to associate detected persons to the tracks, especially as the crowd density increases. This can help to reduce the ID switches and increase tracking accuracy. The prediction system can also be used for the detection of anomalous behaviors and alarming.

1.4.3 Crowd Simulation

The simulation of human crowd motion is useful in multiple domains, such as simulation software, video games, and virtual reality (VR) experiences. Generally, the goal of a crowd simulator is to populate a virtual scene with a crowd that exhibits visually convincing behavior. It is also possible to simulate the realistic motion of agents with a human trajectory prediction system trained on real data. We discuss this in detail in Chapter 6.

1.4.4 Our Target Application

In this thesis, we focus on the Robot Navigation problem in medium- and high-density crowded scenarios. We are mainly interested in service robots (such as Pepper, the humanoid robot) and autonomous wheelchairs, that share space with humans. Mobility devices of this type move at close speed to human walking or jogging speed. As we discussed before, in our application, due to the motion of the robot, and the occlusion of surrounding pedestrians, the inputs of the prediction system can be imperfect and, it should be able to deal with noisy input trajectories, ID-switchings, and tracks with short duration.

1.5 Contributions

This section summarizes the key scientific contributions of the thesis. These contributions which are published at peer-reviewed conferences and journals are achieved through collaborations with other researchers and co-authors that are stated in each Chapter. This thesis is focused on the problem of pedestrian motion prediction for navigation of robots in crowded scenes.

• Assessing the complexity in Human Trajectory Prediction datasets: before proposing our prediction model, we address the question of evaluating how complex is a given human trajectory dataset with respect to the prediction problem. For assessing a dataset complexity, we define a series of indicators around three concepts: Trajectory predictability, Trajectory regularity, Context complexity. We compare the most common datasets used in HTP in the light of these indicators and discuss what this may imply on benchmarking of HTP algorithms.

• A multi-modal human trajectory prediction model based on Generative Adversarial Networks: we propose a novel approach for predicting the trajectory motion of pedestrians interacting with others. It uses a Generative Adversarial Network (GAN) to sample plausible predictions for any agent in the scene. We have designed a toy example dataset of trajectories that can be used to assess the performance of different methods in preserving the predictive distribution modes.

• An approach for imputing occluded crowd structures from robot sensing: we consider the navigation of mobile robots in crowded environments, for which onboard sensing of the crowd is typically limited by occlusions, and for that, we address the problem of inferring the human occupancy in the space around the robot, in blind spots, beyond the range of its sensing capabilities. We propose a solution to sampling predictions of possible human presence based on the state of a fewer set of sensed people around the robot as well as previous observations of the crowd activity.

• A novel data-driven crowd simulation method that can mimic the observed traffic of pedestrians in a given environment: we present a novel data-driven crowd simulation method that can mimic the observed traffic of pedestrians in a given environment. Given a set of observed trajectories, we use a recent form of neural networks, Generative Adversarial Networks (GANs), to learn the properties of this set and generate new trajectories with similar properties. We define a way for simulated pedestrians (agents) to follow such a trajectory while handling local collision avoidance. As such, the system can generate a crowd that behaves similarly to observations while still enabling real-time interactions between agents. Via experiments with real-world data, we show that our simulated trajectories preserve the statistical properties of their input.

• Analyzing crowd behavior in the presence of robots: we give explanations about a crowd-robot experiment conducted in collaboration with University College London (UK) to understand whether and how pedestrian crowd dynamics will be changed in the presence of a robot. This can give insight into designing prediction systems in high crowded scenarios.

1.6 Thesis Overview

This dissertation is structured as follows:

— In Chapter 2, we study the state-of-art solutions proposed for this problem. This includes proposing a taxonomy of models to see better the relation between different approaches.

— In Chapter 3, we address "assessing prediction complexity in Human Trajectory datasets". The Chapter is short-named as *OpenTraj*.

— In Chapter 4, '*Social-Ways*' our GAN-based multi-modal pedestrian trajectory prediction model is presented.

— Chapter 5 is dedicated to our novel approach for "imputation of occluded crowd structures from robot sensing".

— In Chapter 6, we present our "data-driven crowd simulation algorithm based on Generative Adversarial Networks.

— In Chapter 7, we describe the crowd-robot experiment conducted at the PAMELA lab of UCL along with some lessons learned that can be used in designing prediction systems in high-density crowd scenarios.

— Finally, in Chapter 8, we give a discussion upon the results of this thesis, the limitations and future work. And, in the end we present the conclusion to summarize the contributions.

Note that, there might be variations between notations used in different chapters.

Chapter 2

Related Work

2.1 Introduction

This chapter presents the state-of-the-art and the work related to the human motion prediction problem. We propose a taxonomy of methods for better understanding and classifying different approaches to the problem. In the sections that follow (Sec. 2.3 - 2.7), we try to cover a wide variety of solutions.

We begin with simple-dynamic models in Section 2.3, such as Constant-Velocity and Kalman Filtering. Then we review some of the research on crowd models that consider the interactions among agents for prediction of crowd motion in Section 2.4. We discuss Planning-based algorithms in Section 2.5. Later we review statistical pattern-based models and the family of approaches that include neural-network and other machine learning algorithms are reviewed in Section 2.6. In the Section 2.7 we focus on Reinforcement Learning methods and then go on to multi-modal prediction approaches in Section 2.8. We conclude the chapter through Section 2.9.

2.2 Taxonomy of Approaches

Here we present a taxonomy for human motion prediction models and the illustration of taxonomy is provided in Fig. 2.1. In the Fig. 2.1 we map various families of models, for better understanding their relationships. We begin by reviewing major factors considered for categorization for the illustration, and later we discuss other important aspects of prediction models. We mostly follow the surveys on: "Human Motion Trajectory Prediction" by Rudenko et al. [RPH+20],"Microscopic Crowd Simulation Algorithms", by van-Toll and Pettré [vTP21], and "In-depth Analysis of Deep Learning Models for Human Trajectory Forecasting" by Kothari et al. [KKA21] for this section.

2.2.1 Knowledge-driven vs. Data-driven

Primary factor in categorizing prediction models depends on the extent of reliance on on the data. In the Fig. 2.1, this can be seen on the vertical axis with color from blue to green representing the extent. At the bottom are **Knowledge-driven** category designed by human experts

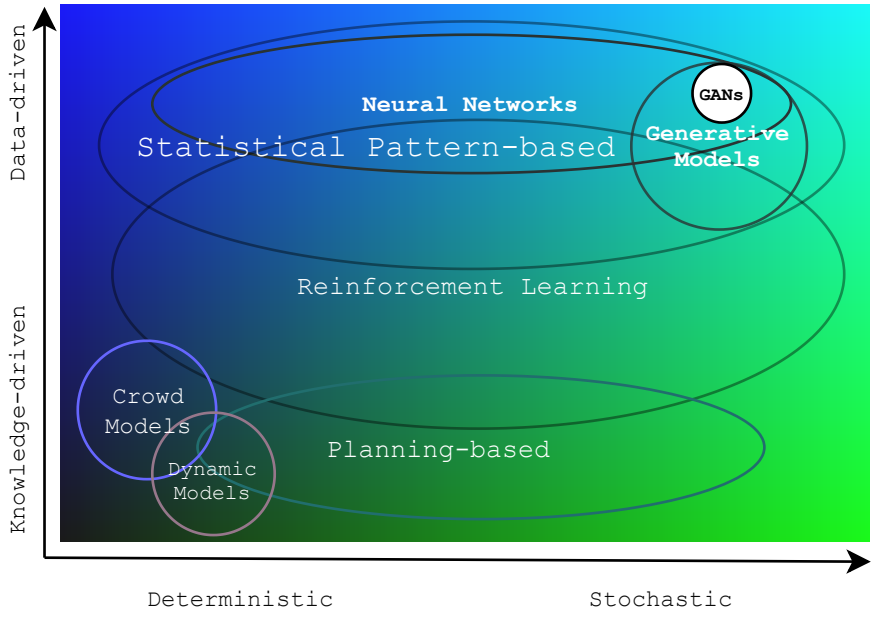


Figure 2.1 – Taxonomy of approaches for modeling and prediction of human motion

and their parameters optimized and adjusted manually. Dynamic models, planning-based models, and crowd models fall under this category. On the other hand, **Data-driven** category learn the patterns in data without much human-intervention. Neural-Networks based and statistical pattern-based models fall into this category. The prediction classes are more overlapping in nature, and hence no strict boundary line can be imposed among them. For example, while the neural-network models are designed by experts a more recent *AutoML* methods aim to make the process more automated and less dependent on human design [HZC21].

An important approach like Reinforcement Learning (RL) extend to both sides. In the traditional RL models, the reward function is designed manually, while in inverse RL and imitation learning models the rewards rely more on data. In a more recent Deep Reinforcement Learning neural networks are used for RL.

2.2.2 Deterministic vs. Stochastic

An important consideration in categorization in prediction models is the way uncertainty is handled, based on which the models may be deterministic or stochastic. The horizontal axis of Fig. 2.1 softly distinguishes between deterministic and stochastic models. The deterministic models, on the left of the figure, provides a single trajectory for a prediction, for a given observation input. Majority of crowd models (e.g., [vdBGLM11]) and dynamic models (e.g. constant-velocity model) are deterministic. Even the neural network based models which returns a single future

trajectory for the input data is a deterministic model [XPG18].

The Stochastic models, deal with the uncertainties while making predictions. In general, stochasticity originates from three different sources:

1. *Input*: here the model is input from a sampled value from a standard probability distribution (e.g., Gaussian). Even though the model is deterministic, the results will be stochastic in nature as the input samples are from the predictive distribution. A good example is approaches based on a deep generative model, such as Social-GAN [GJFF+18]).
2. *Internal*: here the model and the algorithm may be the source of randomness. For example some planning-based models [GSLS11] and some prediction algorithms based on Reinforcement Learning (RL) [CH10].
3. *Output*: here the model's output characterizes a probability distribution. For example, Alahi's Social-LSTM [AGR+16] returns the parameters mean, standard deviation and correlation coefficient of a a Bivariate Gaussian distribution from which one can sample and get an arbitrary number of predictions.

2.2.3 Uni-Modal vs. Multi-Modal

Prediction systems may be categorized based on whether they are uni-modal or multi-modal. Situations where the pedestrians follow several paths and/or interacts in multiple ways with other agents leads to multi-modality. For example, when two pedestrians are walking towards each other they may move to the left or move to the right. Likewise, they may have many choices of paths at intersections.

Ignoring this fact, may lead to problematic circumstances. For instance, a wrong uni-modal prediction can increase the risk of collision between pedestrians and/or vehicles. The deterministic and stochastic families can return uni-modal or multi-modal predictions. Examples of uni-modal deterministic approach includes the Social-Forces crowd model [HM95] and *Transformer* networks for trajectory forecasting [BFO20]. A neural network that returns multiple parallel prediction outputs [DT18], and the multi-hypothesis planning-based algorithm that computes the homotopy classes [GSLS11] are examples of multi-modal deterministic models.

Stochastic models with uni-modal solutions, usually return their outputs in form of a parametric distribution, for example Social-LSTM [AGR+16] and Kalman Filter model [Kal60]. Popular stochastic multi-modal approaches are generative models like Generative Adversarial Networks, we elaborate on this in Section 2.8. Here the models learn the multi-modal distribution of predicted trajectories, this is shown at the top right corner of the Fig. 2.1.

Reinforcement Learning, Pattern-based, and Planning-based prediction models include many unimodal and multimodal models. The Crowd Simulation algorithms are uni-modal and deter-

ministic, since they calculate and provide a single predicted motion for each agent. Likewise the Dynamic models can also be categorized based on modalities.

2.2.4 Additional Categorization Factors

We can categorize prediction models based on the source of information, in addition to the three factors mentioned above. Following the notation of prediction function in Eq. (1.2), the system can take different auxiliary information ($\mathbf{A}$) with the historical trajectories ($\mathbf{X}$) of the pedestrians: $f(\mathbf{X}_{-\tau:0}, \mathbf{A}_{-\tau:0}|\theta)$, where θ is the parameter of the prediction function.

• **Social Interactions**: Following the discussions in the Sec. 1.2 about the importance of social interactions in modeling human trajectory prediction (HTP) systems, it may be found that this is handled in number of ways in the literature. The work in [BHHA18, GHCG20] do not use social interactions at all, while [PESVG09, GJFF+18] considers interactions with neighboring agents, and some models considers interactions among the groups of agents as in [YBOB11, BZC18]. Normally, the interactions are defined as a function of the relative position, relative velocity, or relative orientation between two agents.

• **Scene Context**: Pedestrian behavior is highly influenced by the context surrounding them, like the obstacles, walk surface types (eg. sidewalks vs. grass). The scene context information is used in various ways in the literature. For example the model can be unaware of the static environment, such as in [SG13, GJFF+18], be aware of obstacles, like in [TK10, BKR+16, AGR+16], or handle semantic information such as in [SKS+18, DOLT20, SICP20].

• **Pedestrians' Intentions (Goals)**: Some prediction algorithms need an explicit notion of the goal or intention of pedestrians, such as [YAJR+21, DOLT20, TLT21]. The goal estimation is approached in different ways, depending on whether the scene context information is available. The simplest approach when no scene context information is available is to extrapolate the pedestrian's motion and use it as the long-term goal for the prediction.

• **Individual Cues**: Along with the historical observation of the location of the pedestrian, the individual cues can also be taken into account in a prediction model. Ma et al. [MHLK17] used the gender (male or female) and age (young or old) attributes of pedestrians in their trajectory prediction model. These attributes are populated by a visual classifier. Hasan et al. [HST+18] found that adding short sequences of head pose estimation improves the trajectory prediction.

• **Discrete vs. Continuous Space**: Some models work with real-valued locations of pedestrians [XPG18, SKS+18, ZQRX19], while others use discrete representations such as Grid-maps [WJF15, AGR+16, XHR18]. Sometimes Polar grids are also used to model the surrounding of agents [PPS+18]. Approaches using discrete maps may suffer from a reduced degree of accuracy

in their modeling and prediction results.

Using the proposed taxonomy, we can analyze various approaches in the light of the application needs. In the next six sections, we study these six families of prediction models in detail and Sec. In 2.9 we summarize them briefly to reach conclusions on suitability of the approach to our application.

2.3 Dynamic Models

Here we start with one of the simplest target object motion prediction models referred to as *Constant-Velocity* (CV) model. CV assumes the pedestrian (or in general, a moving object) will continue its motion with simple or no extra considerations like interaction between/among agents, contribution due to the scene or any other. Regarding the prediction function, defined in Eq. (1.2) we can rewrite the CV model as follows:

$$f(\mathbf{x}^i_{-\tau:0}, \mathbf{X}^{\neg i}_{-\tau:0}, \mathbf{A}) = \mathbf{x}^i_0 + t\ \mathbf{v}^i_0, \tag{2.1}$$

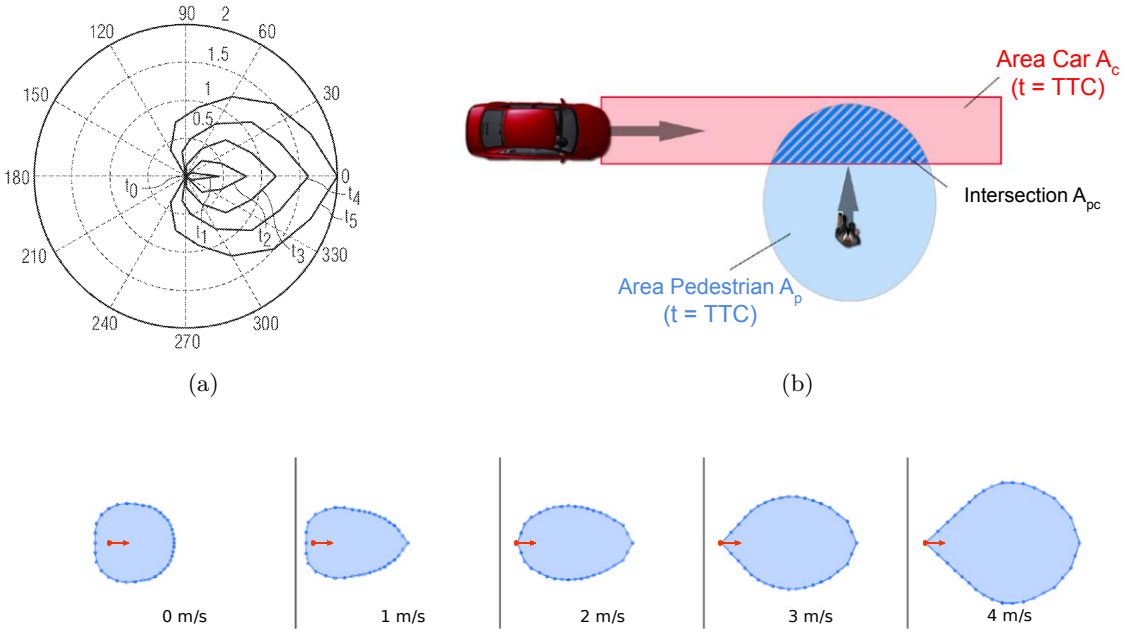
where $\mathbf{v}^i_0 = \frac{d\mathbf{x}^i_0}{dt}$.

Despite being too naive, it is hard to argue that the model is outdated. A recent work, titled "*What the constant velocity model can teach us about pedestrian motion prediction*" [SALK19] made a comparison with some state-of-the-art methods and showed that the CV model can perform on par and in some cases even outperform many approaches, on various datasets. The authors also presented following valuable findings about the HTP task:

- the long motion history of a pedestrian is not relevant for making predictions, and
- the interactions between pedestrians are less relevant than commonly believed.

Multi-modal version of the CV model was presented by the authors, by adding zero-mean Gaussian angular noises (with $\sigma = 25°$) to Eq. 2.1. *Constant-Acceleration* (CA) model is an extension of CV model. CA considers the acceleration (a higher order derivative) of the person. Furthermore, a higher level of granularity can be achieved by taking into account the object rotations around the z-axis. A model of this type is called *Constant Turn Rate and Velocity* (CTRV or simply CT), which is more suitable for prediction of non-holonomic agents.

Using CT model alongwith the physiological constraints of a human [SZ09] one can calculate the *Maximum Pedestrian Movement Area* as in [WGD+12] for a time interval of one second and for different velocities (see Fig. 2.2 (a) and (c)). Intersection of this area with the prediction area of a moving vehicle is used to estimate the collision risk at a given *time-to-collision* (TTC) moment (see Fig. 2.2 - (b)).



(a) (b)

(c)

Figure 2.2 – Maximum pedestrian movement area. (a): Probable movement/location area of a living being [SZ09]. (b): Intersection of predicted areas for pedestrian A_p and vehicle A_c. TTC stands for time to collision. (c): Maximum pedestrian movement areas for a time interval of $1s$. [WGD+12]

Kalman Filtering (KF) [Kal60] which is the backbone of many tracking and robotics systems deals with observation noise and inaccuracies, KF can be used together with the dynamic models for prediction of objects. The algorithm is an implementation of the Bayes filter for a Hidden Markov Model (HMM) with Gaussian distributions, and proposes two phases to *predict* and *update* the motion state of an object iteratively over time using incoming measurements. For predicting a trajectory one may use just the *predict* phase multiple times, while skipping the *update*. The covariance matrix of the predicted states captures the uncertainty in the predictions.

Interacting Multiple Models (IMM) assumes the motion of a target to be modeled with different dynamic models, in different circumstances. It assigns a probability of $p_{i,j}$ for switching from motion model (i) to (j). The combination of IMM and (CV/CA/CT) models alongwith *Extended Kalman Filter* (EKF) is studied in [SG13].

2.4 Crowd Models

Crowd study focuses on the design and use of algorithms to understand, predict, or simulate human crowd behavior. This family of algorithms is best suited for modeling interactions between people (or agents). These models simulate large groups of people, and they may also be used for prediction applications. Intuitively, the approach can be divided into two categories. One, agent-based (*microscopic*) algorithms that models each person in the crowd as an intelligent agent with its properties and goals; and two, flow-based (*macroscopic*) methods that model the crowd as a single entity [TCP06]. The two categories have a different range of applications in the entertainment industry like movies, animations, video games and in the architecture and safety industry in designing public places.

Microscopic models are suitable for predicting pedestrian motion in situations with considerable levels of pedestrian interactions. This could be where there is only one pedestrian in the scene, or when the distance between pedestrians is large, here they act similar to the CV model.

We give an overview of the two large sub-categories for these approaches. With respect to the categories defined in previous section, the majority of the crowd models, can be classified as uni-modal knowledge-based models.

2.4.1 Social Forces

The Social-Force Model (SFM), proposed by Helbing and Molnar in 1995 [HM95], is one of the most influential models in crowd simulation and robotics that inspired many later works. The general principle is to make the agents navigate according to potential fields caused by other agents through a repulsive force, which is also referred to as collision avoidance term, while trying to keep a desired speed and orientation toward the goal through an attractive force.

Later Pellegrini et al. [PESVG09] proposed Linear Trajectory Avoidance (LTA) as an extension to SFM. The first main contribution was predicting the "Expected Point of Closest Approach" between a pair of pedestrians and using that point as the driving force for decisions, instead of modeling the pedestrians as energy potentials at their current locations. Second important contribution of this work is to make the agents move in the optimal direction instead of just applying a gradient-dependent force.

SFM, also introduced an attraction force to model the interaction between agents in a group. Later, Yamaguchi et al. [YBOB11] proposed an extension in which they suggested people in the same group tend to stay close to each other and walk with similar speeds and in similar directions. To detect whether two pedestrians belong to the same group, a set of hand-crafted features were used with a Support Vector Machine (SVM) classifier [CV95].

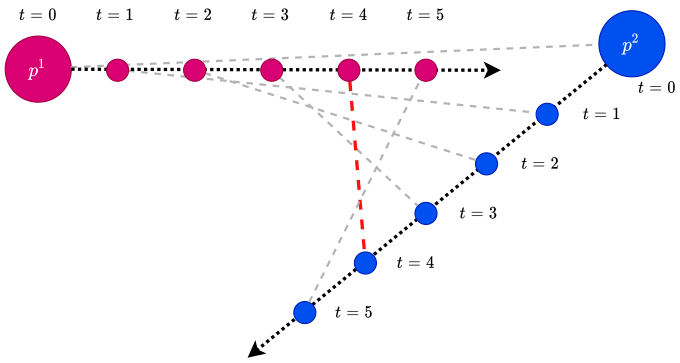


Figure 2.3 – Expected Point of Closest Approach between two moving agents p^1 and p^2. In this example the closest approach has occured at $t = 4$

2.4.2 Velocity Obstacles

Concept of velocity obstacles was an important step to make the simulated agents smart. It enhances the predictive ability of the agents by -linearly- extrapolating the location of neighboring agents. Such methods divide the velocity space of each agent into admissible and inadmissible subspaces and use a cost function to find optimal admissible velocity for the agent. The idea was initially proposed by Fiorini and Shiller [FS98]. A velocity-based collision-avoidance algorithm was introduced in 2007 by Paris et al [PPD07]. This model introduces the inadmissible velocities induced by each neighboring agent, and it uses a cost function to choose among the admissible velocities. Later in 2008, van den Berg et al. [vdBLM08] formulated "Reciprocal Velocity Obstacles" (RVO) to resolve the common oscillation problem in VO when applied to multi-agent navigation. Optimal Reciprocal Collision Avoidance (ORCA) [vdBGLM11], the main successor of RVO, transforms the mathematical definition of the collision-avoidance problem in order to compute its optimal velocity analytically by the agent.

PLEdestrian approach, proposed by Guy et al. [GCC+10] has similarities to the ORCA, but with an emphasis to emulate human behavior. In this approach, the optimal velocity for an agent is computed analytically using energy minimization via 'principle of least effort' (PLE). The energy function proposed measures the amount of 'effort' a pedestrian spends over time. The effort of the candidate velocity depends on the time to first collision (TTC) and on the estimated 'detour length' for steering the agent back to its goal.

Bayesian RVO (BRVO) [KGL+15] is the first online model using velocity obstacles to model the trajectory of moving pedestrians in a robot environment to learn their motion parameters and to predict trajectories via statistical inference techniques.

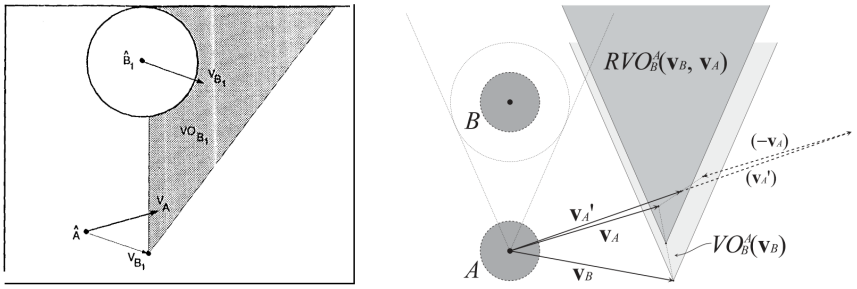


Figure 2.4 – Velocity obstacles. Left: the velocity obstacles caused by B_1 for A which is called VO_{B_1} forms a cone shape with its apex at V_{B_1}. Right: The Reciprocal velocity obstacle (RVO) has the same shape but its apex is located at a point between V_B and V_A.

2.4.3 Optimizing Model Parameters

With growing number of crowd modeling algorithms two key questions must be addressed:

1: How to optimize the model parameters?
2: How to choose a crowd model among the existing algorithms?

Wolinski et al. [WJGO+14] addressed these by introducing an optimization framework to find the best set of parameters for a given model and the reference dataset. The framework is designed with a good level of abstraction to handle different evaluation metrics for instance, microscopic metrics like average/final displacement error (ADE and FDE), path length, inter-pedestrian distance, progressive difference, and macroscopic metrics like vorticity and fundamental diagram metrics that treats the crowd as a continuum entity. The prediction problem is formulated through a parameterized function f, which takes in agents' current location $\mathbf{x}_0^i$, current velocity $\mathbf{v}_0^i$, goal $\mathbf{z}^i$ (for $1 \leq i \leq N$) and also the simulation parameters θ as it's inputs and returns the agents' next location $\hat{\mathbf{x}}_1^i$ (sometimes together with next velocity $\hat{\mathbf{v}}_1^i$) as it's outputs:

$$\hat{\mathbf{x}}_1^{1:N} = f(\mathbf{x}_0^{1:N}, \mathbf{v}_0^{1:N}, \mathbf{z}^{1:N}|\theta). \tag{2.2}$$

The type and the number of parameters assigned to each agents by the algoriths can be different. For example, the SFM assigns radius and comfort speed to each agent. Whereas, RVO assigns radius, comfort speed, neighbor distance, and a time horizon for the collision between agents to each agent.

Given the parameterized crowd model of Eq. (2.2), the goal is to find a parameter set θ^* which leads to the closest match between the model output and the reference data $\mathcal{H}$:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(\mathbf{x}_{0:1}^{1:N}, \mathbf{v}_0^{1:N}, \mathbf{z}^{1:N}) \sim \mathcal{H}}[d(\mathbf{x}_1^{1:N}, f(\mathbf{x}_0^{1:N}, \mathbf{v}_0^{1:N}, \mathbf{z}^{1:N}|\theta))], \tag{2.3}$$

where d is a function that measures the dissimilarity between ground truth and generated mo-

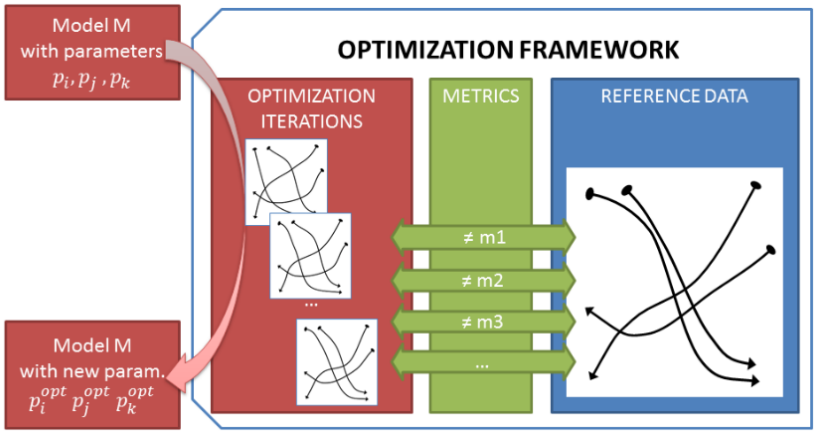


Figure 2.5 – An optimization framework for crowd algorithms [WJGO+14]. The approach optimizes parameters (here $\theta = \{p_i, p_j, p_k\}$) to match the results with the reference data. The framework has 3 components: an optimization module, a metrics module, and the reference data ($\mathcal{H}$).

tions. The samples in Eq. (2.3) are consequent frames of the reference data. Experimental results on multiple scenarios with different metrics show that RVO performs better by a substantial margin compared to SFM.

In some prediction approaches, such as Pellegrini's LTR model [PESVG09], the parameters are shared between the agents, which results in a simpler optimization problem with a much fewer number of parameters. At the same time, this might compromise the performance of the prediction.

2.5 Planning-based Models

The planning-based models present two-phase prediction algorithms that first, explicitly reason about the *intention* of pedestrians, and then find a trajectory that leads the agent toward this goal location. The trajectories are usually optimized according to user-defined criteria, such as path length, smoothness, etc.

The authors of [GSLS11] proposed a multi-hypothesis model for predicting pedestrian trajectories in crowded street scenes. In the first phase, they estimate the pedestrian goals by finding the intersections of street side lines, with infinity points along the street. Then they generate a set of plausible long-term motion plans that do not include collisions, redundancies, and unnecessary loops. This is done by constructing a neighborhood graph, in which each grid point on the ground not occupied by an obstacle is a vertex, and each pair of neighboring points are connected by an edge. Edge weights are the costs of moving from one vertex to another.

The graph vertices and edges are augmented with attributes called *winding numbers/angles*, concepts imported from topology that represent different configurations of passing through obstacles [HCGR11]. Finally they use Dijkstra's shortest paths algorithm [Dij59] to search the augmented graph. (See 2.6 (b, c))

In [VVS17] the scene is modeled through a set of attractive and repulsive potential fields, and Points of Interest (POI) are defined as some areas or spots whose attractiveness influences pedestrian behavior, such as monuments, places of public interest, public transportation, and etc. in the scene. Then a grid is defined for the observed area where each cell can take a different attribute such as road, crosswalk, POI, obstacle, etc. A potential field is defined over the grid map and finally the A^* search algorithm [HNR68] is used together with a parametrized heuristic function that encodes the path safety and distance. By adjusting this parameter, multiple predicted trajectories can be obtained.

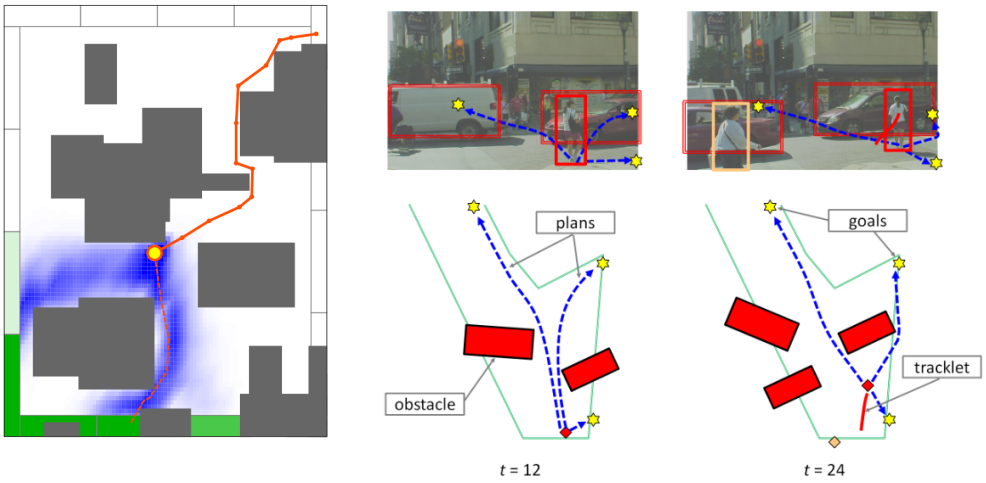
In [BF15] the agent's intention are first estimated and then the resulting probability distribution is used to predict the position of the agent in the future. The intention inference phase employs a Bayesian estimation framework and the trajectory prediction phase extrapolates the agent's position recursively, using a Probabilistic Road-Map (PRM) [KSLO96].

In [KAHS16] the goal of an agent is modeled as a 2D region in $\mathbb{R}^2$ and is inferred from a finite set of goals which is known a-priori. The posterior discrete distribution of goals is estimated using a Rao-Blackwellized Particle Filter (RBPF) [DDFMR13]. The trajectory prediction is then modeled using a Markov Decision Process (MDP) that abstracts a rational navigation into a policy function. This function specifies the optimal 'move direction' for reaching the goal as quickly as possible, while satisfying environment constraints and pedestrian's contextual preferences. Depending on a hyper-parameter α the policy assigns probabilities to (sub)-optimal plans. If $\alpha \to \infty$ it will assign nonzero probability only to the optimal plan (shortest path). On the other hand if $\alpha \to 0$, the policy becomes uniformly random. It is worth noting that the shortest path is not necessarily unique. The method uses different cost values for each surface type (sidewalk, crosswalk, road, grass), and also handles time-dependent information such as traffic signals.

In [RPA18] the interactions between agents in the scene are also taken into account using the social forces model. In this work, the previously learned stochastic policy is used to sample K joint paths.

2.6 Statistical Pattern-based Models

In this section we review another family of models that learns the prediction function from observed agent trajectories through approximation functions using machine learning algorithms.



(a) (b) (c)

Figure 2.6 – Planning-based trajectory prediction. (a): Using Probabilistic Road-Maps in [BF15] for predicting agent positions in the future. (b), (c): Multi-hypothesis motion plans, using homotopy classes in [GSLS11].

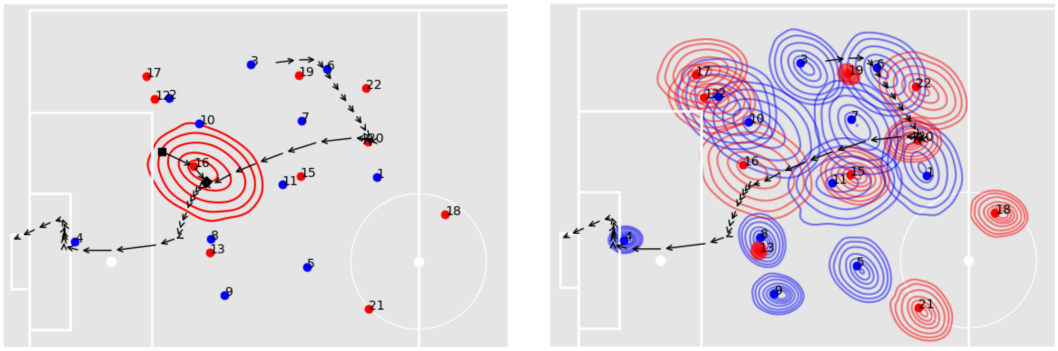
These models follow the *sense-learn-predict* scheme, in contrast to planning-based methods that follow a *sense-reason-predict* scheme [RPH+20]. The main motivation is that hand-tailored deterministic models may fail to adapt to a wide range of contexts compared to machine learning-based techniques that benefit from learning using large human motion datasets.

The approaches in this category mostly rely on neural networks. However, we begin by reviewing pattern-based approaches of machine learning and then dive into the ocean of neural network models. These approaches allow us to learn, identify patterns and make decisions with minimal intervention from humans.

2.6.1 (Conventional) Machine Learning Models

In this section we take a look at non-neural network models.

Support Vector Machines (SVM) [CV95] is maximum margin seperator method which is applied to human trajectory prediction in [XWF15]. SVM approaches classification problem by finding the hyperplane between two classes that maximizes the margin. The authors use a modified version of the *Edit distance* (originally presented for quantifying the dissimilarity between two strings) to trajectories. They use Edit distance, to estimate a minimum number of insertions, deletions, and value changes needed to turn one discretized trajectory into another. Later *K-medoid* algorithm (a clustering algorithm similar to k-means where instead of the mean as cluster center uses an element of the cluster as the representative of the cluster) is applied



(a) Movement prediction for one player (No. 16). The prediction countours are shown in red.

(b) Movement prediction for all the players of the two teams.

Figure 2.7 – Movement prediction for soccer players using Kernel Density Estimation [BLM19]

to cluster the, training, trajectories . After this SVM classifier is trained using these trajectory prototypes, newly observed trajectories are matched and classified to one of the categories and rest of the prototype used as the predicted motion.

Gaussian Processes (GPs) and its variants are also used for the prediction of human trajectories. A Gaussian Process is a collection of random variables with any subset of them having a joint Gaussian distribution. The authors of [ESR09] proposed to model changes in pedestrian positions (the displacements), given their current position, with GPs. The training trajectories are clustered based on the associated entry point into the scene and then a separate model is built for each cluster. The conditional distribution over displacements is then estimated given the current position and the cluster membership.

Interacting Gaussian Processes (IGPs) was introduced in [TK10] by Trautman and Krause. The model capturing non-Markovian nature of the agent trajectories is used for predicting the whereabouts of goal-driven social agents in crowds, here the parameters are learned from training data. The IGP distribution is obtained by coupling the individual GPs and by multiplying them in an interaction potential which is the product of Gaussian functions of the euclidean distance between all pair of agents. Vemula et al. [VMO17] developed IGP using real observations to learn a local interaction model that encodes How agents move based on How populated their vicinity is.

Kernel Density Estimation (KDE) is a probabilistic tool for predicting agents' positions. In [BLM19] Brefeld et al. obtained tables of reachable locations according to different speeds and time intervals for the soccer players and then proposed to incorporate these tables into a probabilistic movement model using KDE. An example of a prediction of soccer players after one second can be found in Fig. 2.7.

2.6.2 Neural Networks

"The terms *(Deep-) Neural-Networks* and *Deep Learning* are used interchangeably in the literature. They basically refer to a concept of Artificial Neural-Networks dating back to 1940s. The current term, 'deep learning', mostly refers to the larger number of layers compared to shallow networks used until 3 decades ago." (Goodfellow et al. [GBC16])

The "*Universal Approximation Theorem* shows that theoretically a neural network (with at least one hidden layer, and enough number of hidden units) can approximate any measurable function, to any desired degree of accuracy" [HSW89]. With the advent of deep neural networks and in particular, after successful developments of deep learning systems like many domains, the field of human trajectory prediction has witnessed phenomenal progress in recent years.

Multi-layer perceptrons (MLP) are a type of *Feed-Forward Networks* applied to a range of applications like diagnostics, control systems, pattern recognition [DBR09], time series prediction [KLSK96], handwritten character recognition [PS10], speech recognition [BDMFK91], and natural language processing [Ma02]. Earlier MLPs have been applied to the human trajectory prediction problem to learn a mapping between the past (observed) and future (predicted) trajectories [GDB+14]. A trajectory with n timesteps is used to learn and to predict m timesteps of the trajecory.

Recurrent Neural Networks (RNN) and their variants, Long Short Term Memories (LSTM) [GJ14] and Gated Recurrent Units (GRUs) [CGCB14], have shown promising results in sequence prediction tasks such as speech recognition [CBCB14] and machine translation [BCB14]. The sequential nature of motion has motivated the use of RNNs for the trajectory prediction task. The Social-LSTM architecture [AGR+16] associates each agent to an LSTM network and a social pooling aggregates the hidden states of the neighboring agents, to form an interaction feature. Then, each agent interaction feature is combined with its hidden state to predict positions for the future frames, with another LSTM network. In [PPS+18], LSTMs are used to capture the evolution of single trajectories, the interaction history is handled through an LSTM using histograms of closest distances over an angular discretization of the surrounding, the local obstacles are embedded in an occupancy grid. The system overview of this model is shown in Fig. 2.8.

Even though the recurrent networks provide promising results they have an important problem of processing input sequence in order, thus making them impossible to handle data in parallel. This becomes crucial bottleneck, especially when the input sequences become longer. This issue lead to use of *Convolutional Neural Networks* (CNNs) instead of RNNs for sequence-to-sequence mapping problems. *WaveNet* proposed for generating raw audio waveforms in a text-to-speech (TTS) system [ODZ+16] is one successful application of this. Taking the cue from here CNN models were later used in trajectory prediction as well [NM19]. Here a simple

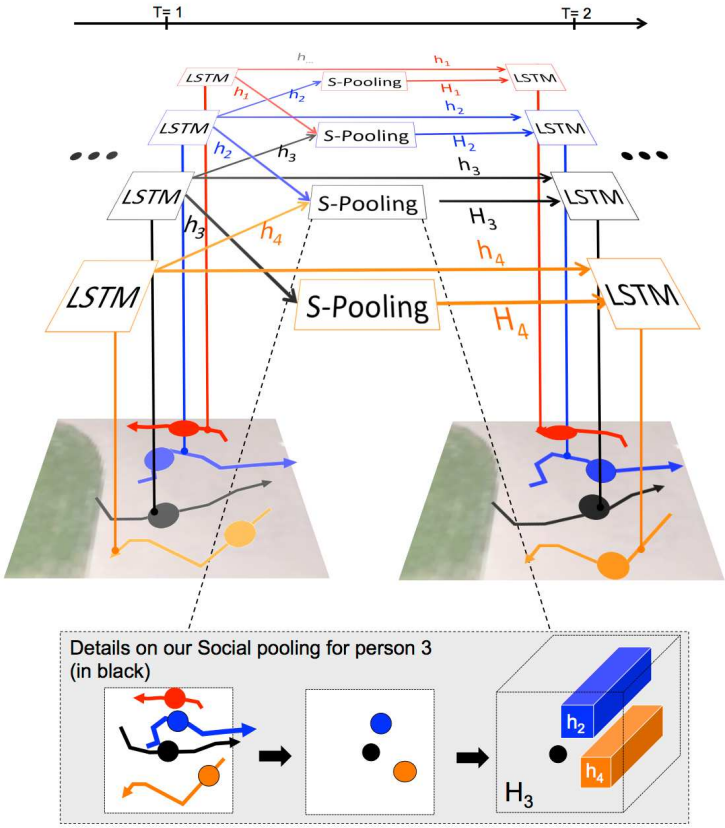


Figure 2.8 – System overview of Social-LSTM model [AGR+16]

CNN network first embeds the trajectory histories, applies the multi-layer convolutions, and decodes the hidden representation at the output to generate all the future time-steps at once. This model doesn't use any social or scene context. The *Convolutional LSTM* networks (ConvLSTM) which replace the linear operators in LSTMs with convolutional operators, have been used in [Li17] for human trajectory prediction.

Attention mechanism is another interesting concept used to measure the interdependence among elements in a system (agents in our case). Attention has been applied in trajectory prediction, and we use it in our proposed system explained in chapter 4. *Social-Attention* [VMO18] proposes a prediction model capturing the relative importance of each person while navigating in the crowd, irrespective of their proximity. The groups of agents are modeled as Spatio-temporal graphs where spatial and temporal edges are associated with RNNs. Temporal edges capture the evolution of single humans while spatial edges capture the evolution of agent-to-neighbor relationships. These features are combined linearly to produce an influence score used in the temporal network. Another method, Crowd Interaction Deep Neural Network model [XPG18], proposes to weight the motion features of pedestrians based on their spatial affinity.

Transformers have an encoder-decoder architecture proposed for sequence-to-sequence problems without relying on recurrent networks. Transformers leverage the attention mechanism. Encoder and decoder each consist of a set of layers that sequentially compute the *self-attention* among the layer inputs and pass them through a feed-forward layer. The first transformer model consisted of six encoders and six decoders and outperformed the Google Neural Machine Translation model on multiple tasks [VSP+17]. The authors of [GHCG20] showed that transformers can give competitive results in trajectory prediction task as well. The model did not take into account the social interactions among the agents.

We review "deep generative models" and "deep reinforcement learning" models in the following sections.

2.7 Reinforcement Learning

Reinforcement Learning (RL) [SB98] is a branch of machine learning in which agents learn from interacting with an environment by trying to maximize a notion of cumulative *reward*. The problem is usually formulated as a Markov Decision Process (MDP). The framework allows the system (e.g. a robot) to directly infer the navigation signals from perceptions, therefore it does not have an explicit notion of trajectory prediction. Chen et al. [CLEH17] proposed a Deep RL framework to learn navigating policies in crowded scenarios. They designed a fully connected neural network to estimate the *value function* in the RL. A value function, represents the expected *reward* of the agent given its *state* and the *policy* it followed. The authors of [CLKA19] proposed an improved framework that jointly models human-robot and human-human interactions and in [KGM+20] the authors model grouping of behaviors in pedestrian groups avoid collisions using an RL algorithm.

In contrast to the pure RL framework, other methods have been proposed to imitate and learn from human (or expert, in general) behavior. The study of *Inverse Reinforcement Learning (IRL)* (also called Inverse Optimal Control (IOC)) aims to learn the reward function from demonstrations by an expert [NR+00]. Chung et al. [CH10] have proposed a framework, to describe the relationships between pedestrian behaviors and environments. They integrate spatial effects in the pedestrian model in their IRL formulation to estimate the cost weighting of each spatial effect. Kitani et al. [KZBH12] have proposed an IRL approach to learn a feature-based cost function that captures agents' motion preferences for instance walking on the sidewalk or keeping distance from the parked cars and so on. In more recent works, the Safe-Critic framework [vdHNWG19] uses GANs with IRL to generate realistic and safe (or collision-free) trajectories. The *Imitation Learning* (IL) tries to directly extract a policy (a mapping from state to action) from data, unlike in RL which first learns a reward function and then apply the planning to generate the predictions. Building upon this, Generative Adversarial Imitation Learning (GAIL)

methods [HE16] take advantage of GANs to form an adversarial training loop between a policy **G**enerator and a **D**iscriminator. GAIL learns to discern 'expert' (or real) actions from that of 'agent' (or fake) actions. The Socially-Aware GAIL [ZSSZ18] approach, proposed by Zou et al. generates multi-modal socially acceptable trajectories via a learned reward function.

2.8 Multi-Modal Prediction

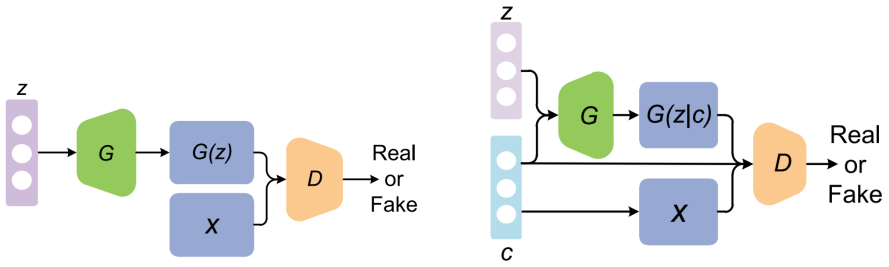
Predictive distribution of human trajectories is not single mode, that is the pedestrian might take one of the several potential trajectories. Hence, we are interested in models which return multi-modal predictive probability distribution of trajectories rather than a single prediction.

Multi-modal, here, does not pertain to multiple information modalities (or sensor types) as used earlier in multi-modal learning. Modality here refers to mode in a probability distribution. In many situations, the predictive distribution of a pedestrian motion is inherently multi-modal, e.g., at crossroads. Without proper modeling of multi-modality, a trajectory predictor, given observed trajectories with multiple possible outcomes may simply average all possible outputs.

Now we turn our attention on to generative machine learning models. A generative model models How the data is generated, and reflects the underlying causal relationships. In motion prediction this can be achieved by learning the joint distribution of $P(\mathbf{X}, \hat{\mathbf{X}})$, where $\mathbf{X}$ and $\hat{\mathbf{X}}$ being the observed and future trajectories, respectively. To further elucidate a generative model will be able to generate new photos of animals that look like real animals, while a discriminative model tells a dog from a cat. Mixture models like Gaussian Mixture Models (GMMs), Bayesian networks, Autoregressive models, Boltzmann machines, Energy-Based Models, and Normalizing flows are some of the generative machine learning models.

Generative Adversarial Networks [GPAM+14] (are refered as "the most interesting machine learning idea over the past decade" [AHTZ20]), together with *Variational Auto-Encoders* [KW13] it forms one of the two major families of *Deep Generative Models*, that learn data distributions and produce new samples using neural networks. Both models and their variations have been applied in: generating imaginary photographs of human faces [Gau14, KLA19, VK20], image-to-image translation [IZZE17], improving the resolution of images [LTH+17], photo blending [WZZH19], generation of new human poses [MJS+17], 3D object generation [WZX+16], video prediction [VPT16], music generation [RER+18], and so on.

In general, deep generative models, map a standard probability distribution (e.g. a multivariate Gaussian distribution) to the manifold of data. The random variables drawn from this distribution (usually denoted by $\mathbf{z}$) are fed to the model. The models that have other inputs than the random vector are called conditional. The block diagrams for a normal-GAN and conditional-GAN are depicted in Fig. 2.9.



(a) GAN (b) Conditional-GAN

Figure 2.9 – Difference between a Normal GAN (left) and a Conditional-GAN (right)

Conditional-GANs have been applied successfully to the task of human trajectory prediction. Social-GAN is a multi-modal trajectory predictor proposed by Gupta et al. [GJFF+18]. The conditional-GAN samples trajectories by handling the interactions between the observed pedestrians. This is done by pooling the GAN input random vector with a vector combining the hidden representations of the other pedestrians trajectories. The block diagram is depicted in Fig. 2.10.

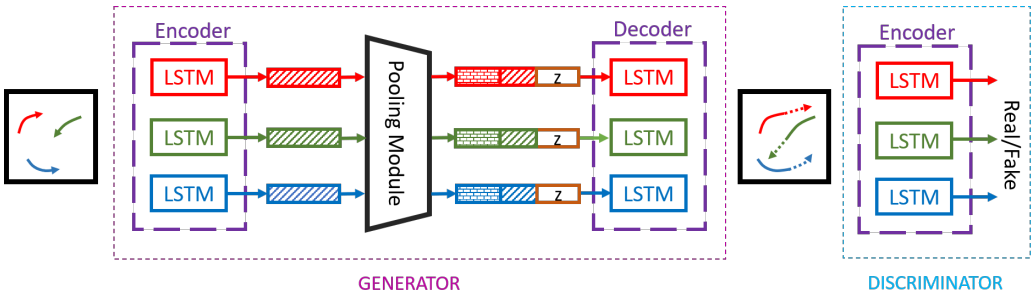


Figure 2.10 – Block diagram of Social-GAN model [GJFF+18]. The model consists of a Generator **G** that takes as input the past trajectories of agents, encodes the history of each person, and a pooling module is used to model the interaction between the agents. The Decoder in **G**, takes as input these encoded trajectories and the encoded social-interactions together with the random vector **z** and generates the predicted trajectories. The discriminator **D** decides whether this prediction looks real or not and returns a feedback to update the network weights.

Sadeghian et al. [SKS+18] proposed another model based on conditional-GANs, called *SoPhie*, that introduces physical attention to be combined with the social attention. This mechanism helps the model to learn where to look in a large scene and extract the most salient parts of the image relevant to the path. In a follow-up model, called *Social-BiGAT* [KSMM+19], the authors proposed to use two discriminators, one that operates at local pedestrian scale, and one that operates at a global scene-level scale. The latent encoding is done using a technique called BicycleGAN [ZZP+17], where the latent noise is mapped to an output trajectory, and then this

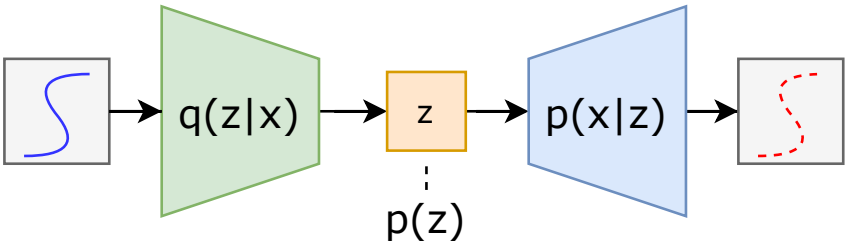


Figure 2.11 – System overview of a variational auto-encoder (VAE) that reconstructs a given trajectory, using encoder (green) and decoder (blue) networks.

trajectory is mapped back to the original latent space, to make sure it will mirror a normal distribution. Dendorfer et al. have proposed GoalGAN [DOLT20], a two-stage prediction model that first predicts the most likely destinations of the agent and then generates a set of plausible trajectories that route towards this goal.

Variational Auto-Encoders (VAEs) are another family of the deep generative models that have shown promising results in this problem. They essentially use the auto-encoding architecture (composed of an encoder network $\mathbf{q}(.)$ and a decoder network $\mathbf{p}(.)$, shown in Fig. 2.11) and are trained to minimize the reconstruction error between input data and the encoded-decoded signal. Unlike the standard auto-encoders, in VAEs, the hidden variable $\mathbf{z}$ represents a probability distribution that will be sampled to generate a variety of prediction outputs.

The *Desire* architecture [LCV+17] handles this multi-modality using variational auto-encoders. A Sample Generation Module based on CVAE generates samples of potential outcome trajectories and the Ranking and Refinement Module evaluates a learned long-term score associated to the sampled trajectories and refines these trajectories, in an inverse optimal control scheme. The *Trajectron* is another CVAE-based model designed by Ivanovic and Pavone [IP19]. The proposed model, jointly reasons and generates a distribution of future trajectories for each agent in the form of a GMM with 16 components. In a follow-up work, coined as *Trajectron++* [SICP20], the authors have incorporated the semantic maps and dynamics constraints into the system. The experimental results did not prove that the dynamical constraints can improve significantly the prediction accuracy of human trajectories. In a recent work, called *BiTraP* [YAJR+21], the authors proposed another CVAE-based model that first estimates the goal of the pedestrians and then use a bi-directional architecture to decode the predicted trajectories.

Beyond GANs and VAEs, other special types of deep generative models have also been applied to human trajectory prediction problem. The *Variational Recurrent Neural Networks* (VRNNs) [CKD+15] explicitly model the dependencies between latent variable across subsequent time-steps. The conditional version is used for multi-future trajectory prediction by Bertugli et al. [BCC+20]. In [SZDZ17], a social-aware LSTM, similar to [AGR+16], embeds the prior from

the training data as a hidden feature. Motion variability is taken into account by using layered Gaussian processes acting on the hidden features of the LSTMs. The authors of [LJM+20] have proposed *Multiverse*, a multi-future trajectory prediction model based on convolutional RNNs. In the proposed architecture, the location history of agents, together with the set of video frames, which are preprocessed by a semantic segmentation model, are fed to a neural network, to be encoded by a convolutional-RNN. The output of the encoder is fed to a convolutional RNN decoder for location prediction. The coarse location decoder outputs a heatmap over the 2D grid, and the fine location decoder outputs a vector offset within each grid cell. These are combined to generate a multimodal distribution over $\mathbb{R}^2$ for predicted locations.

2.9 Conclusion

The models reviewed under 'Dynamic Models' section (Section 2.3), while being very simple, can be implemented with few lines of code and can be run on a robot even with very limited resources. However they do not handle the 'social interactions' and 'scene context' information and also do not model the pedestrians' intention. The crowd simulation algorithms, reviewed in Section 2.4 propose interesting ideas that explicitly model the social interaction between the agents, while still, most of them fail to consider the scene-context information or the agent goal. Their simulation functions are simple enough to be implemented and deployed on a robot easily. As we explained the process of selecting the right parameters for the model is not straightforward. Due to the fact that the optimization function for selecting the parameters is non-convex and highly non-linear, it needs highly expensive algorithms such as genetic algorithms, which are not suitable for real-time robotic applications.

On the other hand, planning-based algorithms explicitly infer the pedestrians goal and run an optimization process to find the 'best' path that reaches the agent to this goal. Even though they might be able to give multiple predictions, still suffer from an important issue: the optimization criterion that is being solved is not necessarily the one that is followed by real human or crowd. The hand-crafted criteria, used in these models might be 'unrealistic'. The standard RL models also share the same problem, suffering from finding a correct reward function for the agents to be trained.

The pattern-based models focused on this aspect, and try to 'learn' the optimization criteria implicitly from data. The neural network systems have showed promising results in learning very complicated functions and the recurrent networks are suitable for prediction of a sequence of variables.

We discussed that even a very good 'single-model' algorithms may sometime fail to return an accurate path, due to the multi-modality nature of human motion. This can result in tak-

ing unsafe decisions by robots or autonomous vehicles. Hence we are interested in deploying multi-modal predictors. Among existing deep generative models we discussed in Section 2.8, the generative adversarial networks (GANs) provide a propitious framework to learn multi-modal distributions. Based on this modeling choice, we propose our multi-modal human trajectory prediction model, in Chapter 4, using GANs, while trying to solve one of their common issues: 'mode collapse'. Our proposed model, proposes a novel approach in handling the social interaction between the agents. We also propose a modified version of the algorithm for realistic simulation of crowds in Chapter 6.

There have been various new methods that proposed since publishing of our work. We comment about these new methods in the conclusion of the thesis (Chapter 8).

Chapter 3

OpenTraj: Assessing Prediction Complexity in Human Trajectories Datasets

3.1 Introduction

Efforts have been made towards a proper benchmarking of the existing techniques. This has led to the creation of pedestrians trajectories datasets for this purpose, or to the re-use of datasets initially designed for other purposes, such as benchmarking Multiple Object Tracking algorithms. Most HTP works [YBOB11, AGR+16, GJFF+18, AHP19] report performance on the sequences of two well-known HTP datasets: the ETH dataset [PESVG09] and the UCY dataset [LCL07]. The metrics for comparing prediction performance involve the Average Displacement Error (ADE) and the Final Displacement Error (FDE) on standardized prediction tasks. Other datasets have been used in the same way, but performance comparisons are sometimes subject to controversy, and it remains hard to highlight how significant good performance on a particular sequence or dataset means about a prediction algorithm. In this chapter, we address the following questions:

(1) How to measure the complexity or difficulty of a particular dataset for the trajectory prediction task?

(2) How do the currently used HTP datasets compare to each other?

(3) Can we draw conclusions about the strengths/weaknesses of state of the art algorithms?

Our contributions in this chapter are in two folds:

(1) We propose a series of meaningful and interpretable indicators to assess the complexity behind an HTP dataset, and

(2) we analyze some of the most common datasets through these indicators.

In Section 3.3, we categorize datasets complexity along three axes, trajectories predictability, trajectories regularity, and context complexity. In Section 3.4, we define indicators quantifying

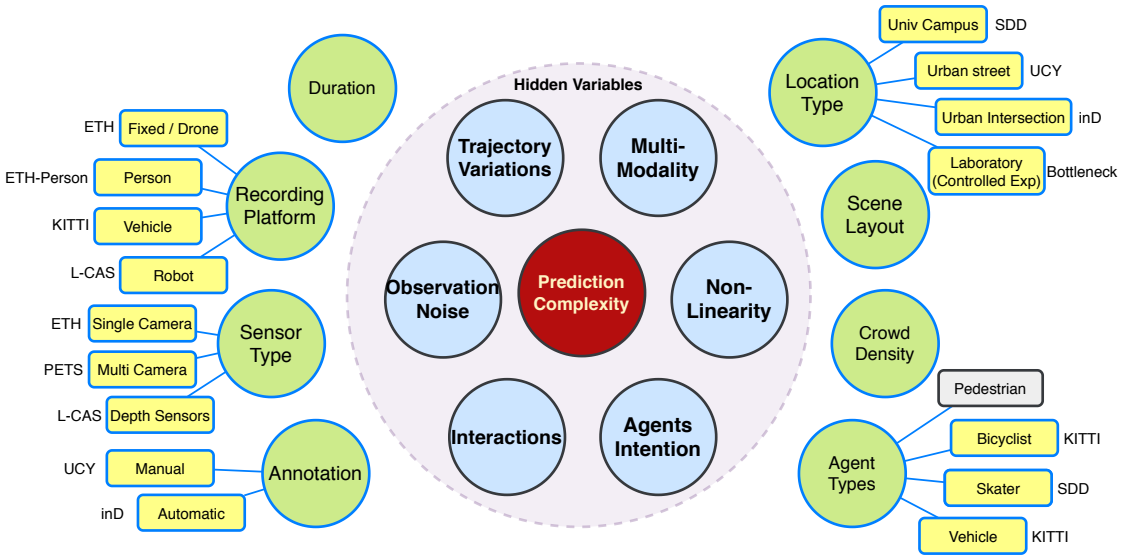


Figure 3.1 – Taxonomy of trajectories datasets for Human Trajectory Prediction.

the complexity factors. In Section 5.4, we apply these indicators on common HTP datasets and we discuss the results in Section 3.6.

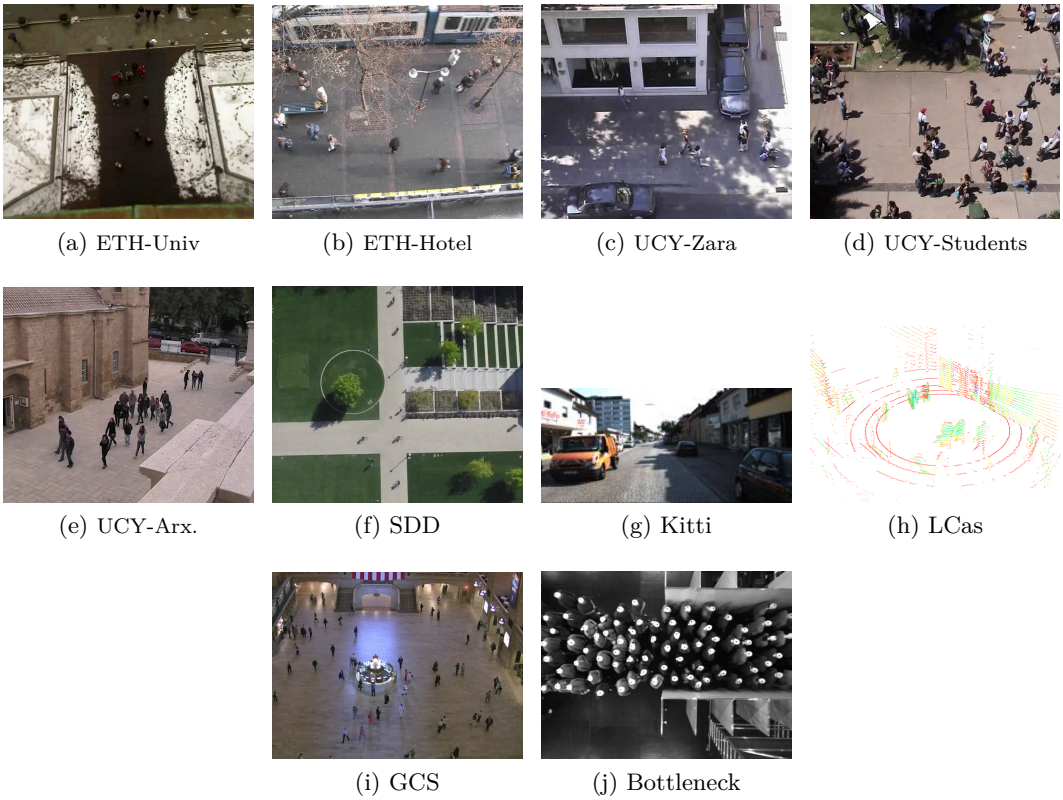
3.2 Related work: HTP datasets

Due to the non-rigid nature of the human body or occlusions, people tracking is a difficult problem and has attracted notable attention. Many video datasets have been designed as benchmarking tools for this purpose and used intensively in HTP. Following the recent progress in autonomous driving, other datasets have emerged, involving more complex scenarios. In this section, we propose a taxonomy of HTP datasets and review some of the most representative ones.

3.2.1 The zoo of HTP datasets: A brief taxonomy

Many intertwined factors explain how some trajectories or datasets are harder to predict than others for HTP algorithms. In Fig. 3.1, we summarize essential factors behind prediction complexity, as circles; we separate hidden (blue) and controlled (green) factors. Among hidden factors, we emphasize those related to the acquisition (noisy data), to the environment (multi-modality), or to crowd-related factors (interactions complexity). Some factors can be controlled, such as the recording platform or the choice of the location. To illustrate the variety of setups, snapshots from common HTP datasets are given in Fig. 3.2.

Raw data may be recorded by a single [PESVG09] or multiple [CBL+20] sensors, ranging from



(a) ETH-Univ (b) ETH-Hotel (c) UCY-Zara (d) UCY-Students

(e) UCY-Arx. (f) SDD (g) Kitti (h) LCas

(i) GCS (j) Bottleneck

Figure 3.2 – Sample snapshots from a few common HTP datasets.

monocular cameras [BKM+19, RSAS16, BR09] to stereo-cameras, RGB-D cameras, LiDAR, RADARs, or a mix [SKD+19, CBL+20]. Sensors may provide 3D annotations, but most HTP algorithms run on 2D data (the ground plane), and we focus here on 2D analysis.

Annotation is either manual [PESVG09, LCL07, GLSU13], semi-automatic [YLRO19], or fully automatic, using detection algorithms [BKM+19]. In most datasets, the annotations provide the agents' positions in the image. Annotated positions can be projected from image coordinates to world coordinates, given homographies or camera projection matrices. For moving sensors (robots [ELVG07] or cars [GLSU13, CBL+20, SKD+19]), the data are sensor-centered, but odometry data are provided to get all positions in a common frame.

3.2.2 A short review of common HTP datasets

HTP Datasets from static cameras and drones. The Performance Evaluation of Tracking and Surveillance (PETS) workshops have released several datasets for benchmarking Multiple Object Tracking [LTMR+15] systems. In particular, the 11 sequences of the PETS'2009 dataset [FS09], recorded through 8 monocular cameras, include data from *acting* pedestrians,

with different levels of density, and have been used in HTP benchmarking [SKG+18]. The Town-Centre dataset [BR09] was also released for visual tracking purposes, with annotations of video footage monitoring a busy town center. It involves around two thousand walking pedestrians with well structured (motion along a street), natural behaviors. The Wild Track dataset [CBB+18] was designed for testing person detection in harsh situations (dense crowds) and provides 312 pedestrian trajectories in 400-frame sequences (from 7 views) at 2fps. The EIF dataset [Maj09] gives ∼90k trajectories of persons in a university courtyard, from an overhead camera. The BIWI pedestrian dataset [PESVG09] is composed of 2 scenes with hundreds of trajectories of pedestrians engaged in walking activities. The ATC [BKIM13] dataset contains annotations for 92 days of pedestrian trajectories in a shopping mall, acquired from 49 3D sensors.

The UCY dataset [LCL07] provides three scenes with walking/standing activities. Developed for crowd simulation, it exhibits different crowd density levels and a clear flow structure. The Bottleneck dataset [SPS+09] also arose from crowd simulation and involved crowd controlled experiments (e.g., through bottlenecks).

VIRAT [OHP+11] has been designed for activity recognition. It contains annotated trajectories on 11 distinct scenes, in diverse contexts (parking lot, university campus) and mostly natural behaviors. It generally involves one or two agents and objects. A particular case of activity recognition is the one of sports activities [HLK16], for which many data are available through players tracking technology.

The Stanford Drone Dataset (SDD) [RSAS16] is a large scale dataset with 60 sequences in eight scenes, filmed from a still drone. It provides trajectories of ∼19k moving agents in a university campus, with interactions between pedestrians, cyclists, skateboarders, cars, buses. DUT and CITR [YLRO19] datasets have also been acquired from hovering drones for evaluating inter-personal and car-pedestrian interactions. They include, respectively, 1793 and 340 pedestrian trajectories. The inD dataset [BKM+19], acquired with a static drone, contains more than 11K trajectories of road users, mostly motorized agents. The scenarios are oriented to urban mobility, with scenes at roundabouts or road intersections. Ko-PER [SMS+14] pursues a similar motivation of monitoring spaces shared between cars and non-motorized users. It provides trajectories of pedestrians and vehicles at one road intersection, acquired through laser scans and videos. Similarly, the VRU dataset [BZH+18] features around 80 cyclists trajectories, recorded at an urban intersection using cameras and LiDARs. The Forking Paths Dataset [LJM+20] was created under the Carla 3D simulator, but it uses real trajectories, which are extrapolated by human annotators to simulate multi-modality with different latent goals.

AV datasets. Some datasets offer data collected for training/benchmarking algorithms for autonomous vehicles (AV). They may be more difficult because of the mobile data acquisition and because the trajectories are often shorter. LCAS [YDB17] was acquired from a LiDAR sensor on a mobile robot. KITTI [GLSU13] has been a popular benchmarking source in computer

vision and robotics. Its tracking sub-dataset provides 3D annotations (cars/pedestrians) for ∼20 LiDAR and video sequences in urban contexts. AV companies have recently released their datasets, as Waymo [SKD+19], with hours of high-resolution sensor data or Argo AI with its Argoverse [CLS+19] dataset, featuring 3D tracking annotations for 11k tracked objects over 113 small sequences. Nutonomy disclosed its nuScenes dataset [CBL+20] with 85 annotated scenes in the streets of Miami and Pittsburgh.

Benchmarking through meta-datasets. Meta-datasets have been designed for augmenting the variety of environments and testing the generalization capacities of HTP systems. TrajNet [SKG+18] includes ETH, UCY, SDD and PETS; in [BHHA18], Becker et al. proposed a comprehensive study over the TrajNet training set, giving tips for designing a good predictor and comparing traditional regression baselines vs. neural-network schemes. TrajNet++ [KKA21] proposes a hierarchy of categorization among trajectories to better understand trajectory distributions within datasets. By mid-2020, over 45 solutions have been submitted on Trajnet, with advanced prediction techniques [BHHA18, AGR+16, GJFF+18, ESR09, GHCG20], but also Social-Force-based models [HM95], and variants of linear predictors, that give accuracy levels of 94% of the best model [GHCG20]. In this work, we give tools to get a deeper understanding of the intrinsic complexities behind these datasets.

3.3 Problem description and formulation of needs in HTP

3.3.1 Notations and problem formulation

A trajectory dataset is referred to as $\mathbb{X}$. We assume that it is made of N_a trajectories of distinct agents. To be as fair as possible in our comparisons, we mainly reason in terms of absolute time-stamps, even though the acquisition frequency may vary. Within $\mathbb{X}$, the full trajectory of the i-th agent ($i \in [1, N_a]$) is denoted by $\mathbf{T}^i$, its starting time as τ^i, its duration as δ^i. For $t \in [\tau^i, \tau^i + \delta^i]$, we refer to the state of agent i at t as $\mathbf{x}_t^i$. We observe $\mathbf{x}_t^i$ only for a finite subset of timestamps (at camera acquisition times). The *frames* are defined as the set of observations at those times and are denoted by $\mathbf{F}_t$. Each frame contains K_t agents samples.

The state $\mathbf{x}_t^i$ includes the 2D position $\mathbf{p}_t^i$ in a Cartesian system in *meter*. It is often obtained from images and mapped to a world frame; the velocity $\mathbf{v}_t^i$, in m/s, can be estimated by finite differences or filtering.

To compare trajectories, following a common practice in HTP, we split all the original trajectories into N_t trajlets with a common duration $\Delta = 4.8s$. HTP uses trajlets of Δ_{obs} seconds as observations and the next Δ_{pred} seconds as the prediction targets. Hereafter, the set of distinct trajectories of duration Δ obtained this way are referred to as $\mathbf{X}^k$ where $k \in [1, N_t]$ covers the trajlets (with potentially repetitions of the same agent). Typically, $N_t \gg N_a$. Each trajlet may

be seen as an observed part of a longer trajectory and its corresponding target is referred to as $\mathbf{X}_+^k$.

In the following, we use functions operating at different levels, with different writing conventions. *Trajectory-level functions* $F(\mathbf{X})$, with capital letters, act on trajlets $\mathbf{X}$. Sometimes, we consider the values of F at specific time values t, at we denote the functions as $F_t(\mathbf{X})$. *Frame-level functions* $\mathcal{F}(\mathbf{F})$ act on frames $\mathbf{F}$.

3.3.2 Datasets complexity

We define three families of indicators over trajectory datasets that allow us to compare them and identify what makes them more "difficult" than other.

Predictability. A dataset can be analyzed through how easily individual trajectories can be predicted given the rest of the dataset, independently from the predictor. Low predictability on the trajlet distribution $p(\mathbf{X})$ makes forecasting systems struggle with multi-modal predictive distributions, e.g., at crossroads. In that case, stochastic forecasting methods may be better than deterministic ones, as the latter typically average over the outputs seen in the training data.

Trajectory (ir)regularity. Another dataset characterization is through geometrical and physical properties of the trajectories, to reflect irregularities or deviations to "simple" models. We will use speeds, accelerations for that purpose.

Context complexity. Some indicators evaluate the complexity of the context, i.e., external factors that influence the course of individual trajectories. To give an example, crowd density has a strong impact on the difficulty of HTP.

These indicators operate at different levels and may be correlated. For example, complex scenes or high crowdedness levels may lead to geometric irregularities in the trajectories and to lower predictability levels. Finally, even though it is common to *combine* datasets, our analysis is focused on individual datasets.

3.4 Numerical Assessment of a HTP Dataset complexity

Based on the elements from Section 3.3, we propose several indicators for assessing a dataset difficulty, most of the kind $F(\mathbf{X}^k)$, defined at the level of trajlets $\mathbf{X}^k$.

3.4.1 Overall description of the set of trajlets

To explore the distribution $p(\mathbf{T})$ in a dataset, we first consider the distributions of pedestrian positions at a timestep t. We parametrize each trajlet by fitting a cubic spline $\mathbf{p}_k(t)$ with $t \in [0, 4.8]$. For $t \in [0, 4.8]$, we get 50 time samples $\mathcal{S}(t) = \{\mathbf{p}_k(t),\ 1 \leq k \leq N_t\}$ and analyze $\mathcal{S}(t)$ through clustering and entropy:

— **Number of Clusters** $M_t(\mathbb{X})$: We fit a Gaussian Mixture Model (GMM) to our sample set using Expectation Maximization and select the number of clusters with the Bayesian Information Criterion [CH08].

— **Entropy** $H_t(\mathbb{X})$: We get a kernel density estimation of $\mathcal{S}(t)$ (see below in Section 3.4.2) and use the obtained probabilities to estimate the entropy.

High entropy means that many data points do not occur frequently, while low entropy means that most data points are "predictable". Similarly, a large number of clusters would require a more complex predictive model. Both indicators give us an understanding of how homogeneous through time are all the trajectories in the dataset.

3.4.2 Evaluating datasets trajlet-wise predictability

To quantify the trajectory predictability, we use the conditional entropy of the predicted part of the trajectory, given its observed part. Some authors [LWFZ16] have used alternatively the maximum of the corresponding density. For a trajectory $\mathbf{X}^k \cup \mathbf{X}^k_+$, we define the conditional entropy conditioned to the observed $\mathbf{X}^k$ as

$$H(\mathbf{X}^k) = -E_{\mathbf{X}_+}[\log p(\mathbf{X}_+|\mathbf{X}^k)]. \tag{3.1}$$

We use kernel density estimation with the whole dataset $\mathbb{X}$ (N_t trajectories) to estimate it. We have N_{obs} observed points during the first Δ_{obs} seconds (trajlet $\mathbf{X}_k$) and N_{pred} points to predict during the last Δ_{pred} seconds (trajlet $\mathbf{X}^+_k$). We define a Gaussian kernel K_h over the sum of Euclidean distances between the consecutive points along two trajectories $\mathbf{X}$ and $\mathbf{X}'$ with N points each (in $\mathbb{R}^{2N}$):

$$K_{h,N}(\mathbf{X}, \mathbf{X}') = \frac{1}{(2\pi h^2)^N} \exp\left(-\frac{1}{2h^2}\|\mathbf{X} - \mathbf{X}'\|^2\right), \tag{3.2}$$

where h is a common bandwidth factor for all the dimensions. We get an approximate conditional density as the ratio of the two kernel density estimates

$$p(\mathbf{X}_+|\mathbf{X}^k) \approx \frac{\frac{1}{N_t}\sum_{l=1}^{N_t} K_{h,N_{obs}+N_{pred}}(\mathbf{X}^k \cup \mathbf{X}_+, \mathbf{X}^l \cup \mathbf{X}^l_+)}{\frac{1}{N_t}\sum_{l=1}^{N_t} K_{h,N_{obs}}(\mathbf{X}^k, \mathbf{X}^l)}. \tag{3.3}$$

Since $K_{h,N_{obs}+N_{pred}}(\mathbf{X}^k \cup \mathbf{X}_+, \mathbf{X}^l \cup \mathbf{X}^l_+) = K_{h,N_{obs}}(\mathbf{X}^k, \mathbf{X}^l) K_{h,N_{pred}}(\mathbf{X}_+, \mathbf{X}^l_+)$, we can express

the distribution of Eq. 3.3 as the following mixture of Gaussian:

$$\begin{aligned} p(\mathbf{X}_+|\mathbf{X}^k) &\approx \sum_{l=1}^{N_t} \omega_l(\mathbf{X}^k) K_{h,N_{pred}}(\mathbf{X}_+, \mathbf{X}_+^l) \\ \text{with } \omega_l(\mathbf{X}^k) &= \frac{K_{h,N_{obs}}(\mathbf{X}^k, \mathbf{X}^l)}{\sum_{l=1}^{N_t} K_{h,N_{obs}}(\mathbf{X}^k, \mathbf{X}^l)}. \end{aligned} \tag{3.4}$$

For a trajlet $\mathbf{X}^k$, we estimate $H(\mathbf{X}^k)$ by sampling M samples $\mathbf{X}_+^{(i)}$ from Eq. 3.4:

$$H(\mathbf{X}^k) \approx -\frac{1}{M} \sum_{m=1}^{M} \log\left(\sum_{l=1}^{N_t} \omega_l(\mathbf{X}^k) K(\mathbf{X}_+^{(m)}, \mathbf{X}_+^l) \right). \tag{3.5}$$

3.4.3 Evaluating trajectories regularity

In this section, we define geometric and statistical indicators evaluating how *regular* individual trajectories $\mathbf{X}^k$ in a dataset may be.

(a) Motion properties

A first series of indicators are obtained through *speed distributions*, where speed is defined as: $s(\mathbf{x}_t) = \|\mathbf{v}_t\|$. At the level of a trajectory $\mathbf{X}^k$, we evaluate the mean and the largest deviation of speeds along the trajectory

$$S^{avg}(\mathbf{X}^k) = \underset{t\in[\tau^k, \tau^k+\delta^k]}{\text{average}} (s(\mathbf{x}_t)) \tag{3.6}$$

$$S^{rg}(\mathbf{X}^k) = \max_{t\in[\tau^k, \tau^k+\delta^k]} (s(\mathbf{x}_t)) - \min_{t\in[\tau^k, \tau^k+\delta^k]} (s(\mathbf{x}_t)). \tag{3.7}$$

The higher the speed, the larger the displacements and the more uncertain the target whereabouts. Also, speed variations can reflect on high-level properties such as people activity in the environment or the complexity of this environment.

Regularity is evaluated through accelerations $a(\mathbf{x}_t) \approx \frac{1}{dt}[s(\mathbf{x}_{t+dt}) - s(\mathbf{x}_t)]$. It can reflect the interactions of an agent with its environment according to the social-force model [HM95]: agents typically keep their preferred speed while there is no reason to change it. High accelerations appear when an agent avoids collision or joins a group. We consider the average and maximal accelerations along $\mathbf{X}^k$

$$A^{avg}(\mathbf{X}^k) = \underset{t\in[\tau^k, \tau^k+\delta^k]}{\text{average}} (|a(\mathbf{x}_t)|); \tag{3.8}$$

$$A^{max}(\mathbf{X}^k) = \max_{t\in[\tau^k,\tau^k+\delta^k]}(|a(\mathbf{x}_t)|). \tag{3.9}$$

(b) Non-linearity of trajectories

Path efficiency is defined as the ratio of the distance between the trajectory endpoints over the trajectory length:

$$F(\mathbf{X}^k) = \frac{\left\|p_{\tau^k+\delta^k} - p_{\tau^k}\right\|}{\int_{t=\tau^k}^{\tau^k+\delta^k} dl}. \tag{3.10}$$

The higher its value, the closer the path is to a straight line, so we would expect that the prediction task will be "easier" for high values of $F(\mathbf{X}^k)$.

Another indicator is the average angular deviation from a linear motion. To estimate it, we align all trajlets by translating them to the origin of the coordinate system and rotating them such that the first velocity is aligned with the x axis:

$$\hat{\mathbf{X}}^k = \begin{bmatrix}\mathbf{R}(-\measuredangle\mathbf{v}_0^k) & -\mathbf{p}_0^k\end{bmatrix}\begin{bmatrix}\mathbf{X}^k \\ 1\end{bmatrix}^T. \tag{3.11}$$

Then the deviation of a trajectory $\mathbf{X}^k$ at t and its average value are defined as:

$$D_t(\mathbf{X}^k) = \measuredangle\hat{\mathbf{X}}_t^k \text{ and } D(\mathbf{X}^k) = \underset{t\in[\tau^k,\tau^k+\delta^k]}{\text{average}}(D_t(\mathbf{X}^k)). \tag{3.12}$$

3.4.4 Evaluating the context complexity

The data acquisition context may impact HTP in different ways. It may ease the prediction by introducing correlations: With groups, it can be easier to predict one's motion from the other group members. In general, social interactions result into adjustments that may be generate non-linearities (and lower predictability).

(a) Collision avoidance

Collision avoidance is one of the most basic types of interaction between two pedestrians. Higher density can result into more interactions, this aspect is also evaluated by the density metrics below. However, high-density crowds may even ease the prediction (e.g., laminar flow of people). To reflect the intensity of collision avoidance-based interactions, we use the *distance of closest approach* (DCA) [OMC+13] at t, for a pair of agents (i, j):

$$\text{dca}(t,i,j) = \sqrt{\|\mathbf{x}_t^i - \mathbf{x}_t^j\|^2 - (\max(0, \frac{(\mathbf{v}_t^i - \mathbf{v}_t^j)^T(\mathbf{x}_t^i - \mathbf{x}_t^j)}{\|\mathbf{v}_t^i - \mathbf{v}_t^j\|}))^2}, \tag{3.13}$$

and for a trajlet $\mathbf{X}^k$ (relative to an agent i_k), we consider the overall minimum

$$C(\mathbf{X}^k) = \min_{t\in[\tau^k,\tau^k+\delta^k]} \min_j \mathrm{dca}(t, i_k, j). \tag{3.14}$$

In [KSG14], the authors suggest that time-to-collision (TTC) is strongly correlated with trajectory adjustments. The TTC for a pair of agents i, j, modeled as disks of radius R, for which a collision will occur when keeping their velocity, is

$$\tau(t, i, j) = \frac{1}{\|\mathbf{v}_t^i - \mathbf{v}_t^j\|^2}[\delta_t^{ij} - \sqrt{(\delta_t^{ij})^2 - \|\mathbf{v}_t^i - \mathbf{v}_t^j\|^2(\|\mathbf{x}_t^i - \mathbf{x}_t^j\|^2 - 4R^2)}] \tag{3.15}$$

where $\delta_t^{ij} = (\mathbf{v}_t^i - \mathbf{v}_t^j)^T(\mathbf{x}_t^i - \mathbf{x}_t^j)$. In [KSG14], the authors also proposed quantifying the interaction strength between pedestrians as an energy function of τ:

$$E(\tau) = \frac{k}{\tau^2} e^{-\frac{\tau}{\tau^+}}, \tag{3.16}$$

with k a scaling factor and τ^+ an upper bound for TTC. Like [KSG14], we estimate the actual TTC probability density between pedestrians (from Eq. 3.15) over the probability density that would arise without interaction (using the time-scrambling approach of [KSG14]). Then we estimate $E(\tau)$ with Eq. 3.16. As the range of well-defined values for τ may be small, we group the data into $0.2s$ intervals and use t-tests to find out the lower bound τ^- when two consecutive bins are significantly different ($p < 0.05$). The upper bound τ^+ is fixed as $3s$. TTC and energy interaction are extended for trajlets (only if there exists future collision):

$$T(\mathbf{X}^k) = \min_{t\in[\tau^k,\tau^k+\delta^k]} \min_j \tau(t, i_k, j) \text{ and } E(\mathbf{X}^k) = E(T(\mathbf{X}^k)). \tag{3.17}$$

(b) Density and Distance measures.

For a frame $\mathbf{F}_t$, the *Global Density* is defined as the number of agents per unit area $\mathcal{D}(\mathbf{F}_t) = \frac{K_t}{\mathbf{A}(\mathbb{X})}$, with K_t the number of agents present at t and $\mathbf{A}(\mathbb{X})$ the spatial extent of $\mathbb{X}$, evaluated from the extreme x, y values. The *Local Density* measures the density in a neighborhood. Plaue et al. [PCBS11] infer it with a nearest-neighbour kernel estimator. For a point $\mathbf{x}_t$,

$$\rho(\mathbf{x}_t) = \frac{1}{2\pi} \sum_{i=1}^{K_t} \frac{1}{(\lambda d_t^i)^2} \exp\left\{\left(-\frac{\|\mathbf{x}_t^i - \mathbf{x}_t\|^2}{2(\lambda d_t^i)^2}\right)\right\}, \tag{3.18}$$

with $d_t^i = \min_{j\neq i} \|\mathbf{x}_t^i - \mathbf{x}_t^j\|$ the distance from i to its nearest neighbor and $\lambda > 0$ a smoothing parameter. ρ is used to evaluate a trajlet-wise local density indicator

$$L(\mathbf{X}^k) = \max_{t\in[\tau^k,\tau^k+\delta^k]} \rho(\mathbf{x}_t^{i_k}). \tag{3.19}$$

3.5 Experiments

In this section, we analyze some common HTP datasets in the light of the indicators presented in the previous section. In Table 3.1, we give statistics (location, number of agents, duration...) for the datasets we have chosen to evaluate. We gather the most commonly used in HTP evaluation (ETH, UCY, SDD in particular) and datasets coming from a variety of modalities (static cameras, drones, autonomous vehicles...), to include different species from the zoo of Section 3.2.1.

For those including very distinct sub-sequences, e.g., ETH, UCY, SDD, inD, and Bottleneck (also denoted by BN in the figures), we split them into their constituting sequences. Also, note that we have focused only on pedestrians (no cyclist nor cars). We also ruled out any dataset containing less than 100 trajectories (e.g., UCY Arxiepiskopi or PETS).

To analyze a dataset $\mathbb{X}$, we apply systematically the following preprocessing

1. projection to world coordinates, when necessary,
2. down-sampling the annotations to a 2-3 fps framerate,
3. application of a Kalman smoothing with a constant acceleration model,
4. splitting of the resulting trajectories into trajlets $\mathbf{X}^k$ of length $\Delta = 4.8$s and filtering out trajlets shorter than 1m.

3.5.1 Overall description of the set of trajlets

For the indicators of Section 3.4.2, we have chosen $h = 0.5m$ for the Gaussian in the kernel-based density estimation; the number of samples used to evaluate the entropy is $M = 30$; the maximal number of clusters when clustering unconditional or conditional trajectories distributions is 21. In Fig. 3.3, we plot the distributions of the overall entropy and number of clusters, at different progression rates along the dataset trajectories. Without surprise, higher entropy values are observed for the less structured datasets (without main directed flows) such as SDD or inD. The number of clusters follows a similar trend, indicating possible multi-modality.

3.5.2 Predictability indicators

In Fig. 3.4, we depict the values of $H(\mathbf{X}^k)$, with one dot per trajlet $\mathbf{X}^k$. Interestingly, excepting the Bottleneck sequences, where high density generates randomness, the support for the entropy distributions are similar among datasets. What probably makes the difference are the tails in these distributions: large lower tails indicate high proportions of easy-to-predict trajlets, while large upper tails indicate high proportions of hard-to-predict trajlets.

Table 3.1 – General statistics of assessed HTP datasets. The columns present the type of location where the data is collected, the acquisition means, number of annotated pedestrians, the -rounded- duration (in minute or hour), the total duration of all trajectories, number of trajlets, and percent of non-static trajlets, respectively.

Dataset		Location	Acquisition	#peds	duration	total duration	#trajlets	non-stationary
ETH	Univ	univ entrance	top-view cam	360	13m	1h	823	93%
	Hotel	urban street		390	12m	0.7h	484	66%
UCY	Zara	urban street	top-view cam	489	18m	2.1h	2130	75%
	Students	univ campus		967	11.5m	4.5h	4702	96%
SDD	Coupa	univ campus	drone cam	297	26m	4.5h	5,394	41%
	Bookstore			896	56m	9.5h	11,239	54%
	DeathCircle			917	22.3m	4.2h	8,288	62%
inD	Loc(1)	urban intersection	drone cam	800	180m	7.1h	8302	94%
	Loc(2)			2.1**k**	240m	18h	21234	95%
Bottleneck	1D Flow(w=180)	simulated corridor	top-view cam	170	1.3m	1h	940	99%
	2D Flow(w=160)			309	1.3m	1.5h	1552	100%
Edinburgh	Sep{1,2,4,5,6,10}	univ forum	top-view cam	1.2**k**	9h	3h	2124	83%
GC Station	-	train station	surveillance cam	17**k**	1.1h	79h	76866	99%
Wild-Track	-	univ campus	multi-cam	312	3.3m	1.3h	1215	57%
KITTI	-	urban streets	lidar& multi-cam	142	5.8m	0.3h	253	93%
LCas	Minerva	univ-indoor	lidar	878	11m	4.8h	3553	83%

Table 3.2 – The list of proposed indicators for benchmarking HTP datasets

Overall description	Entropy $H_t(\mathbb{X}^k)$ and clusters $M_t(\mathbb{X})$ (section 3.4.1).
Predictability	Cond. entropy $H(\mathbf{X}^k)$ (Eq. 3.5).
Regularity	Speed $S^{avg}(\mathbf{X}^k), S^{rg}(\mathbf{X}^k)$ (Eq. 3.6). Acceleration $A^{avg}(\mathbf{X}^k), A^{max}(\mathbf{X}^k)$ (Eq. 3.8). Efficiency $F(\mathbf{X}^k)$ (Eq. 3.10). Angular deviation $D(\mathbf{X}^k)$ (Eq. 3.12).
Context	Closest approach $C(\mathbf{X}^k)$ (Eq. 3.14). Time-to-collision $T(\mathbf{X}^k)$, energy $E(\mathbf{X}^k)$ (Eq. 3.17). Local density $L(\mathbf{X}^k)$ (Eq. 3.19).

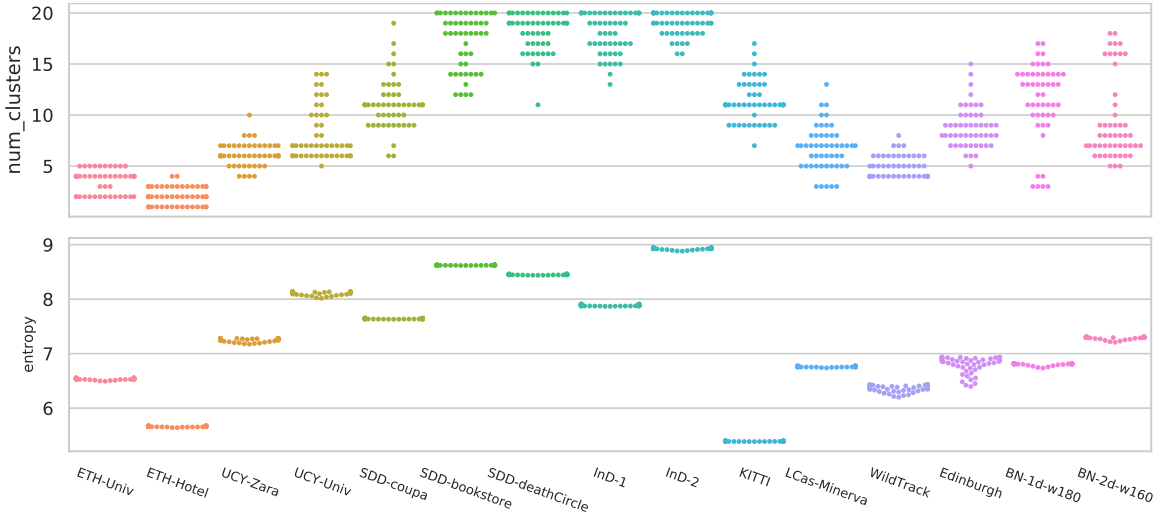


Figure 3.3 – Entropy $H_t(\mathbb{X})$ and number of clusters $M_t(\mathbb{X})$, as described in Section 3.4.1, at different progression rates t, for a dataset $\mathbb{X}$. Each dot corresponds to one t.

3.5.3 Regularity indicators

In Fig. 3.5, we depict the distributions of the regularity indicators $S^{avg}(\mathbf{X}^k)$, $S^{rg}(\mathbf{X}^k)$, $A^{avg}(\mathbf{X}^k)$, $A^{max}(\mathbf{X}^k)$ from Eqs. 3.6 and 3.8. Speed averages are generally centered around 1 and $1.5m/s$. Disparities among datasets appear with speed variations and average accelerations: ETH or UCY Zara sequences do not exhibit large speed variations, e.g. compared to Wild Track. In Fig. 3.6a, we depict the path efficiency $F(\mathbf{X}^k)$ fro Eq. 3.10, and we observe that ETH, UCY paths tend to be straighter. More complex paths appear in Bottleneck, due to the interactions within the crowd, or in SDD-deathCircle, EIF, due to the environment complexity. In Fig. 3.6b, deviations $D_t(\mathbf{X}^k)$ are displayed for different progression rates along the trajectories, and reflect similar trends.

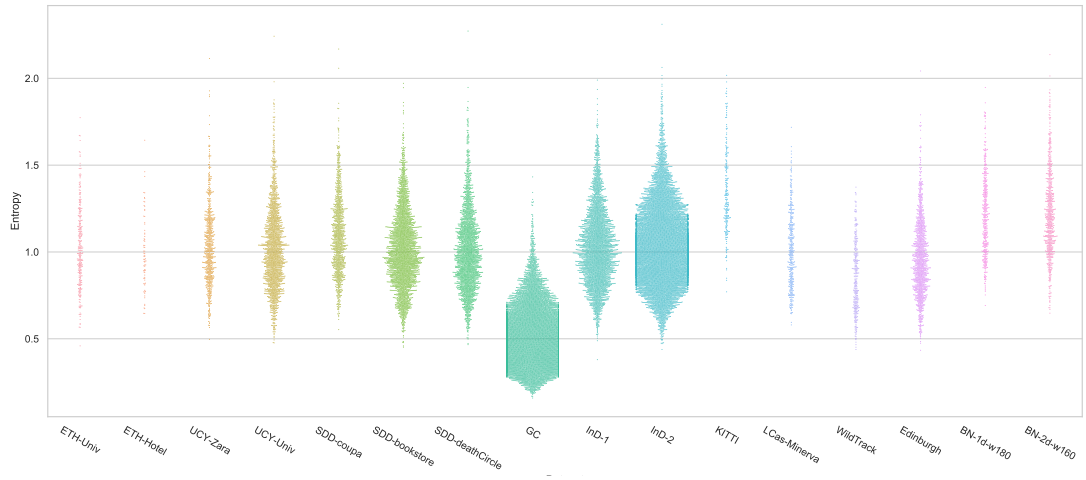


Figure 3.4 – Conditional Entropies $H(\mathbf{X}^k)$.

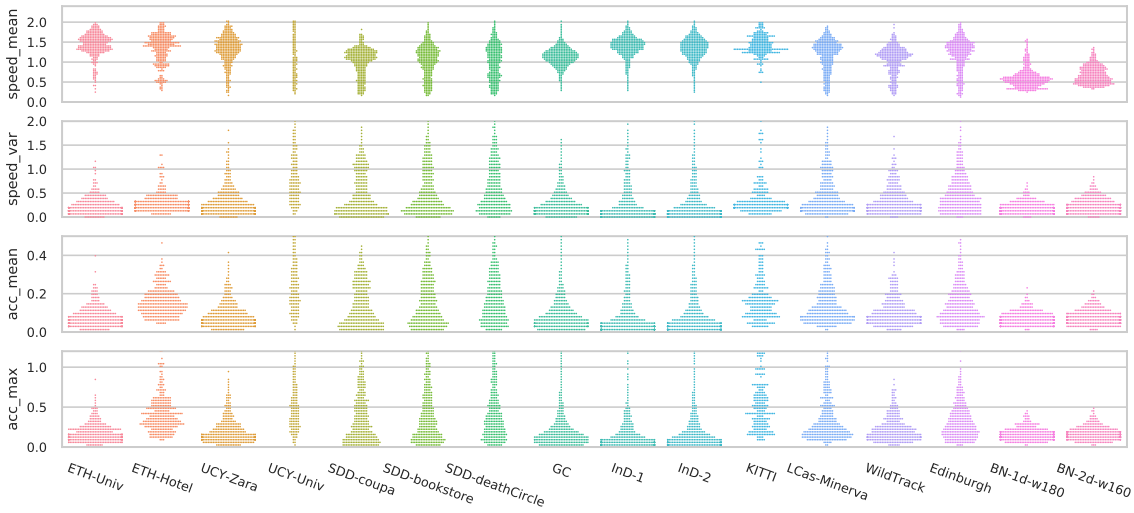
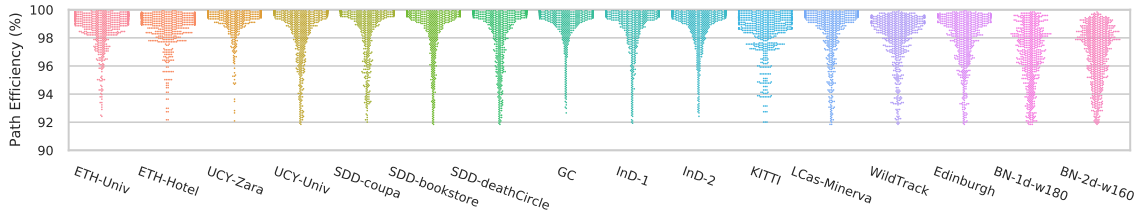


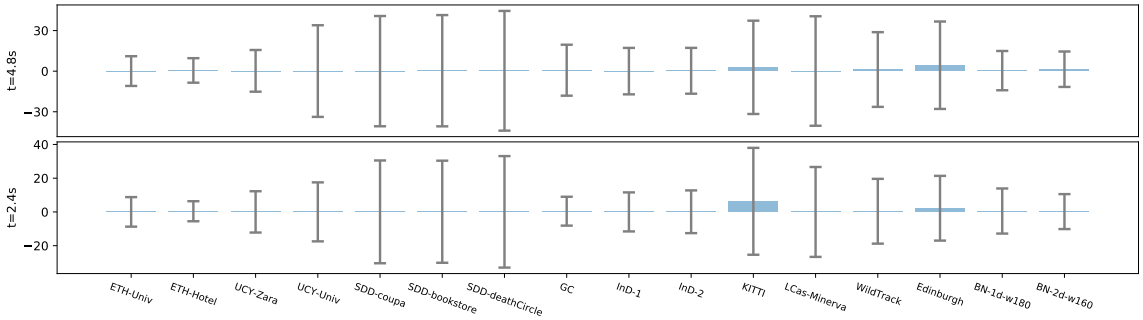
Figure 3.5 – Speed and acceleration indicators $S^{avg}(\mathbf{X}^k), S^{rg}(\mathbf{X}^k), A^{avg}(\mathbf{X}^k), A^{max}(\mathbf{X}^k)$. From top to bottom: speed means and variations, mean and max. accelerations.

3.5.4 Context complexity indicators

For estimating the TTC in Eq. 3.15, we set $R = 0.3m$, and for the interaction energy of Eq. 3.16, we set $k = 1$. The local density of Eq. 3.19 uses $\lambda = 1$. In Fig. 3.7, we display the collision avoidance-related indicators (TTC, DCA and interaction energy) described in Section 3.4.4, while in Fig. 3.8, we depict the density-related indicators. Most samples have low interaction energy, but interesting interaction levels are visible in Zara and InD. The global density for most datasets stays less than 0.1 p/m^2 while in InD(1-2), Edinburgh and SDD (Coupa & Bookstore),



(a) Path Efficiency. The higher, the closer to a straight line.



(b) Deviation from linear motion

Figure 3.6 – Regularity indicators: Path efficiency and deviation from linear motion.

it is even less than 0.02. Bottleneck (1d & 2d) are significantly high density scenarios. For this reason we depict them separately. Most natural trajectory datasets have a local density about $0-4p/m^2$ while such number is higher ($2-4p/m^2$) in Bottleneck. With both density indicators, a dataset such as WildTrack has a high global density and low local density, indicating a relatively sparse occupation. Conversely, low global density and high local density in Ind suggests the pedestrians are more clustered. This observation is also reflected in the interaction and entropy indicators as well.

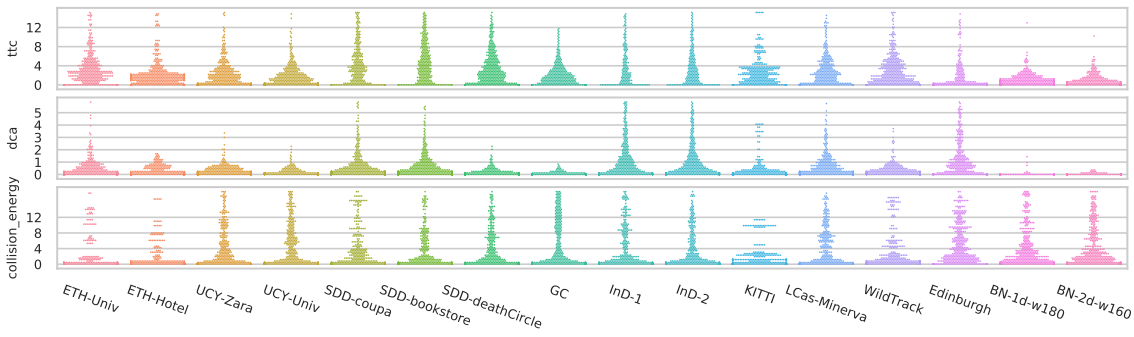


Figure 3.7 – Collision avoidance-related indicators: From top to bottom, time-to-collision, distance of closest approach and Interaction energy.

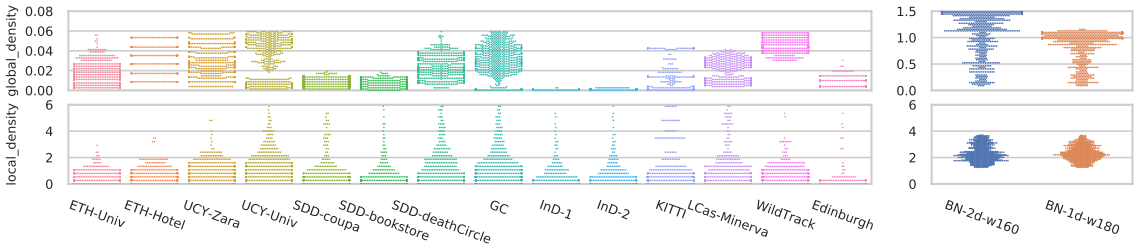


Figure 3.8 – Density indicators: On the top, global density (one data point for each frame); on the bottom, local density (one data point for each trajlet).

3.6 Discussion

Among the findings from the previous Section, Fig. 3.4 shows that the predictability among most datasets varies in mostly the same ranges. Regarding the motion properties of the datasets (see Fig. 3.5), another finding is pedestrians' average speed, which, in most cases, varies from 1.0 to 1.5 m/s. However, this is not the case for Bottleneck dataset, because the high density of the crowd does not allow the pedestrians to move with a 'normal' speed. In the SDD dataset, we observe multiple pedestrians strolling the campus. As shown in Fig. 3.6b these low-speed motions are usually associated with high deviation from linear motion, though part of this effect is related to the complexity of the scene layout.

Also, for most of the datasets, the speed variation of trajlets remains almost below 0.5. This is not a true hypothesis for LCas and WildTrack. As one would expect, the distribution of mean/max acceleration of trajlets is highly correlated with speed variations. In Fig. 3.6a we see that almost all values are bigger than 90%. For Bottleneck we see this phenomenon, where by increasing the crowd density and decreasing crowd speed, the paths become less efficient.

3.7 Conclusions & Future Work

We have presented in this chapter a series of indicators for gaining insight into the intrinsic complexity of Human Trajectory Prediction datasets. These indicators cover concepts such as trajectory predictability and regularity, and complexity in the level of inter-pedestrian interactions. In the light of these indicators, datasets commonly used in HTP exhibit very different characteristics. In particular, it may explain why predictions techniques that do not use explicit modeling of social interactions, and consider trajectories as independent processes, may be rather successful on datasets where e.g., most trajectories have low collision energy; it may also indicate that some of the more recent datasets with higher levels of density and interaction between agents could provide more reliable information on the quality of the prediction algorithm. Finally, the trajlet-wise analysis presented here opens the door to some evolution in

benchmarking processes, as we could evaluate scores by re-weighting the target trajlets in the function of the presented indicators.

Chapter 4

Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories with GANs

4.1 Introduction

In this chapter, we address the problem of learning multi-modal distributions of pedestrian trajectories, by proposing a new model based on Generative Adversarial Networks (GANs). The HTP problem is quite a complicated problem to solve. First, because there are many variables which are strongly relevant for the trajectories of single pedestrians: The nature of the surrounding obstacles and their spatial distribution, the nature of the ground, the long-term goal of the pedestrian, his age, his mental state, etc. Then, to make things even more difficult, the motions of a whole set of agents sharing a common space are dependent, through a whole range of interactions that can go from avoidance to meeting intention or person following. A number of interesting studies from neuroscience and bio-mechanics have isolated single factors or optimization principles governing the human motion in very specific contexts (one-to-one interactions, well-stated goals...). However, in more general cases, one may rapidly attain the limits of hand-tailored mathematical models. This has motivated the pursuit of more flexible, data-driven statistical approaches that can automatically select the most relevant features for explaining pedestrians walks, and that can benefit from the great efficiency of machine learning techniques.

The work presented in this chapter belongs to the aforementioned category of data-driven methods (see Chapter 2) for predicting the motion of pedestrians in the horizon of a few seconds, given a set of observations of their own past motion and of those of the pedestrians sharing the same space, as illustrated in Fig. 4.1. It relies on a Generative Adversarial Network (GAN)-based trajectory sampler to propose plausible future trajectories. It naturally encompasses the uncertainty and the potential multi-modality of the pedestrian steering decision, which is of

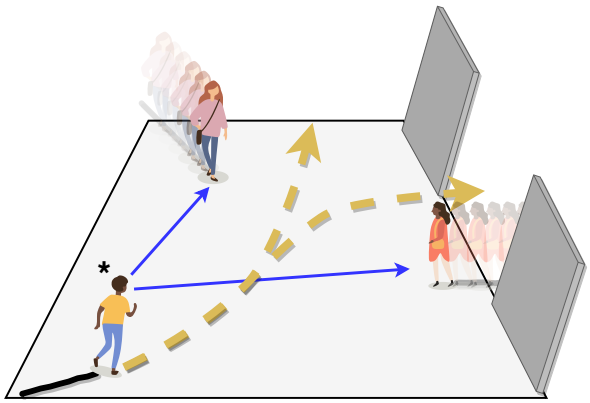


Figure 4.1 – Illustration of the multi-modal trajectory prediction problem. Having the observed trajectories of a pedestrian of interest, here shown with a star, and the ones of other pedestrians in the environment, the system should be able to build a predictive distribution of possible trajectories (here with two modes in dashed yellow lines).

critical importance when using this predictive distribution as a belief in higher level decision-making processes.

The main contributions of this work are the following:

— An efficient, unsupervised process to train a trajectory prediction GAN architecture based on Info-GAN [CDH+16], without L2 loss, which gives better results than previous works [GJFF+18, SKS+18] in preserving the multi-modal nature of the predictive distribution.

— The definition of an attention-based pooling scheme that relies on a few hand-designed interaction features inspired from the neuroscience/bio-mechanics literature, as a form of prior; the best way to combine them to assess the interaction is learned by our system.

— The design of a synthetic dataset specifically oriented to the evaluation of the preservation of multi-modality in trajectories predictive distributions.

Our architecture is described in Fig. 4.2. It adopts a new strategy to produce plausible samples for an agent from the joint predictive distribution of the set of agents. Our Sampler (Fig. 4.2 and Section 4.2.2) is trained to generate plausible predictions for a single agent, given past observations of trajectories for the whole set of the agents.

4.2 Problem statement and system overview

4.2.1 Notations and problem formulation

In the following, we use indices $i, j \in \{1, ..., N\}$ to refer to pedestrians, where N is the total number of pedestrians; a single observation of pedestrian i in the scene at time t is denoted by the 4×1 vector $\mathbf{x}_t^i$, which itself contains the position $\mathbf{p}_t^i$ and velocity $\mathbf{v}_t^i$ of the pedestrian:

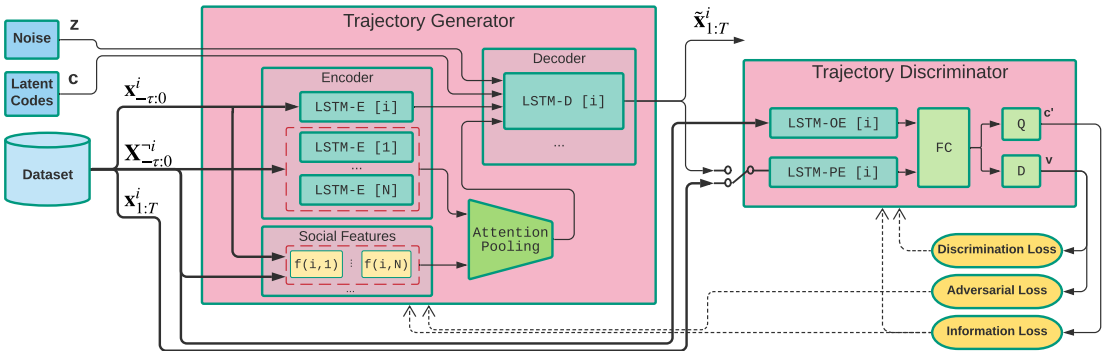


Figure 4.2 – Block Diagram of the Social Ways prediction system. The yellow ellipses represent loss calculations. The dashed arrows show the backpropagation directions. The bold arrows carry ground truth data.

$\mathbf{x}^i_t \triangleq ((\mathbf{p}^i_t)^T, (\mathbf{v}^i_t)^T)^T$. We assume that we have access to $\tau + 1$ consecutive observed samples $\mathbf{x}^i_{-\tau:0}$ of the pedestrians trajectory for each $i \in \{1, ..., N\}$. We also handle the set of observed samples of all pedestrians except i with $\mathbf{X}^{\neg i}_{-\tau:0} \triangleq \{\mathbf{x}^j_{-\tau:0} | j \in \{1, ..., N\}, j \neq i\}$.

The problem is then to predict the trajectories of each pedestrian for the next T time steps, i.e. $\mathbf{x}^i_{1:T}$. The rationale behind our approach is the following: When deciding his steering actions, a pedestrian anticipates likely scenarios about the evolution of his surrounding in the near future. Now, this anticipation may not be always very easy, because of the uncertainties in the neighbors future motion and intentions. In NN-based motion prediction systems [VMO18, XPG18, PPS+18], the input is taken as the set of most recent observations of the surrounding pedestrians. Hence, the mappings from observations to predicted trajectories built through the networks do not consider explicitly the uncertain and multimodal nature of the neighbors future trajectories, and, in a way, the network is expected to learn it too, which may be too much to expect.

4.2.2 GAN-based Individual Trajectory Sampler

Our Social Ways GAN generates independent random trajectory samples that mimic the distribution of trajectories among our training data, conditioned on observed initial tracklets of duration τ for all the agents in the scene. This system is depicted in Fig. 4.2. It takes as an input the observed trajectories of N pedestrians, $\mathbf{X}_{-\tau:0}$ and a random vector $\mathbf{z}$ sampled from a fixed distribution p_z. It samples a plausible trajectory $\tilde{\mathbf{x}}^{i,k}_{1:T}$ for agent i for the next T time steps, where k identifies one generated sample. The network should learn the whereabouts of an agent altogether with the impact a surrounding crowd has on its trajectory.

A GAN contains two components that act in opposition to each other during the training phase [GPAM+14]. The Discriminator D is trained to detect fake samples from real ones, while

the Generator G should produce new samples that fool the Discriminator and confuse its predictions. In a conditional version, both the Generator and the Discriminator are conditioned on some given data. Here, our GAN is conditioned on recent observations $\mathbf{x}^i_{-\tau:0}$, for agent i, and $\mathbf{X}^{\neg i}_{-\tau:0}$, for the other agents, and the Generator uses a noise vector $\mathbf{z}$ to complete $\mathbf{x}^i_{-\tau:0}$ into a full trajectory $G(\mathbf{z}|\mathbf{x}^i_{-\tau:0}, \mathbf{X}^{\neg i}_{-\tau:0})$.

4.2.3 Description of the Generator network

Our system shares a number of characteristics with existing trajectory generation systems [GJFF+18, SKS+18, KSMM+19] but it also includes critical novelties. The Generator network uses one LSTM layer (denoted as LSTM-E) to learn the temporal features along trajectories. The encoding of past trajectories $\mathbf{x}^i_{-\tau:0}$ for an agent is similar to [GJFF+18]. The LSTM-E cell encodes the history of the agent i through the recursive application of:

$$\mathbf{h}^i_t = \lambda^e(\mathbf{h}^i_{t-1}, \mu(\mathbf{x}^i_t; \mathbf{W}_\mu); \mathbf{W}_{\lambda^e}), \tag{4.1}$$

with $t \in [-\tau, 0]$, μ a linear embedding of the agent state and λ^e the cell of LSTM-E. $\mathbf{h}^i_t$ is the hidden state vector in LSTM-E at time t. It is depicted at the left part of Fig. 4.2.

For the decoding process and the generation of samples, we apply a similar process through another LSTM layer (denoted as LSTM-D) with hidden state $\mathbf{k}^i_t$

$$\mathbf{k}^i_t = \lambda^d(\mathbf{k}^i_{t-1}, \mathbf{o}^i_{t-1}; \mathbf{W}_{\lambda^d}), \tag{4.2}$$

with $t \in [1, T]$ and λ^d the decoding LSTM-D layer. The input vector is:

$$\mathbf{o}^i_t = [(\mathbf{h}^i_t)^T, (\sum_{j \neq i} a^{ij} \mathbf{h}^j_t)^T, (\mathbf{c})^T, (\mathbf{z})^T]^T. \tag{4.3}$$

It stacks information from the encoded history of observations of agent i up to t, $\mathbf{h}^i_t$, from the noise vector $\mathbf{z}$, and from the impact of future trajectories of the neighboring agents j, $\sum_{j \neq i} a^{ij} \mathbf{h}^j_t$. The construction of this term is described hereafter.

4.2.4 Social Ways: Attention pooling

The influence of the other agents on agent i is evaluated by encoding the vector $\mathbf{X}^{\neg i}_{1:T}$, through LSTM-E, and by applying an attention weighting process that produces weights $\mathbf{a}^i \triangleq [a^{i1}, .., a^{ij}, ..., a^{iN}]^T$ for agent i. They are defined as in [SKS+18], for $j \neq i$, based on pre-defined geometric features $\delta^{ij} \in \mathbb{R}^3$ stacking (1) the Euclidean distance between agents i and j, (2) the bearing angle of agent j from agent i (i.e. the angle between the velocity vector of agent i and the vector joining agents i and j), and (3) the distance of closest approach (i.e. the smallest

distance two agents would reach in the future if both maintain their current velocity) [KSFG14].

An interaction feature vector between agents i and j is defined as an embedding in $\mathbb{R}^{d_\sigma}$ of the social features δ^{ij}, through a Fully-Connected (FC) layer $\mathbf{f}^{ij} = \phi(\delta^{ij}; \mathbf{W}_\phi)$. Finally, the attention weights are obtained with the following scalar products and softmax operations between the hidden history vectors $\mathbf{h}^k$ and the interaction feature vectors $\mathbf{f}^{ik}$

$$\sigma(\mathbf{f}^{ik}, \mathbf{h}^k) = \frac{N-1}{\sqrt{d_\sigma}} < \mathbf{f}^{ik}, \mathbf{W}_\sigma \mathbf{h}^k >, \tag{4.4}$$

$$a^{ij} = \frac{\exp(\sigma(\mathbf{f}^{ij}, \mathbf{h}^j))}{\sum_{k \neq i} \exp(\sigma(\mathbf{f}^{ik}, \mathbf{h}^k))} \tag{4.5}$$

where d_σ is the common number of rows of the embedded features $\mathbf{f}$ and of the linear mapping $\mathbf{W}_\sigma$ applied on the hidden features.

4.2.5 Discriminator

The Discriminator is described on the right part of Fig. 4.2. It contains two encoding LSTM layers, one (applied $\tau + 1$ times) for observations, and one (applied T times) for predictions, and 2 FC layers to predict the samples labels. It takes as an input either a composite candidate trajectories for agent i, $[\mathbf{x}^i_{-\tau:0}, \tilde{\mathbf{x}}^{i,k}_{1:T}]$, or a ground truth trajectory, $[\mathbf{x}^i_{-\tau:T}]$, and outputs a probability for any of them to have been taken as a sample from the data.

4.2.6 Training the GAN

GAN training is known to be hard, as it may not converge, exhibit vanishing gradients when there is imbalance between the Generator and the Discriminator, or may be subject to mode collapsing, i.e. sampling of synthetic data without diversity. When predicting pedestrian motion, it is critical to avoid mode collapsing, as it could result in catastrophic decisions, i.e. for an autonomous driving agent.

Here, we have introduced two major changes in the GAN training. First, we do not use, as in other stochastic prediction methods [GJFF+18, SKS+18], an L2 loss term $\|G(\mathbf{z}|\mathbf{x}^i_{-\tau:0}, \mathbf{X}^{\neg i}_{-\tau:0}) - \mathbf{x}^i_{-\tau:T}\|^2$ enforcing the generated samples to be close to the true data, because we have observed negative impact of this term in the diversity of the generated samples.

Also, we have implemented an Info-GAN [CDH+16] architecture, which, as we will see in the experimental results section, has a very positive impact on avoiding the mode collapsing problem with respect to other versions of GANs. Info-GAN learns disentangled representations of the sources of variation among the data, and does so by introducing a new coding variable c as an input (see Fig. 4.2). The training is performed by adding another term to maximize a lower bound of the mutual information between the distribution of c and the distribution of the

generated outputs, which requires training another sub-network $Q(c|\mathbf{x}_{1:T})$ (with parameters θ_Q) which serves as a surrogate to evaluate the likelihoods $p(c|\mathbf{x}_{1:T})$ over the generated data $\mathbf{x}_{1:T}$. The training optimization problem is written as:

$$\begin{aligned}&\min_{\theta_G,\theta_Q}\max_{\theta_D} V(\theta_G,\theta_Q,\theta_D) = \\ &\mathbb{E}_{p_{data}(\mathbf{x}^i_{-\tau:T})}[\log D(\mathbf{x}^i_{1:T}|\mathbf{x}^i_{-\tau:0};\theta_D)]+ \\ &\mathbb{E}_{p_z(\mathbf{z})}[\log(1-D(G(\mathbf{z}|\mathbf{x}^i_{-\tau:0},\mathbf{X}^{\neg i}_{-\tau:0};\theta_G);\theta_D))]- \\ &\lambda\mathbb{E}_{p(c),p_z(\mathbf{z})}[\log Q(c|G(\mathbf{z}|\mathbf{x}^i_{-\tau:0},\mathbf{X}^{\neg i}_{-\tau:0};\theta_G);\theta_Q)]\end{aligned} \tag{4.6}$$

where $\mathbf{z}$ is the noise input and c the new latent code.

4.3 Experimental results

4.3.1 Implementation details

We implemented our system using the PyTorch framework. First, note that all the internal FC layers of both the Generator and the Discriminator are associated to LeakyReLU activation functions, with slope 0.1.

The **Generator** comprises a first FC linear embedding μ of size 4×128, over positions and velocities. The Encoder block in Generator contains one LSTM layer of 128 units (LSTM-E). Using 2 continuous latent code, noise vector with length of 62, and pooling vectors of size 64, which totally gives a 256-d vector, the Decoder LSTM (LSTM-D's) then uses 128 LSTM units in one layer and 3 FC layers with size of 64, 32, 2 to decode the predictions. Weights are shared among LSTM layers with the same function.

The **Discriminator** uses two LSTM blocks (LSTM-OE and LSTM-PE) with hidden layers of size 128 to process both the observed trajectories (size $4 \times \tau + 4$) and the predicted/"future" trajectories (size $4 \times T$); these outputs are processed in parallel with two 64×64 FC layers. Then they are concatenated and fed to two separate FC blocks: soft-classifier (D) $[64 \times 1]$ and latent-code reconstructor $[64 \times 2]$ (Q). Finally, τ and T are set to 7 and 12 respectively.

In each dataset, we train the GAN network with the following hyper-parameters setting: mini-batch size 64, learning rate 0.001 for Generator and 0.0001 for Discriminator, momentum 0.9. The GAN is trained for 20000 epochs.

4.3.2 Datasets

For the evaluation of our approach, we use ETH [PESVG09] and UCY [LCL07] (See also Chapter 3). These datasets consist of real-world human trajectories. They are labeled manually at a rate of 2.5 fps. The ETH dataset contains 2 experiments (coined as ETH and Hotel) and the UCY dataset contains 3 experiments (ZARA01, ZARA02 and Univ). In order to evaluate

the prediction algorithm, each dataset is split into 5 subsets, where we train and validate our model on 4 sets and test on the remaining set.

4.3.3 Baseline Predictors and Accuracy Metrics

We consider two sets of baselines.

1. Deterministic prediction models, that generate one trajectory for each observation:
   - Linear: This is a simple constant velocity predictor.
   - S-Force: It uses an energy function based on Social Forces to optimize the next agent action. The function penalizes jerky movements, high minimum distance to other agents and so on. We use the version by Yamaguchi et al. [YBOB11], in which a term enforces the agent to stay close to the group it belongs to.
   - S-LSTM [AGR+16]: It associates each pedestrian to one LSTM unit (the Social-LSTM) and gathers the hidden states of neighboring pedestrians with a so-called social-pooling mechanism to perform the prediction.
2. Stochastic prediction models, that generate a set of samples from a surrogate of the predictive distribution:
   - Social-GAN: A GAN-based prediction [GJFF+18]. We consider the variants S-GAN-P and S-GAN, with and without a pooling mechanism, respectively.
   - SoPhie [SKS+18] which implements Social and Physical attention mechanism in a GAN predictor.

Similarly to previous works [GJFF+18, VMO18], we use the following metrics to evaluate the proposed system over the prediction on one testing data $\mathbf{x}^i_{-\tau:T}$:

1. Average Displacement Error (ADE), averaging Euclidean distances between ground truth and predicted positions over all time steps:

$$\text{ADE}(\mathbf{x}^i_{-\tau:T}) = \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{x}^i_t - \hat{\mathbf{x}}^i_t\|. \tag{4.7}$$

2. Final Displacement Error (FDE), i.e. Euclidean distance between the ground truth and predicted final position:

$$\text{FDE}(\mathbf{x}^i_{-\tau:T}) = \|\mathbf{x}^i_T - \hat{\mathbf{x}}^i_T\|. \tag{4.8}$$

Then, we evaluate the expectations of these errors over all the samples in our testing datasets. We observe $\tau = 8$ frames (3.2 seconds) and predict the next $T = 12$ frames (4.8 seconds). To evaluate stochastic models (that generate a set of samples), we use the methodology proposed

Table 4.1 – Comparison of prediction error of our proposed method (S-Ways) vs. deterministic and stochastic baseline methods.

		Deterministic Models			Stochastic Models			
	Dataset	Linear	S-Force	S-LSTM	S-GAN	S-GAN-P	SoPhie	S-Ways
ADE	**ETH**	**0.59**	0.67	1.09	0.68	0.77	0.70	**0.39**
	Hotel	**0.36**	0.52	0.79	0.47	0.44	0.76	**0.39**
	Univ	0.82	0.74	**0.67**	0.56	0.75	**0.54**	0.55
	ZARA01	0.44	**0.40**	0.47	0.34	0.35	**0.30**	0.44
	ZARA02	0.43	**0.40**	0.56	**0.31**	0.36	0.38	0.51
FDE	**ETH**	**1.22**	1.52	2.35	1.26	1.38	1.43	**0.64**
	Hotel	**0.64**	1.03	1.76	1.01	0.89	1.67	**0.66**
	Univ	1.68	**1.12**	1.40	**1.18**	1.50	1.24	1.31
	ZARA01	0.98	**0.60**	1.00	0.69	0.69	**0.63**	0.64
	ZARA02	0.95	**0.68**	1.17	**0.64**	0.72	0.78	0.92

in [GJFF+18]. We generate K samples and take the closest one to Ground truth for evaluation. Hereafter, we consider $K = 20$. The modified metrics are called k-ADE and k-FDE in some related work, but for simplicity we use ADE and FDE.

4.3.4 Evaluation of Prediction Errors

The average prediction errors for both ADE and FDE metrics are shown in Table 4.3.4. As it can be seen, the use of our approach leads to significantly lower prediction errors for the ETH and Hotel experiments, but not on the ZARA experiments. We attribute this behavior in that, in the ZARA experiments, the width of the waypath for pedestrians is significantly smaller than in the Hotel and ETH scenes. Hence, there is less variance in the trajectories. Our proposed system intrinsically tends to generate various samples that result in good performance with more complex scenes and non-linear trajectories.

Among the deterministic models, though Social-LSTM model uses a much more complex system than its counterparts, it fails to outperform the other baselines and as the authors in [GJFF+18] mention it, it needs a synthetic dataset as a second source of training to improve the system accuracy.

In Figure 4.3, we give qualitative examples of the outputs and intermediate elements in our approach. We generated 128 samples with our method and the predictive distribution are shown with magenta points. In most of the scenarios (including non-linear actions, collision avoidance and group behaviors), the distribution has a good coverage of the ground truth trajectories and also generates what seems to be plausible alternative trajectories.

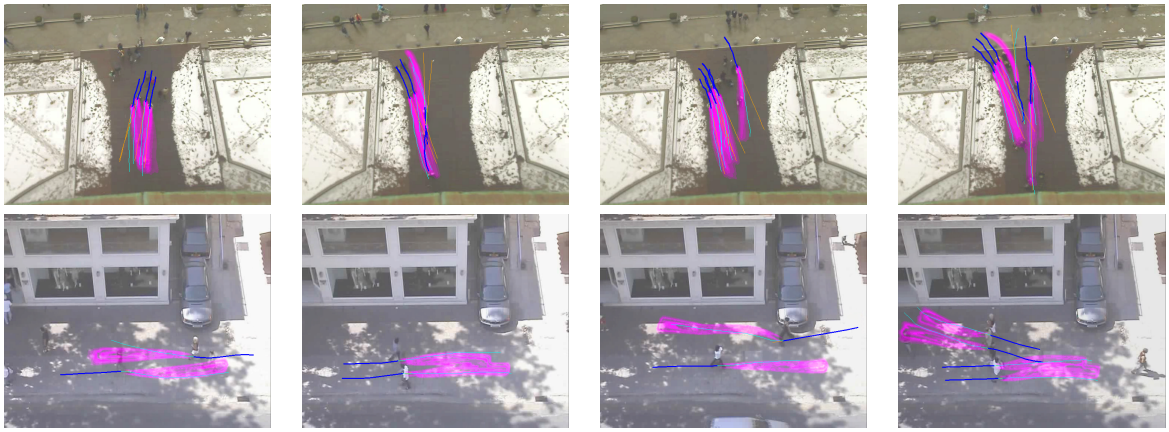


Figure 4.3 – Illustration of sample outputs of Social-Ways (in magenta color). The observed trajectories are shown in blue and ground truth prediction and constant-velocity predictions are shown in cyan and orange lines, respectively. [Best viewed in color.]

4.3.5 Quality of the Predictive Distributions

As commented in Section 4.2.2, our architecture and its training process are designed to preserve the modes of the predictive trajectory distribution. However, in all the datasets that we have tested, there are very few examples of clearly multi-modal predictive trajectory distributions. Hence, we have created a toy example dataset to study the mode collapsing problem with stochastic predictors.

This toy example is depicted in Fig. 4.4: Given an observed sub-trajectory (blue lines), the Generator should predict the rest of the trajectory (red lines). Each of the 6 groups represents one separate condition to the system ($\mathbf{x}^i_{-\tau:0}$), and each of the 3 sub-groups represents a different mode in the conditional distribution $p(\mathbf{x}^i_{1:T}|\mathbf{x}^i_{-\tau:0})$. Note that the interactions between agents are not considered here.

In order to compare our approach with other GAN-based techniques, we implemented several baselines. In all of them, the prediction architecture is the one we proposed without the attention-pooling; the GAN subsystem changes.

— **Vanilla-GAN**: This is simplest baseline, where the Generator is just trained with the adversarial loss.

— **L2-GAN**: In addition to adversarial loss, a L2 loss is added to the Generator optimizer.

— **S-GAN-V20**: The Variety loss proposed in Social-GAN method [GJFF+18] is added to the adversarial loss. This L2-loss only penalizes the closest prediction to ground truth among $V = 20$ predictions and gives more freedom to choose prediction samples.

— **Unrolled10**: Vanilla-GAN with the unrolling mechanism proposed in [MPPSD17]. The num-

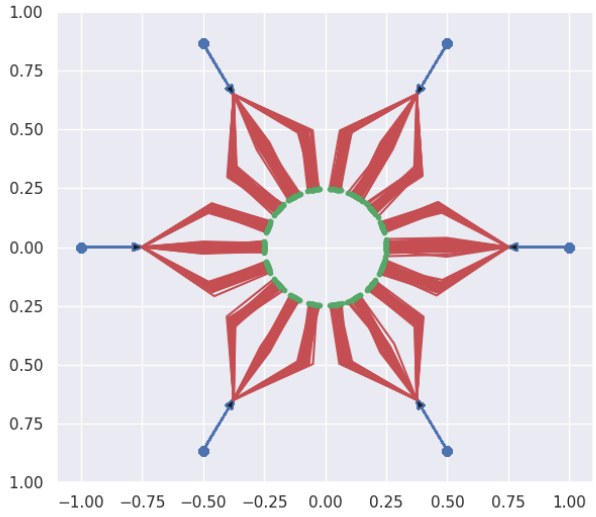


Figure 4.4 – The Toy Trajectory Dataset. There are six groups of trajectories, all starting from one specific point located along a circle (blue dots). When approaching the circle center, they split into 3 subgroups. Their endpoints are the green dots.

ber of unrolling steps is 10.

For each of the 6 possible observations, we generate 128 samples, which are depicted in Fig. 4.5. The Info-GAN together with Unrolled-GAN performs the best, with a slight advantage for Info-GAN, since almost all of the modes are preserved successfully after 90,000 iterations. At the same time, Vanilla-GAN, L2-GAN and S-GAN-V20 could not preserve the multi-modality of the predictions. One can see that using L2 loss, the model is converging faster than VanillaGAN and S-GAN-V20.

For a more quantitative evaluation of generative models, we have used the following two metrics to assess the set of fake trajectories versus the set of real samples [XHY+18]. Given two sets of samples $S_r = \{\mathbf{x}_r^i\}$ and $S_g = \{\mathbf{x}_g^j\}$ with $|S_r| = |S_g|$ and $\mathbf{x}_r^i \sim P_r$ and and $\mathbf{x}_g^j \sim P_g$:

1. A 1-Nearest Neighbor classifier, used in two-sample tests to assess whether two distributions are identical. We compute the leave-one-out accuracy of a 1-NN classifier trained on S_r and S_g with positive labels for S_r and negative labels for S_g. The classification accuracy for data from an ideal GAN should be close to 50% when $|S_r| = |S_g|$ is large enough. Values close to 100% mean that the generated samples are not close to real samples enough. Values close to 0% mean that the generated samples are exact copies of real samples, and that there is a lack of innovation in such system.

2. The Earth Mover's Distance (EMD) between the two distributions. It is computed as in

Eq. 4.9:

$$
\begin{aligned}
EMD(P_r, P_g) &= \min_{\mathbf{w} \in \mathbb{R}^{n \times m}} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{w}^{ij} d(\mathbf{x}_r^i,\ \mathbf{x}_g^j) \\
\text{s.t. } \forall i, j\ \mathbf{w}^{ij} \geq 0, &\sum_{k=1}^{m} \mathbf{w}^{ik} = \frac{1}{n}\ ,\ \sum_{k=1}^{n} \mathbf{w}^{kj} = \frac{1}{m}.
\end{aligned}
\tag{4.9}
$$

where $d()$ is called the ground distance. In our case we use the ADE of Eq. 4.7, between the future parts of the two trajectories.

We computed both 1-NN and EMD metrics on our toy dataset with $|S_r| = |S_g| = 20$, for each of the 6 observed trajectories. The results for different baselines are shown in Figures 4.6. We added evaluations for a few combinations of the aforementioned baselines (e.g., Info-GAN+unrolling steps or Unrolled+L2). The lower 1-NN accuracy of our approach using Info-GAN shows its higher performance for matching the target distribution, compared to Vanilla-GAN and other baselines. It is worth noting that the fluctuations in the accuracies are related to the small size of the set of samples. As it can be seen, Unrolled10 and Info+Unrolled5 have also better performances, while it is obvious that by adding L2 loss, the results are getting worse. The results of the EMD test also proves that both Info-GAN and Unrolled10 offer more stable predictors with lower distances between the fake and real samples. There is no evidence that the Variety loss offers better results than a Vanilla-GAN.

Moreover, on real trajectories, we have tested our algorithm on the Stanford Drone Dataset (SDD) [RSAS16]. In fact, we have used subsets of trajectories from two scenes (Hyang-6 and Gates-2). As you see in Fig. 4.7, with our system (left column), separate modes of the predictions appear clearly where the intuition would set them, while the Vanilla-GAN (right column) could not produce various paths.

4.4 Conclusions and Future Works

This chapter presents a novel approach for the prediction of pedestrians trajectories among crowds. It uses an Info-GAN to produce samples from the predictive distribution of individual trajectories, and integrates a few hand-designed interaction features inspired from the neuroscience/bio-mechanics literature, as a form of prior over the attention pooling process. We have shown through extensive evaluations on commonly used datasets that this approach partly improves the prediction accuracy of state-of-the-art methods on the datasets where the predictive distributions have the largest variances. We have also proposed a specifically designed

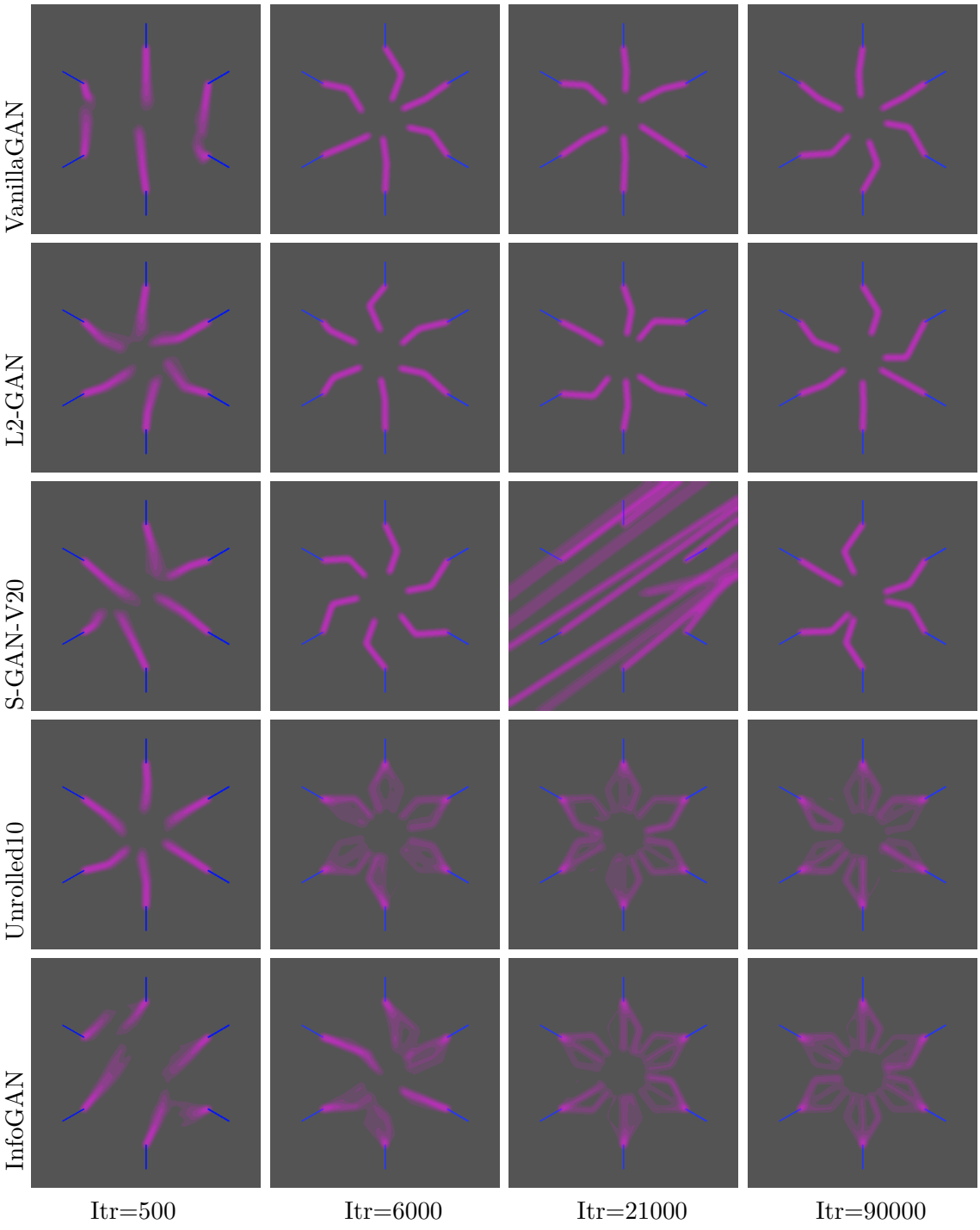


Figure 4.5 – Results of learning baselines on Toy Example, for different numbers of iterations. [Best viewed in color.]

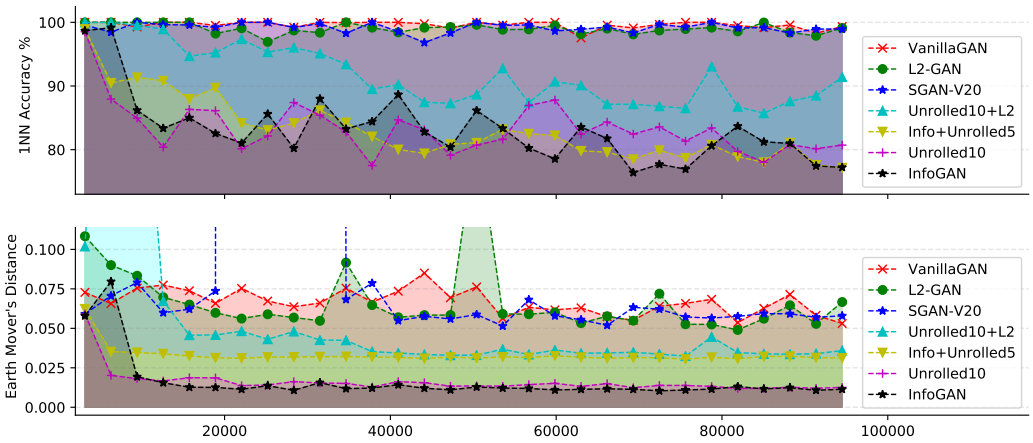


Figure 4.6 – Statistics for different GAN implementations over training iteration. Upper row: 1-NN accuracy metric (closer to %50 is better). Lower row: Earth Mover's Distance between generated and ground truth samples (the lower, the better).

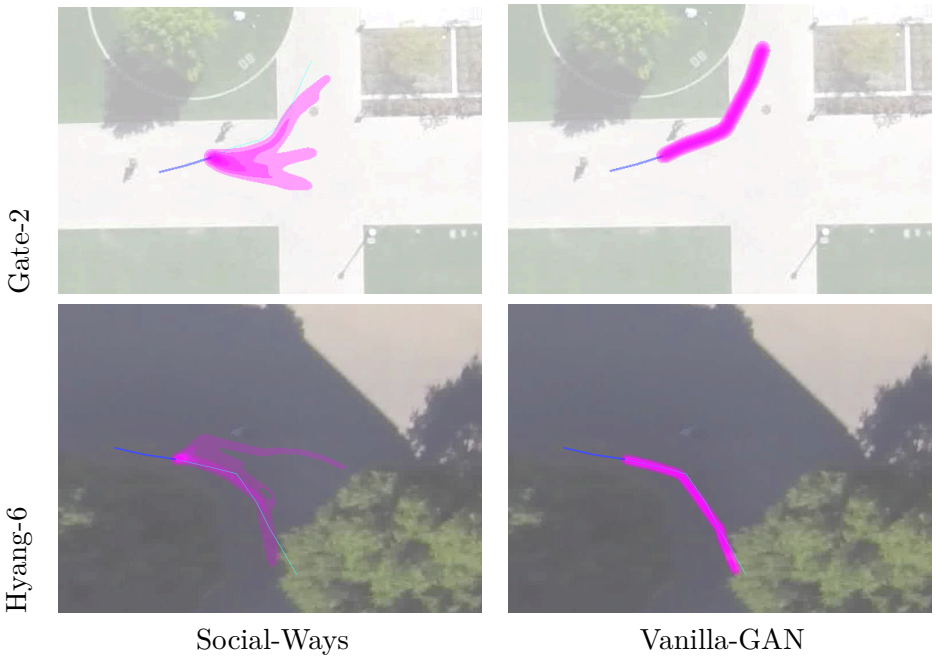


Figure 4.7 – Multi-modal trajectory predictive distributions on the SDD dataset: Social-Ways vs. Vanila-GAN. [Best viewed in color.]

dataset and an evaluation benchmark to show that Info-GANs achieve the best results in preserving multi-modality, compared with other variants. Finally, we are aware that is still room for improving the current generative models in pedestrian motion prediction and, above all, for exploiting these models in decision making.

Chapter 5

Imputing Occluded Crowd Structures from Robot Sensing

5.1 Introduction

As French economist Frédéric Bastiat pointed out back in the 19th century, to make wise decisions in the economy, "it is important to consider, in addition to what we see, what we do not see" (in French: ce qu'on voit et ce qu'on ne voit pas[Bas50]). We believe such an elementary rule that once has been neglected in some economic decisions, is missing from one of the most critical tasks of robots: navigation in crowded environments, where the on-board sensing of the crowd is typically limited by occlusions and can substantially impact the robot's ability to anticipate possible collisions with occluded agents.

In the last chapter we addressed the problem of human trajectory prediction. However, an equally important topic is crowd state imputation (i.e., filling out the missing data in the occupancy map around the robot) to tackle potentially unobserved surrounding agents and it has not received the same attention.

In this chapter, we address the problem of crowd state imputation from a mobile robot perspective. When dealing with this problem from the robot perspective, the input data have properties that makes the prediction much harder, e.g. as opposed to more classical motion prediction problem with data coming from a static, bird-view surveillance camera. In particular, due to the low height of the sensor (e.g. LiDAR) installed on the robot, noticeable parts of the crowd can be occluded. Moreover, many pedestrians may remain undetected for long sequences of frames. Depending on the density of the crowd and the characteristics of the robot's sensor, the proportion of non-detected people can be negligible or severe.

This can impact the performance of the robot in predicting future collisions and building a safe and valid motion plan. An example of typical crowd perception from a mobile robot, with multiple occlusions, is presented in Fig. 5.1 for a simulated mobile robot within a high-density crowd.

Our proposal is to leverage the statistical patterns extracted from past observations over a surrounding crowd to estimate the probability of the presence of people in unseen areas, i.e.

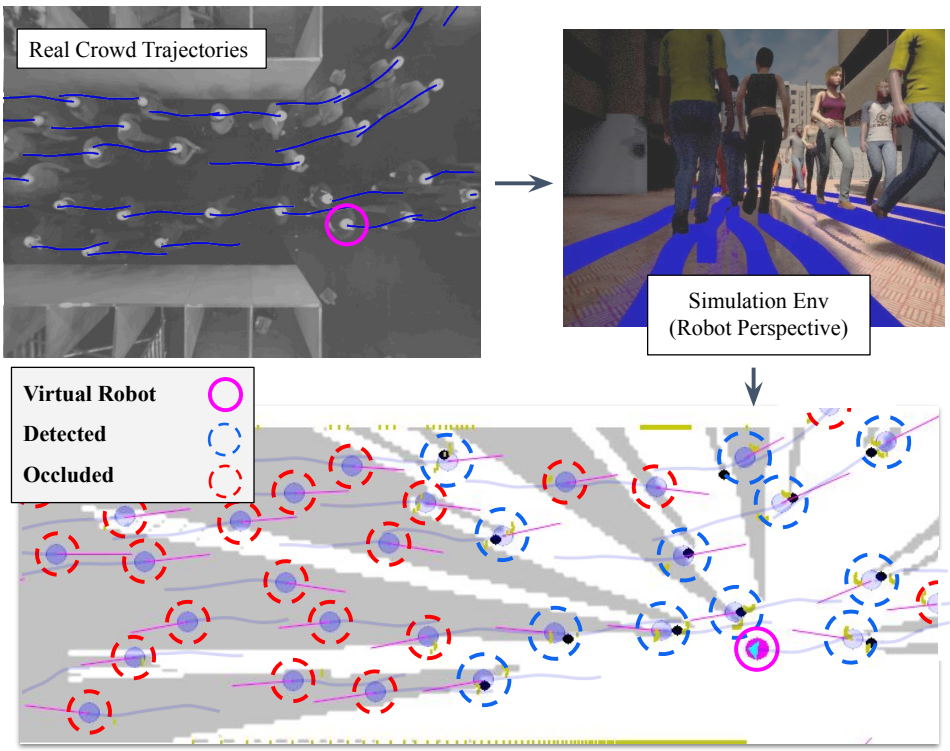


Figure 5.1 – Occluded crowd: In the top left, a top-view image of a crowd is shown. The pedestrians form a bidirectional flow in a narrow corridor. If we *simulate* a robot through one of the pedestrians (simulation of robot perspective in the top right image), there would be noticeable occlusion (gray area at the bottom) where the pedestrians are un-detected (red circles). The black dots within the blue circles show the detection locations, before filtering. Note: the crowd activity is recorded in the form of xy trajectories (blue lines).

we perform statistical imputation of the occupancy levels in these areas. Our model takes as input the stream of range data and the positions of the detected persons and gives as an output a prediction of the surrounding crowd motion. We state the problem as follows: "Given the robot observations about the crowd surrounding it and given a query point $\mathbf{x}$ (potentially out of robot's sight), what is the probability of presence of a person, that is not already detected, at $\mathbf{x}$?".

In most related research, if no person is detected/tracked at an unobserved location $\mathbf{x}$, then no mechanism allows anticipating a potential collision coming from an agent located there. Addressing this problem is the major contribution in this chapter. To the best of our knowledge, this is the first effort to use the statistical crowd patterns in order to predict the state of people around a robot.

This chapter is structured as follows: We review related work in section 5.2. Then we formalize

the problem and elaborate the details of the proposed method in section 5.3. In section 5.4, we present the evaluation methodology and experimental results on real and simulated crowd data. Finally, we conclude and discuss the future works in section 3.7.

5.2 Related Work

Prediction of static unknown spaces takes inspiration in neuroscience research. Cognitive psychologists have suggested that humans are able to explore new unknown environments by making predictions of the occupied space beyond their current line of sight [Buc10]. The authors of [KPP+19] study the ability to generate future predictions of occupancy maps using a U-Net style AutoEncoder neural network. Wang et al. [WYW+20] have proposed a neural-network-based method to predict the occupancy of the unknown space. Their model utilizes contextual information about the environment and learns from prior knowledge to predict the obstacles distribution in occluded spaces. The FASTER model [TLEH20] has shown that optimizing the robot local planner by considering both the known and unknown free spaces, can lead to higher-speeds and safer maneuvers for UAVs and ground robots.

Crowd Motion Prediction research focuses more on prediction on time domain, and is becoming a key building block for social robot navigation and self-driving vehicles. Social-LSTM [AGR+16] predicts the joint motion of dynamic pedestrians in crowded scenes by using LSTM networks and by pooling hidden states of neighboring agents. Social-GAN [GJFF+18] and Social-Ways [AHP19] were proposed later to cope with multi-modal predictive distributions of future trajectories, using Generative Adversarial Networks. In [PPS+18], both static obstacles and surrounding pedestrians are used for trajectory forecasting. In the above works, the prediction is performed under assumption of a fully-observable environment.

Other works have addressed the trajectory forecasting problem with first-person- (or robot-) view perception to deal with occlusions. The authors of [BZM+20] have created a simulation environment using Unity game engine, and have simulated the view of pedestrians in a crowd for prediction of trajectories. In [PSS+16], an interaction-aware motion model is learned based on human-human interactions observed by the robot with onboard sensors, but it is experimented only in very low-density scenarios.

The idea of extracting crowd patterns from collective behaviors has a long history. For example, Moussaid et. al [MPG+10] have studied the grouping behavior on crowd videos recorded in public places and have reported the patterns such as group sizes and the distance and angle between the group members. In [DTL17], the authors propose a classification of crowd structures inspired from fluid dynamics. A common mathematical framework to study these crowd structures is probabilistic graph models, and in particular Conditional Random Fields [PEVG10]. Alahi et. al [ARFF14] have created a huge dataset of human trajectories, recorded in a train

station, and have proposed *Social Affinity Maps (SAM)* to capture the spatial position of an agent's neighbors by radially binning the positions of his neighboring agents. In fact, they end up with a 10-bit binary radial histogram which is suggested to be a robust feature, not changing frequently and significantly over time. This feature improves the re-identification of groups in consequent *tracklet*s extracted from different cameras in the train station.

5.3 Proposed Method

Suppose that a robot navigates in an environment shared by n pedestrians. Each pedestrian state is described through its position $\mathbf{x}_i \in \mathbb{R}^2$ and instantaneous velocity $\mathbf{v}_i \in \mathbb{R}^2$. The robot uses its sensors to get raw measurements and passes them to a human detection unit that returns a set of m detections as $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_m\}$ (see Fig. 5.2), which are handled as noisy observations of $\{\mathbf{x}_1, ..., \mathbf{x}_m\}$. We assume there are $u = n - m$ unobserved pedestrians, either due to errors by the detector or because the pedestrians are not visible by the sensors, e.g. because of an occlusion or because they are out of the robot field of view.

Our algorithm leverages the information about geometric relations between the people in the surrounding crowd, extracted from previously observed trajectories, to infer a probabilistic occupancy map covering occluded regions and to impute the position of unobserved pedestrians. This way, we can improve the robot knowledge about the environment and build more reliable motion plans estimate and plan beyond what it can see. Before detailing our imputation algorithm, we first introduce the concept of Social Tie.

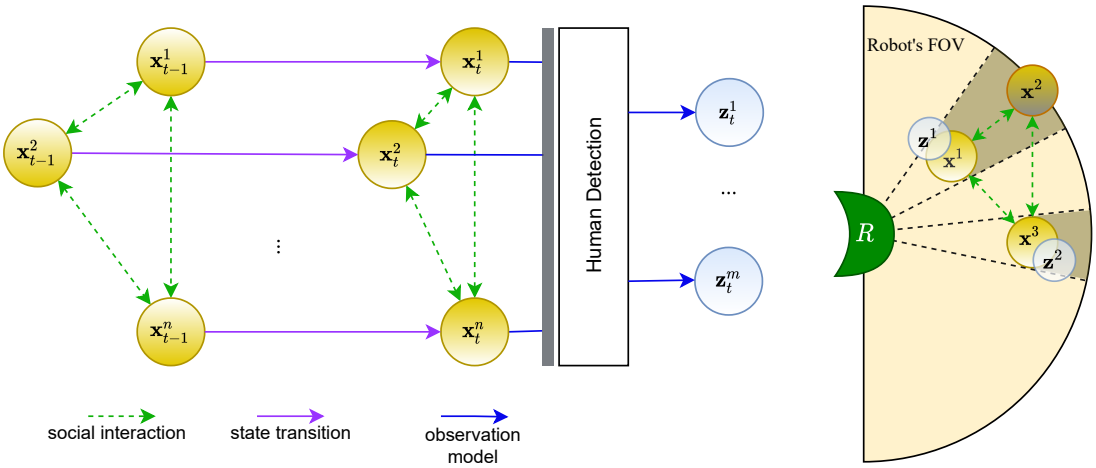


Figure 5.2 – Graphical representation for Occlusion- and Socially-aware Object Tracking. The yellow circles show the state of each agent at two consecutive time instants $t-1$ and t. The blue circles are the observations at current time instant t. The dashed green arrows represent the social interaction between the agents and the blue arrows represent the observation model.

5.3.1 Social Ties

As a modeling tool to understand the geometrical structure of the flows within crowds, we introduce the notion of Social Tie. Inspired by ideas from other works [ARFF14], we define a Social Tie as a displacement vector between a pair of agents, expressed in the local frame of the first one:

$$\delta(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{R}_i)^\top (\mathbf{x}_j - \mathbf{x}_i), \tag{5.1}$$

where $\mathbf{R}_i^t$ is the 2D rotation matrix giving the global orientation angle of agent i at time t.

We further classify the social ties as *strong ties* and *absent ties*. The terms are borrowed from social networks literature [Gra73] to represent, respectively, long-term interactions (e.g. grouping or leader-follower behaviors) versus instant interactions (e.g. collision avoidance overtaking) between pair of agents. The motivation behind this modeling choice is that these two categories (strong/absent) define two distributions of displacements δ, that are better treated dissimilarly.

The classification between the two tie types is based on the history of the distance between two agents. We keep the history of the distance vector between all pairs of detected agents. A tie is labeled as *strong*, if 1) during the last t_c time-steps, the two agents are closer than a threshold distance r_{max}, 2) in the same time interval, the Euclidean distance between the agents remains fixed (up to a tolerance threshold ϵ_l on distance variations) over the last second, and 3) the agents have similar orientations (within another threshold ϵ_θ). If 1) holds but either 2) or 3) does not, then the link is categorized as absent. While we found these simple rules to be enough in our setup, more advanced classification rules can also be used. An example is depicted in Fig. 5.3 to illustrate the classification process.

The above definition implies that a tie (strong or absent) is assigned to any pair of agents that have a distance lower than r_{max} during the last t_c time-steps. It is worth observing that the tie definition is symmetric: a strong (absent) tie from $\mathbf{x}_i$ to $\mathbf{x}_j$ implies a strong (absent) tie from $\mathbf{x}_j$ to $\mathbf{x}_i$. However this symmetry property is not directly implied by the definition of social tie, but by the classification rule, which means that one-way ties may also exist, by using other classification rules.

Next, we evaluate the distribution of social ties across agents by classifying and accumulating the observed ties and then taking a polar histogram for each. The two histograms represent the empirical distribution of the strong and absent ties. They are denoted by $p(\delta|\tau = S, \mathcal{H})$ and $p(\delta|\tau = A, \mathcal{H})$, respectively, where τ is the tie type (strong or absent) and $\mathcal{H}$ denotes the historical observations used for training. Note that these distributions on displacements give us access to the conditional distribution of the absolute location of a queried position $\mathbf{x}$, conditioned on seeing a pedestrian at $\mathbf{x}_i$, and on the type of social tie τ: $p(\mathbf{x}|\mathbf{x}_i, \tau, \mathcal{H})$.

Some examples of $p(\delta|\tau, \mathcal{H})$ are shown in Fig. 5.4, where one can observe that the different

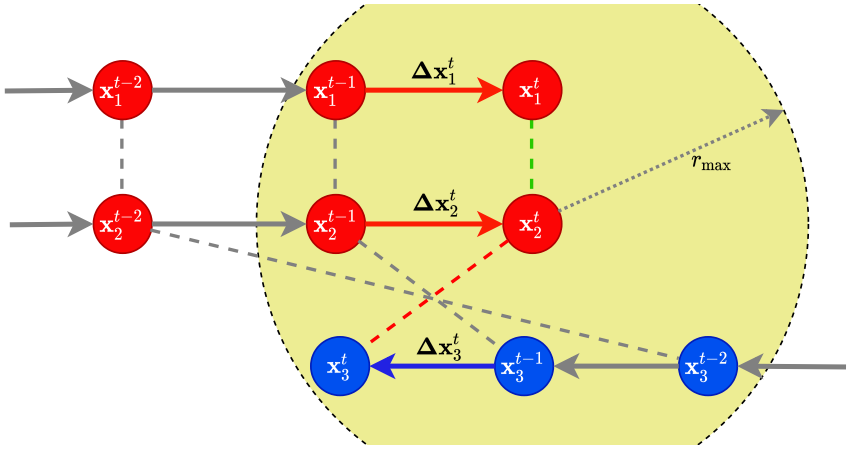


Figure 5.3 – Classification of Social Ties. The tie between $\mathbf{x}_1$ and $\mathbf{x}_2$ is classified as a *strong* tie at time t, (shown in green). On the other hand, the link between $\mathbf{x}_2$ and $\mathbf{x}_3$ is *absent* (red dashed line) .

collective patterns lead to significantly different social tie patterns in the HERMES-Bottleneck, SDD, ETH and Zara Datasets. For example, in Zara (upper right), the strong ties are mostly observed as side-by-side configurations, while in HERMES-Bottleneck (bottom right), the strong ties come mostly as lines of people (because of the nature of the dataset). We will discuss in the following how these patterns can help us in estimating the location of people in unseen (occluded) areas.

5.3.2 Communities (clusters)

We define *communities* (or groups) as subsets of pedestrians that are connected (directly or indirectly) by strong ties, i.e., that can be seen as clusters of people moving together as a group, or as a continuous flow of people moving in one direction (see examples in Fig. 5.5). The m observed agents are partitioned into K communities: $\mathbf{C} = \{\mathbf{c}_1, ..., \mathbf{c}_K\}$. Note that a community can be as small as containing only one person. The velocity of a community $\mathbf{c}_k$ is defined as the average velocity of its members:

$$\bar{\mathbf{v}}_k = \frac{1}{|\mathbf{c}_k|} \sum_{i \in \mathbf{c}_k} \mathbf{v}_i. \tag{5.2}$$

We also define a *territory* area for each community, by running a k-nearest neighbor classifier that assigns, to every location in the plane, a label that represents the nearest community. The probability of being, at a location $\mathbf{x}$, in the territory of $\mathbf{c}_k$ is denoted by $\phi_k(\mathbf{x})$. To take the orientation of each agent into account, the k-nearest neighbor is not implemented with the Euclidean distance but with a Mahalanobis distance $d(\mathbf{x}, \mathbf{y})$ with a 2×2 covariance matrix Σ chosen with its first eigenvector aligned with the agent velocity $\mathbf{v}_i$ and assigned to an eigenvalue $\alpha\|\mathbf{v}_i\| + \beta$ while the second one is assigned an eigenvalue β:

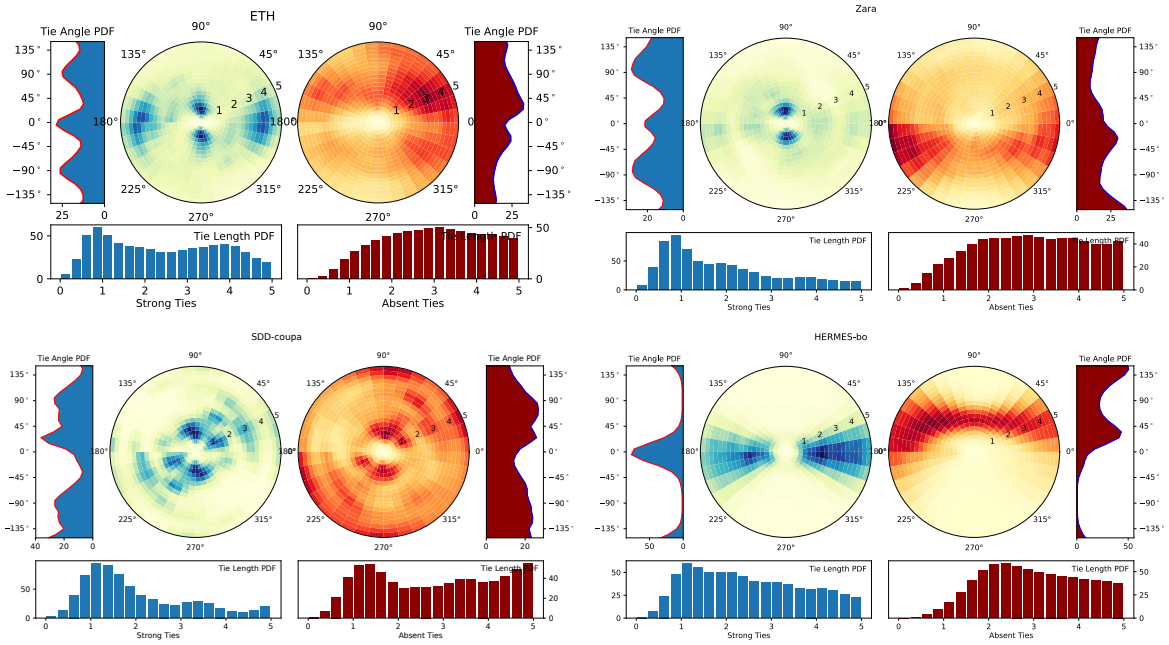


Figure 5.4 – Distributions $p(\delta|\tau = S, \mathcal{H})$ and $p(\delta|\tau = A, \mathcal{H})$ of *strong* (left side) and *absent* (right side) social ties for 4 different datasets. The different flow structures lead to very distinct distributions along the datasets. The graphs emphasize some structural elements of the crowd flow, such as the communities width (very small in the Hermes-Bottleneck case), the permanent asymmetries of the absent ties (Zara or Hermes-Bottleneck).

$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y}). \tag{5.3}$$

5.3.3 Imputing New Pedestrians

We have explained in 5.3.1 that the social ties are extracted from crowd activity observations. Here we leverage this data to predict plausible positions for the unobserved agents that might exist in the blind spot areas and beyond the field of view of the sensor. This idea is inspired by *inpainting* technique in Computer Graphics, where Pair Correlation Functions (PCF) [ENMGC19, NENMC20] are used to detect textures from an image and propagate them to other areas.

A PCF measures the probability density of the distance between pairs of particles. However, in our context, considering only the distance between agents is difficult, since the orientation of the agents is critical in modeling human collective activities [MPG+10]. Hence, we propose an extension of the concept of PCF with the distribution of social ties.

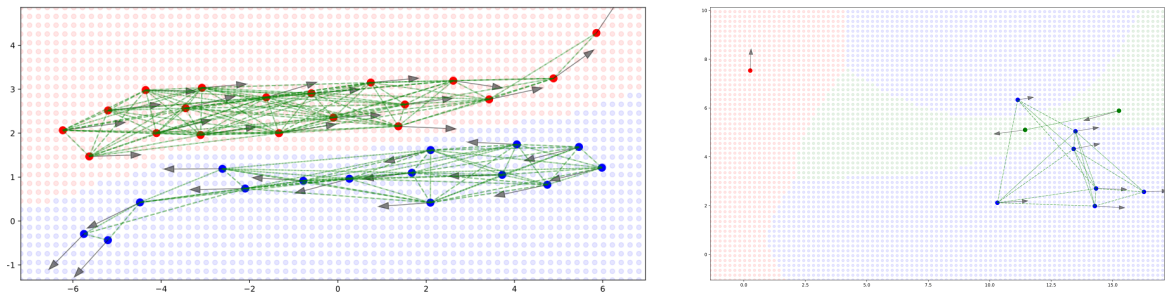


Figure 5.5 – Illustratoin of 'Communities' and 'Territories'. Green lines: strong ties, black arrows: velocity vector of the agents. Left: Bottleneck dataset, a bi-directional flow of pedestrians that form two communities, Right: A frame of UCY dataset (Zara) with 3 communities moving in different directions. The territory of each community is shown with different colors on a mesh.

We are interested in finding $p(o(\mathbf{x})|\mathbf{Z})$, the probability of the presence of an unobserved agent at a query point $\mathbf{x} \in \mathbb{R}^2$, given the m detected agents $\mathbf{Z}$.

By expressing this distribution as a marginalization over the two latent variables $v(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ for, respectively, the visibility from the robot at position $\mathbf{x}$ and the class indicating the community at that position, we can write this as (we removed the $\mathbf{x}$ for shortening equations, but be aware that the random variables o, c, v are all defined in a specific $\mathbf{x}$)

$$p(o|\mathbf{Z}) = \sum_{w\in\{0,1\}} \sum_{k} p(o, \mathbf{c} = \mathbf{c}_k, v = w|\mathbf{Z}) \tag{5.4}$$

$$= p(v = 0) \sum_{k} p(\mathbf{c} = \mathbf{c}_k) p(o|v = 0, \mathbf{c} = \mathbf{c}_k, \mathbf{Z}). \tag{5.5}$$

Using the community-dependent ties distributions, we estimate $p(o(\mathbf{x})|v(\mathbf{x}) = 0, \mathbf{c}(\mathbf{x}) = \mathbf{c}_k, \mathbf{Z})$ as follows:

$$\begin{aligned} p(o|v = 0, \mathbf{c} = \mathbf{c}_k, \mathbf{Z}) \propto & \prod_{\mathbf{z}_i \in \mathbf{Z}\cap\mathbf{c}^k} \int_{\mathbf{x}_i} p(\delta(\mathbf{x}_i, \mathbf{x})|\tau = S, \mathcal{H}) p(\mathbf{z}_i|\mathbf{x}_i) d\mathbf{x}_i \\ & \times \prod_{\mathbf{z}_j \in \mathbf{Z}\setminus\mathbf{c}^k} \int_{\mathbf{x}_j} p(\delta(\mathbf{x}_j, \mathbf{x})|\tau = A, \mathcal{H}) p(\mathbf{z}_j|\mathbf{x}_j) d\mathbf{x}_j \end{aligned} \tag{5.6}$$

where $p(\mathbf{z}_i|\mathbf{x}_i)$ is the sensor error model and $p(\delta|\tau, \mathcal{H})$ denotes the likelihood of a social tie δ (see Section 5.3.1). If the sensor model is Gaussian, the integrals above can be evaluated easily by using pre-filtered versions of the maps $p(\delta(\mathbf{x}_i, \mathbf{x})|\tau, \mathcal{H})$. This model factors the joint distribution through pairs of agents. In practice, we only consider the agents in $\mathbf{X}$ within a radius r_{max} from

$\mathbf{x}_q$. Rewriting Eq. (5.5) using the notation introduced above $\phi_k = p(\mathbf{c} = \mathbf{c}_k)$:

$$p(o|\mathbf{Z}) = p(v = 0) \sum_k \phi_k p(o|v = 0, \mathbf{c} = \mathbf{c}_k, \mathbf{Z}). \tag{5.7}$$

By interpreting this distribution as a likelihood function $q(\mathbf{x}) = p(o(\mathbf{x}) = 1|\mathbf{Z})$ over $\mathbf{x}$ and by discretizing the $\mathbf{x}$ along a regular grid, we can sample a new agent at location $\mathbf{x}_s$. We tested different sampling strategies, and found out the following sampling to be more effective: we draw a sample from $q(\mathbf{x})$ and then we search within a small disk r_s for the local maximum of q in this region. After this, a virtual agent is created at this location $\mathbf{x}_s$, and assigned to community $\arg\max_{\mathbf{c}_k} p(\mathbf{c}(\mathbf{x}_s) = \mathbf{c}_k)$, with velocity $\mathbf{v}_k$. By iterating the sampling, we create more agents in the occluded area. After each sampling iteration $\#i$, we add the sample $\mathbf{x}_s^{(i)}$ to $\mathbf{Z}^{(i-1)}$ (having $\mathbf{Z}^{(0)} \equiv \mathbf{Z}$) and update $q^{(i)}$ using Eq. (5.7). We repeat the process until $\max_{\mathbf{x}} q^{(i)}(\mathbf{x}) < \epsilon_s$ in a neighborhood of radius r_{nav} around the robot.

5.3.4 Sampling an Imputed Crowd

By repeating the sampling process, explained in the previous section, we obtain an augmented set $\mathbf{Z}^+$ of detected and virtual agents. We repeat this entire process, for H times to get multiple sets of hypotheses $\mathbf{Z}^{+(h)}$ for $h = 1...H$.

The pseudo-code for our proposed algorithm "Occlusion-Aware Crowd Imputation" can be found in Algo.1.

5.4 Experimental Results

In this section we first study the feasibility/importance of the proposed method on multiple trajectory datasets and then we discuss the results.

5.4.1 Implementation Details

We developed our algorithm using Python. For 3D simulation of the crowd motions and the robot perception, we used the CrowdBot Simulator, which is built on top of Unity game engine and ROS system. Perceptions are obtained from a simulated 360-degree LiDAR with a resolution of 0.5^o and working range of $[0.05m, 8m]$ installed at a height of $40cm$. We use DR-SPAAM [JHL20], a deep learning-based person detector that detects persons (legs) in 2D range data sequences. We couple the detector to a Constant-Acceleration Kalman-Filter to track multiple targets. In our algorithm, the gridmap has a resolution of 8 cells per meter.

The hyper-parameters of the algorithm are chosen as follows: $r_{max} = 5m$, $t_c = 1s$, $\epsilon_l = 0.5m$ and $\epsilon_\theta = 45^o$ for classifying the ties. The polar histogram used for the representation of strong and absent tie distributions have radial and angular resolution of $25cm$ ($[0 - 5m]$) and 10^o

Algorithm 1 Occlusion-Aware Crowd Imputation

Inputs: $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_m\}$ ▷ detected persons
Output: Sample sets $\{\mathbf{Z}^{+(h)}\}_h$

1: Compute Ties and Tie types between detections (Eq. 5.1)
2: Find Connected Components (Communities) $\{\mathbf{c}_1..\mathbf{c}_K\}$
3: Compute Territory of each Community $\{\phi_1..\phi_K\}$
4: Compute Velocity of each Community $\{\bar{\mathbf{v}}_1..\bar{\mathbf{v}}_K\}$
5: Initialize: $\forall \mathbf{x}\ q(\mathbf{x}) \leftarrow p(v(\mathbf{x}) = 0)$
6: **for** $\mathbf{x}$ **do**
7: $\quad k' \leftarrow \mathbf{c}(\mathbf{x})$
8: $\quad$ **for** $\mathbf{z}_i \in \mathbf{Z}$ **do**
9: $\quad\quad \delta \leftarrow \mathbf{R}_i^T(\mathbf{x} - \mathbf{z}_i)$
10: $\quad\quad k \leftarrow \mathbf{c}(\mathbf{z}_i)$
11: $\quad\quad$ **if** $k = k'$ **then**
12: $\quad\quad\quad q(o) \leftarrow q(o) * p(\delta|\tau = S, \mathcal{H})$
13: $\quad\quad$ **else**
14: $\quad\quad\quad q(o) \leftarrow q(o) * p(\delta|\tau = A\dagger, \mathcal{H})$
15: $\quad\quad$ **end if**
16: $\quad$ **end for**
17: **end for**
18: **for** $h \in 1..H$ **do**
19: $\quad \mathbf{Z}^{+(h)} \leftarrow \mathbf{Z}$
20: $\quad$ **while** $\max q(\mathbf{x}) > \epsilon_s$ **do**
21: $\quad\quad \mathbf{x}_s \leftarrow$ Sample virtual pedestrian
22: $\quad\quad$ add $\mathbf{x}_s$ to $\mathbf{Z}^{+(h)}$
23: $\quad\quad$ assign $\mathbf{x}_s$ to $\mathbf{c}_k = \arg\max_{\mathbf{c}} p(\mathbf{c}(\mathbf{x}_s) = \mathbf{c})$
24: $\quad\quad$ update $q(\mathbf{x})$ for all $\mathbf{x}$ (Lines 6:17)
25: $\quad$ **end while**
26: **end for**
27: Return $\{\mathbf{Z}^{+(h)}\}_h$

($[-180, 180^o]$) respectively, with a constant-padding $p = 1$ for any bin out of this range. Also, the coefficients α and β used in Eq. (5.3) are set to 0.2 and 0.1 respectively. ϵ_s is set to 50% and r_s (the radius of the search disk) is set to $2m$.

5.4.2 Real Crowd + Simulated Robot

To evaluate the proposed method and validate the system, it is critical to work with real datasets. However, to the best of our knowledge, there is no public robot-crowd datasets in which the ground truth annotations of occluded pedestrians are available. Hence, we use crowd-only datasets, select one random pedestrian in the crowd and replace it by a simulated robot. The robot traverses the same trajectory taken by the pedestrian. The motivation for not simulating the robot navigation is to make the system independent of the navigation algorithm and to be able to compare the performance against new methods.

5.4.3 Crowd Datasets

We consider multiple Human Trajectory datasets that cover different crowd densities and crowd structures:

— **SDD** or Stanford Drone Dataset contains trajectories of moving agents in the campus of Stanford University, mostly with low crowd densities [RSAS16].

— **ETH** is a small dataset of pedestrians entering/exiting the entrance of a university building [PESVG09].

— **Zara** is a dataset of crowd activity in sidewalk of a shopping street (subset of UCY [LCL07]).

— **Hermes** includes multiple high-density crowd controlled experiments (e.g., through Bottlenecks) in Unidirectional or Bidirectional settings [SPS+09].

We use the OpenTraj toolkit [AZC+20] to load and process HTP datasets. We split each dataset, into a training set (for extracting social patterns and tie distributions $p(\delta)$) and a testing set, using the 70:30 rule of thumb.

5.4.4 Occlusion Severity in Crowd

We have measure the severity of occlusions for multiple datasets, using the simulation explained above. This is done by repeating the simulation, each time replacing one pedestrian with the robot. Then we estimate the percentage of sensing rays that are occluded for each pedestrian. We have classified occlusion values into to 4 categories: Fully-Visible (0-15%), Partially-Occluded (15-50%), Largely-Occluded (50-85%) and Fully-Occluded (85-100%). The results are shown in Fig. 5.6. As expected, the HERMES sequences exhibit severe occlusions, while ETH/Zara do

not. We will see in the next sections that our crowd imputation algorithm does not improve the baseline in low/medium-density situations.

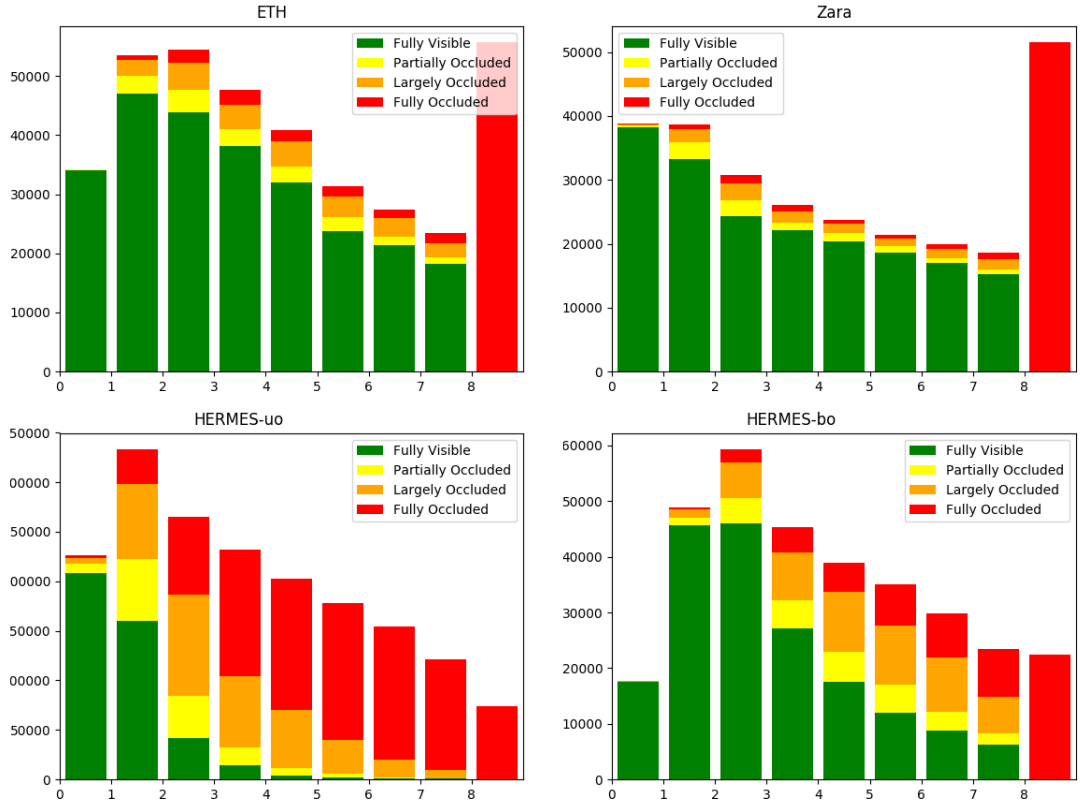


Figure 5.6 – Occlusion Severity: ETH / Zara / Hermes (Uni-directional and Bi-directional flows)

5.4.5 Analysis of Tie Patterns

In order to measure the amount of information captured by the tie patterns, we calculate the entropy of each pattern for each dataset. The equation of normalized entropy for the distribution is derived by considering different bin sizes in a polar histogram [Wal06]:

$$\bar{H}(p) = -\sum_{r,\theta} p^{r,\theta} \log\left(\frac{p^{r,\theta}}{A^{r,\theta}}\right) / H_{max} \tag{5.8}$$

where $p^{r,\theta}$ and $A^{r,\theta}$ are the normalized value and the area of the bin (r,θ). The total value is divided by $H_{max} = \log\left(A^{R,2\pi}\right)$, the maximum entropy of a uniform disk to obtain a normalized entropy between $[0,1]$. In Fig. 5.7, we see the entropy values for different datasets.

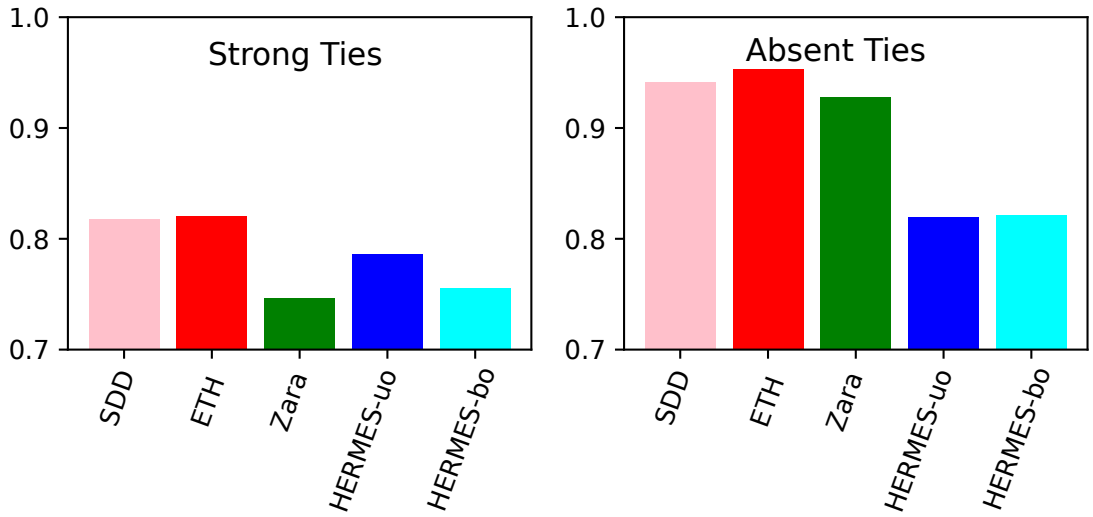


Figure 5.7 – Entropies of Strong/Absent Ties distributions for different datasets. Lower entropy means the distribution contains more structured patterns.

5.4.6 Baselines

We consider two baselines to compare with our algorithm:

1. **Vanilla-MOT**: A Multi-object tacking without handling the occluded agents, making no assumption about possible agents in occluded areas.
2. **PCF**: A comparable algorithm for analysis and synthesis of point distributions based on Point Correlation Functions (PCF) [ÖG12]. The algorithm uses a target PCF (or distribution), that is extracted by analyzing the historical data $\mathcal{H}$, and tries to reconstruct a given set of points (here: the partially occluded crowd) by iteratively sampling new points and accepting/rejecting them based on matching the source and the target PCF curves.

5.4.7 Performance Evaluation

In order to evaluate the results, we first perform Kernel Density Estimation from the ground truth distribution of agents location, by using Gaussian kernels centered at the location of each agent with $\sigma = 0.5m$. We denote the result (a probabilistic occcupancy map) by π. The Mean Squared Error (MSE) is calculated by averaging the squared difference of predicted and ground truth occupancy grid maps:

$$MSE = \frac{1}{A \times H} \sum_{h=1}^{H} \sum_{x,y} \left\| \pi - \hat{\pi}^{(h)} \right\|^2 \tag{5.9}$$

where H is the number of generated samples, and A is the area of the map. The map $\hat{\pi}^{(h)}$ is

built similarly to π, based on the sample $\mathbf{Z}^{+(h)}$.

We run the simulation, for each of the trajectory $\#i$ in the test set, and execute the crowd prediction algorithm to obtain $\hat{\pi}_i^{(h)}$ at each time-step. The prediction errors for Hermes and ETH datasets are shown in Fig. 5.8. As you can see the proposed algorithm has improved the Vanilla-MOT baseline on Hermes dataset and also outperforms the PCF baseline. On the other hand, on ETH dataset the algorithm has not improved the results which means it's better no make no prediction in scenarios with few-occlusions. Due to this issue we do not report the prediction results on other low-occlusion datasets (SDD and Zara).

In Fig. 5.9 some imputation examples are shown for Hermes dataset. The algorithm has proposed interesting samples in many cases using the social-tie patterns it has learned.

5.5 Conclusion and Future Work

We have proposed a new approach to impute the structure of a crowd in which a robot is performing navigation with its limited sensing. We leverage several new concepts that we have introduced to describe crowd patterns around the robot (strong and absent ties, communities and territories) to form a generative model for crowd patterns and use it to samples of imputed occupation maps, based on what is observed through the robot perception. We have shown on real crowd datasets that the proposed indicators reflect the nature of the typical pairwise relation within crowds, and we have obtained competitive prediction results, in particular on datasets with well-structured pedestrian flows.

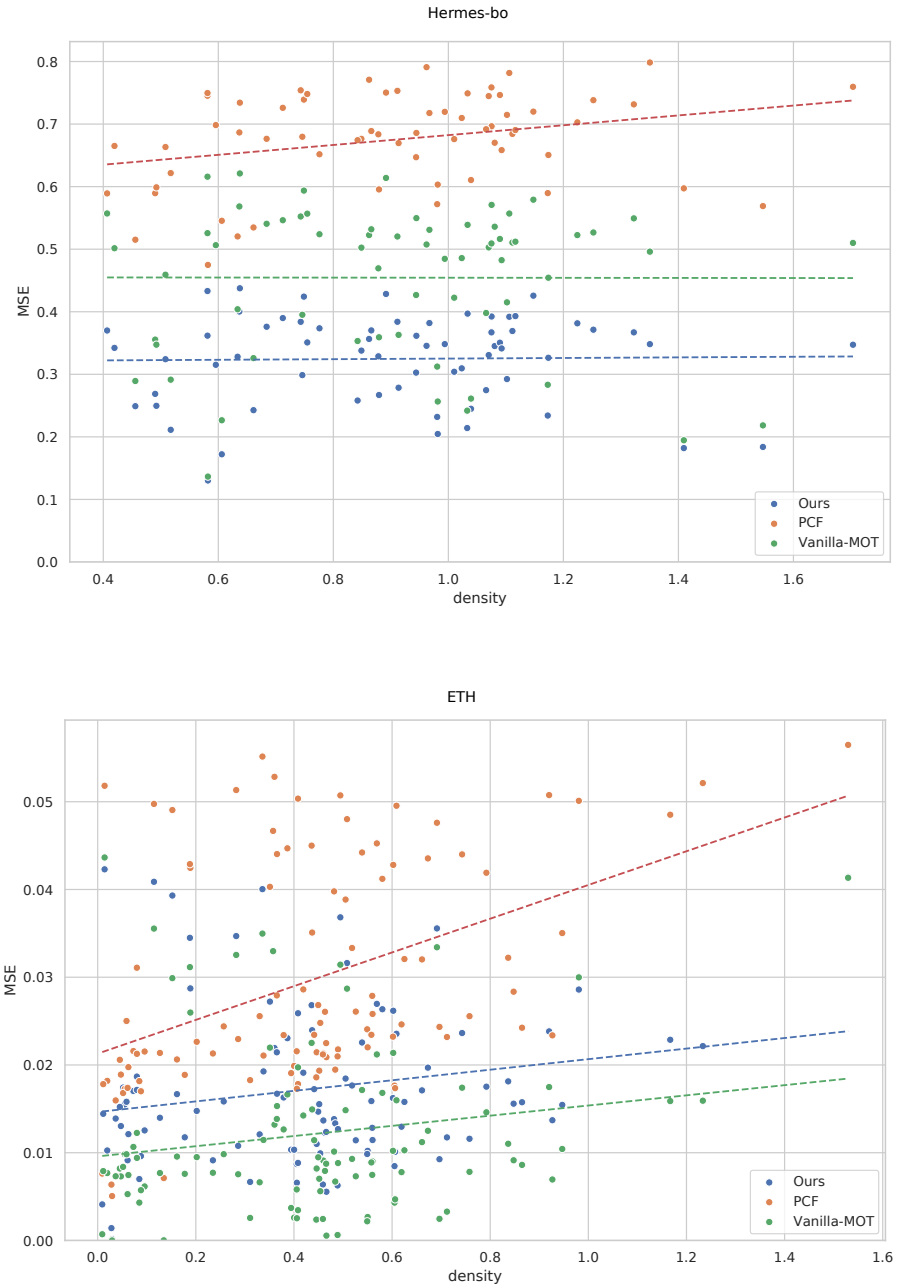
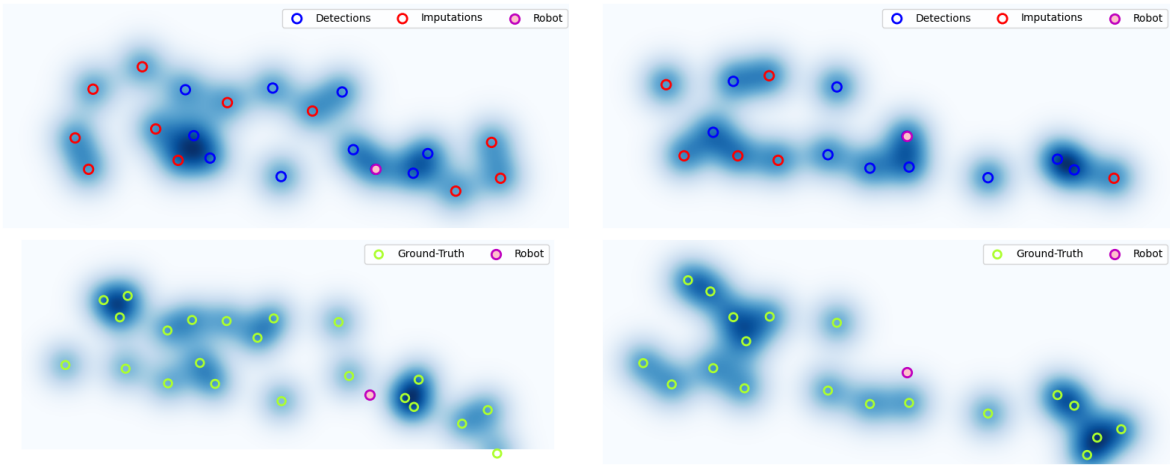
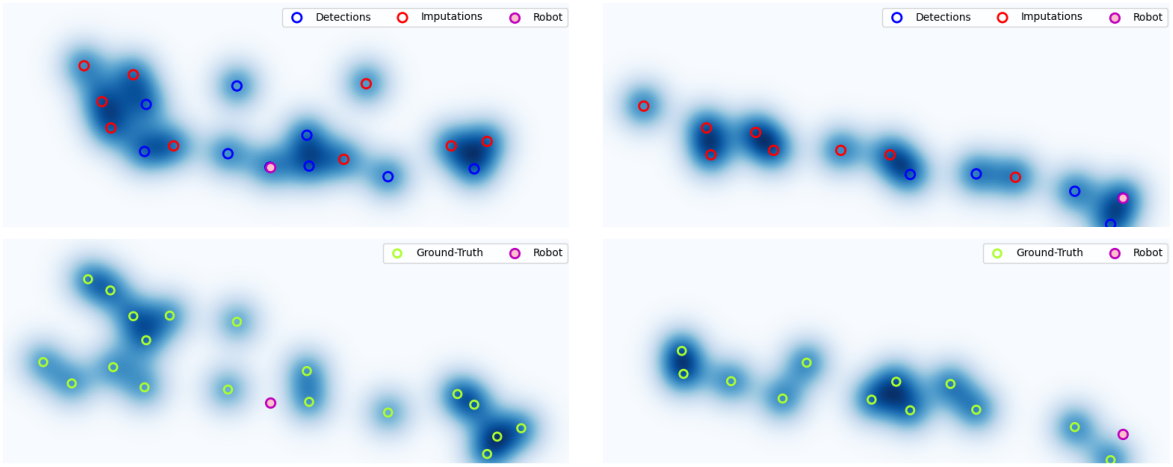


Figure 5.8 – Prediction (imputation) error (MSE) on Hermes and ETH datasets. Each point on the plot represents the average error of the predictions for one trajectory, sorted by the average crowd density around the robot.



(a) (b)



(c) (d)

Figure 5.9 – Qualitative results of crowd-imputation on the HERMES dataset. In four examples, we show the ground truth crowd (lower picture), in addition to the imputed crowd and the detections.

Chapter 6

Data-Driven Crowd Simulation with GANs

6.1 Introduction

As we discussed in the introduction of the thesis (Chapter 1), realistic simulation of crowd can be one of the applications of human trajectory prediction. Generally, the goal of a crowd simulation algorithm is to populate a virtual scene with a crowd that exhibits visually convincing behavior. The simulation should run in real time to be usable for interactive applications such as games, training software, and virtual-reality experiences.

Many simulations are *agent-based*: they model each pedestrian as a separate intelligent agent with individual properties and goals. To simulate complex behavior, *data-driven* crowd simulation methods use real-world input data (such as camera footage) to generate matching crowd motion. Usually, these methods cannot easily generate *new* behavior that is not literally part of the input. Also, they are often difficult to use for applications in which agents need to adjust their motion in real-time, e.g., because the user is part of the crowd.

In this chapter, we present a data-driven crowd simulation method. Our system enables the real-time simulation of agents that behave similarly to observations, while allowing them to deviate from their trajectories when needed. More specifically, our method:

1. learns the statistics of input trajectories, and can generate new trajectories from this probability distribution;
2. embeds these trajectories in a crowd simulation, in which agents follow a trajectory while allowing for local interactions.

For item 1, we use Generative Adversarial Networks (GANs) [GPAM+14], and for item 2, we extend the concept of 'route following' [JCIG13] to trajectories with a *temporal* aspect, prescribing a speed that may change over time. Using a real-world dataset as an example, we will show that our method generates new trajectories with matching styles. Our system can (for example) reproduce an existing scenario with additional agents, and it can easily be combined with other crowd simulation methods.

6.2 Related Work

Agent-based crowd simulation algorithms model pedestrians as individual intelligent agents. In this paradigm, many researchers focus on the *local* interactions between pedestrians (e.g. collision avoidance) using *microscopic* algorithms [HM95, vdBLM08]. In environments with obstacles, these need to be combined with *global* path planning into an overall framework [vTJG15]. A growing research topic lies in measuring the 'realism' of a simulation, by measuring the similarity between two fragments of (real or simulated) crowd motion [WOO16].

Complex real-life behavior can hardly be described with simple local rules. This motivates *data-driven* simulation methods, which base the crowd motion directly on real-world trajectories, typically obtained from video footage. One category of such methods stores the input trajectories in a database, and then pastes the best-matching fragments into the simulation at run-time [LCL07, LCHL07]. Another technique is to create pre-computed *patches* with periodically repeating crowd motion, which can be copied and pasted throughout an environment [YMPT09]. Such simulations are computationally cheap, but difficult to adapt to interactive situations.

Researchers have also used input trajectories to train the parameters of (microscopic) simulation models [WJGO+14], so as to adapt the agents' local behavior parameters to match the input data. However, this cannot capture any complex (social) rules that are not part of the used simulation model.

To replicate how agents move through an environment at a higher level, some algorithms subdivide the environment into cells and learn how pedestrians move between them [PGSVG12, ZCLZ16]. Our goal is similar (reproducing pedestrian motion at the full trajectory level), but our approach is different: we learn the spatial and temporal properties of complete trajectories, generate new trajectories with similar properties, and let agents flexibly follow these trajectories.

Our work uses Generative Adversarial Networks (GANs) [GPAM+14], a machine learning technique for generating new data. We showed in Chapter 4 how to adopt GANs for short-term prediction of pedestrian motion. To our knowledge, this is the first time that GANs are applied in crowd simulation at the full trajectory level.

6.3 Generating Trajectories

In this section, we describe our GAN-based method for generating trajectories that are similar to the examples in our training data.

As in most crowd-simulation research, we assume a planar environment and we model agents as disks. We define a *trajectory* as a mapping $\pi\colon [0, T] \to \mathbb{R}^2$ that describes how an agent moves through an environment during a time period of T seconds. Note that a trajectory encodes *speed*

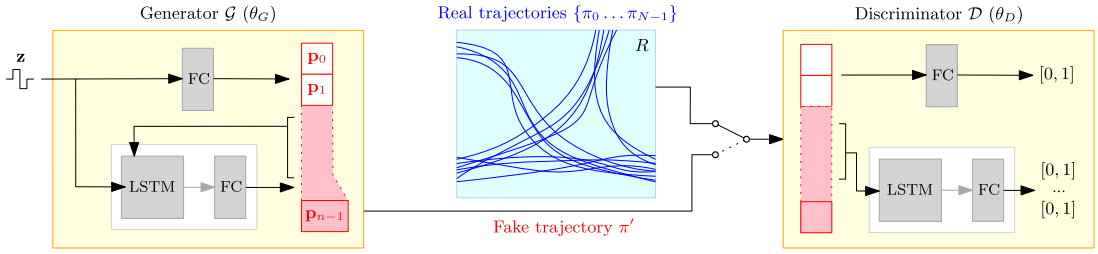


Figure 6.1 – Our GAN architecture for learning and generating pedestrian trajectories.

information: our system should capture when agents speed up, slow down, or stand still. In practice, we will represent a trajectory π by a sequence of n_π points $[\mathbf{p}_0^\pi, \dots, \mathbf{p}_{n_\pi-1}^\pi]$ separated by a fixed time interval Δt; that is, each $\mathbf{p}_i^\pi$ has a corresponding timestamp $i \cdot \Delta t$. In our experiments, we use $\Delta t = 0.4$s because our input data uses this value as well. We will use the notation $\mathbf{p}_{j:k}^\pi$ to denote a sub-trajectory from $\mathbf{p}_j^\pi$ to $\mathbf{p}_k^\pi$.

Given a dataset of trajectories $\Pi = \{\pi_0, \dots, \pi_{N-1}\}$, our generator should learn to produce new trajectories with properties similar to those in Π. We assume that all trajectories start and end on the boundary of a *region of interest* R, which can have any shape and can be different for each environment.

6.3.1 Overview of Our System

Figure 6.1 displays an overview of our GAN system, which consists of two components: a *generator* $\mathcal{G}$ that creates new samples and a *discriminator* $\mathcal{D}$ that judges whether a sample is real or generated. The generator and discriminator both have two tasks:

— generating or evaluating the *entry points* of a trajectory π (i.e. the first two points $\mathbf{p}_0^\pi$ and $\mathbf{p}_1^\pi$),

— and generating or evaluating the *continuation* of a trajectory (i.e. the next point $\mathbf{p}_{k+1}^\pi$ after a sub-trajectory $\mathbf{p}_{0:k}^\pi$).

For the continuation tasks, we use 'conditional GANs' because the generator and discriminator take extra data as input. We will now describe the system in more detail. Parameter settings will be mentioned in Section 6.5.

6.3.2 Generator

To generate *entry points*, the generator $\mathcal{G}$ feeds a random vector $\mathbf{z}$ to a fully connected (FC) block of neurons. Its output is a 4D vector that contains the coordinates of $\mathbf{p}_0^\pi$ and $\mathbf{p}_1^\pi$.

To generate the *continuation* of a trajectory $\mathbf{p}_{0:k}^\pi$, the generator $\mathcal{G}$ feeds $\mathbf{p}_{0:k}^\pi$ and a noise vector $\mathbf{z}$ to a LSTM layer that should encode the relevant trajectory dynamics.

The output of this LSTM block is sent to an FC block, which finally produces a 2D vector with the coordinates of $\mathbf{p}^{\pi}_{k+1}$. Let $g(\mathbf{z}|\mathbf{p}^{\pi}_{0:k};\theta_G)$ denote this result. Ideally, this point will be taken from the (unknown) distribution of likely follow-ups for $\mathbf{p}^{\pi}_{0:k}$.

The continuation step is repeated iteratively. If the newly generated point $\mathbf{p}^{\pi}_{k+1}$ lies outside of the region of interest R, then the trajectory is considered to be finished. Otherwise, the process is repeated with inputs $\mathbf{p}^{\pi}_{0:k+1}$ and a new noise vector.

6.3.3 Discriminator

The discriminator $\mathcal{D}$ takes an entire (real or fake) trajectory π as input. It splits the discrimination into two tasks with a similar structure as in $\mathcal{G}$. For the *entry point* part, an FC block evaluates $\mathbf{p}^{\pi}_{0:1}$ to a scalar in $[0,1]$, which we denote by $v_e(\mathbf{p}^{\pi}_{0:1};\theta_D)$. For the *continuation* part, an LSTM+FC block separately evaluates each point $\mathbf{p}^{\pi}_{k}$ (for $2 \leq k < n_\pi$) given the sub-trajectory $\mathbf{p}^{\pi}_{0:k-1}$. We denote the result for the kth point by $v_c(\mathbf{p}^{\pi}_{0:k};\theta_D)$.

So, for a full trajectory π of n_π points, the discriminator computes $n_\pi - 1$ scalars that together indicate the likelihood of π being real. The training phase uses these numbers in its loss function.

6.3.4 Training

Each training iteration lets $\mathcal{G}$ generate a set Π' of N trajectories for different (sequences of) noise vectors. We then let $\mathcal{D}$ classify all trajectories (both real and fake). The loss function of our GAN is the sum of two components:

— the log of classification accuracy of discriminating the entry points:

$$\sum_{\pi\in\Pi} \log v_e(\mathbf{p}^{\pi}_{0:1};\theta_D) + \sum_{\pi\in\Pi'} \log(1 - v_e(\mathbf{p}^{\pi}_{0:1};\theta_D)),$$

— the log of classification accuracy of discriminating all other points:

$$\sum_{\pi\in\Pi} \sum_{k=2}^{n_\pi-1} \log v_c(\mathbf{p}^{\pi}_{0:k};\theta_D) + \sum_{\pi\in\Pi'} \sum_{k=2}^{n_\pi-1} \log(1 - v_c(\mathbf{p}^{\pi}_{0:k};\theta_D)).$$

To let our GAN train faster, we add a third component. For each real trajectory $\pi \in \Pi$, we take all valid sub-trajectories $\mathbf{p}^{\pi}_{k:k+4}$ of length 5 and let $\mathcal{G}$ generate its own version of $\mathbf{p}^{\pi}_{k+4}$ given $\mathbf{p}^{\pi}_{k:k+3}$. We add to our loss function:

$$\sum_{\pi\in\Pi} \sum_{k=0}^{n_\pi-5} ||\mathbf{p}^{\pi}_{k+4} - g(\mathbf{z}|\mathbf{p}^{\pi}_{0:k+3};\theta_G)||$$

i.e. we sum up the Euclidean distances between real and generated points. We observed that this additional component leads to much faster convergence and better back-propagation.

To reduce the chance of 'mode collapse' (i.e. convergence to a limited range of samples), we use an 'unrolled' GAN [MPPSD17]. This is an extended GAN where each optimization step for θ_G uses an improved version of the discriminator that is u steps further ahead (where u is a parameter).

6.4 Crowd Simulation

Recall that our goal is to use our trajectories in a real-time interactive crowd simulation, where agents should be free to deviate from their trajectories if needed. This section describes how we combine our trajectory generator with a crowd simulator.

Our approach fits in the paradigm of multi-level crowd simulation [vTJG15], in which global planning (i.e. computing trajectories) is detached from the simulation loop. This loop consists of discrete timesteps. In each step, new agents might be added, and each agent tries to follow its trajectory while avoiding collisions.

6.4.1 Adding Agents

To determine when a new agent should be added to the simulation, we use an exponential distribution whose parameter λ denotes the average time between two insertions. This parameter can be obtained from an input dataset (to produce similar crowdedness), but one may also choose another value deliberately. Each added agent follows its own trajectory produced by our GAN.

6.4.2 Trajectory Following

In each frame of the simulation loop, each agent should try to proceed along its trajectory π while avoiding collisions. The main difference with classical 'route following' [JCIG13] is that our trajectories have a *temporal* component: they prescribe at what speed an agent should move, and this speed may change over time. Therefore, we present a way to let an agent flexibly follow π while respecting its spatial *and* temporal data. Our algorithm computes a *preferred velocity* v_{pref} that would send the agent farther along π. This v_{pref} can then be fed to any existing collision-avoidance algorithm, to compute a velocity that is close to v_{pref} while avoiding collisions with other agents.

Two parameters define how an agent follows π: the *time window* w and the *maximum speed* $s_{\max}$. An agent always tries to move to a point that lies w seconds ahead along π, taking $s_{\max}$ into account. During the simulation, let t be the time that has passed since the agent's insertion. Ideally, the agent should have reached $\pi(t)$ by now. Our algorithm consists of the following steps:

1. Compute the *attraction point* $\mathbf{p}_{\text{att}} = \pi(t_{\text{att}})$, where $t_{\text{att}} = \min(t + w, T)$ and T is the end time of π. Thus, $\mathbf{p}_{\text{att}}$ is the point that lies w seconds ahead of $\pi(t)$, clamped to the end of π if needed.
2. Compute the *preferred velocity* $\mathbf{v}_{\text{pref}}$ as $\frac{\mathbf{p}_{\text{att}} - \mathbf{p}}{t_{\text{att}} - t}$, where $\mathbf{p}$ is the agent's current position. Thus, $\mathbf{v}_{\text{pref}}$ is the velocity that will send the agent to $\mathbf{p}_{\text{att}}$, with a speed based on the difference between t and t_{att}.
3. If $||\mathbf{v}_{\text{pref}}|| > s_{\max}$, scale $\mathbf{v}_{\text{pref}}$ so that $||\mathbf{v}_{\text{pref}}|| = s_{\max}$. This prevents the agent from receiving a very high speed after it has been blocked for a long time.

6.4.3 Collision Avoidance.

The preferred velocity $\mathbf{v}_{\text{pref}}$ computed by our algorithm can be used as input for any collision-avoidance routine. In our implementation, we use the popular ORCA method [vdBLM08]. In preliminary tests, other methods such as social forces [HM95] proved to be less suitable for our purpose.

6.5 Experiments and Results

Set-up. We have implemented our GAN using the *PyTorch* library [1]. The input noise vectors are 3-dimensional and drawn from a uniformly random distribution. In both $\mathcal{G}$ and $\mathcal{D}$, the entry-point FC blocks consist of 3 layers with 128, 64, and 32 hidden neurons, respectively. For the continuation part, the LSTM blocks consist of 62 cells, and the FC blocks contain 2 layers of 64 and 32 hidden neurons. To save time and memory, the LSTM blocks only consider the last 4 samples of a sub-trajectory.

For training the GAN, all FC layers use *Leaky-ReLU* activation functions (with slope 0.1), to let the gradient always back-propagate, which avoids vanishing gradient issues. We train the GAN for 50,000 iterations, using an unrolling parameter $u = 10$.

In the crowd simulation, we model agents as disks with a radius of 0.25m, and we use a simulation loop with a fixed frame length of 0.1s. In each frame, all agents perform our route-following algorithm (with $w = 5$ and $s_{\max} = 2\text{m/s}$), followed by the ORCA algorithm [vdBLM08] as implemented by the original authors. We remove an agent when it reaches the end of its trajectory.

We test our method on the *ETH* dataset [PESVG09] that contains recorded trajectories around the entrance of a university building. We have defined the region of interest R as an axis-aligned bounding box, and we use only the 241 trajectories that both enter and exit R.

1. `https://pytorch.org/`

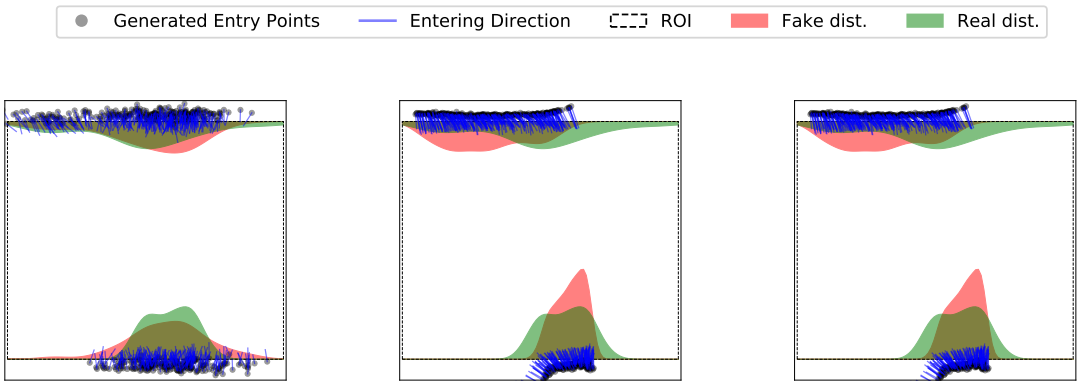


Figure 6.2 – The distribution of entry points created by three different methods.

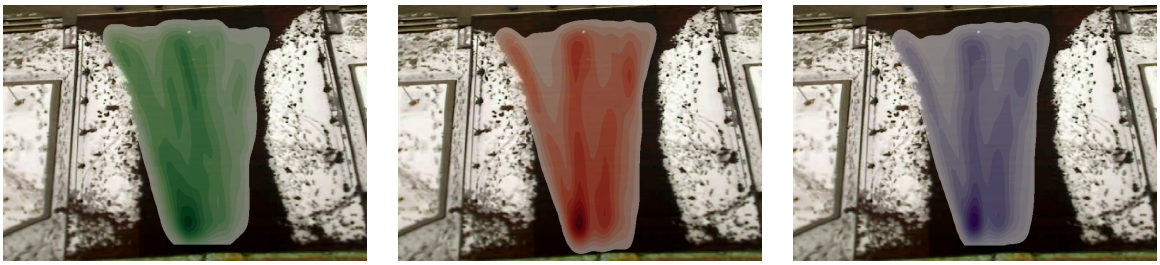


Figure 6.3 – Trajectory heatmaps: the input data, the generated trajectories, and the final simulated agent motion.

6.5.1 Result 1: Entry Points

To show the performance of our GAN in learning the distribution of entry points, we computed 500 (fake) entry points in the ETH scene, and we calculated the distribution of the samples over the boundary of R. We also compared these results against two other generative methods: a Gaussian Mixture Model (GMM) with 3 components, and a 'vanilla' GAN variant that does not use the unrolling mechanism. As shown in Fig. 6.2, the entry points of the unrolled GAN (right) are closer to the real data than those of the other two methods.

6.5.2 Result 2: Trajectories

Next, we used our system to generate 352 new trajectories, and we used them to simulate a crowd. The first two heatmaps in Fig. 6.3 show that generated trajectories (middle) are similarly distributed over the environment as the real data (left).

The third heatmap shows the final motion of the simulated agents with route following and

collision avoidance. In this scenario, agents are well capable of following their given trajectories.

6.5.3 Computation time

We used CUDA to run our GAN on a NVIDIA Quadro M1200 GPU with 4GB of GDDR5 memory. With this set-up, generating a batch of 1024 trajectories (with a maximum length of 40 points) took 152ms, meaning that the average generation time was 0.15ms per trajectory. Thus, after training, the system is sufficiently fast for real-time insertions of trajectories into a crowd.

6.6 Conclusions & Future Work

In this chapter, we presented a data-driven crowd simulation method that uses GANs to learn the properties of input trajectories and then generate new trajectories with similar properties. Combined with flexible route following that takes temporal information into account, the trajectories can be used in a real-time crowd simulation. Our system can be used, for example, to create variants of a scenario with different densities. It can easily be combined with other simulation methods, and it allows interactive applications.

In the future, we will perform a thorough analysis of the trajectories produced by our system, and compare them to other algorithms. We will also investigate the exact requirements for reliable training. Furthermore, our system generates trajectories for individuals, assuming that agents do not influence each other's choices. As such, it cannot yet model group behavior, and it performs worse in high-density scenarios where agents cannot act independently. We would like to handle these limitations in future work.

Chapter 7

Crowd-Robot Interaction: Understanding the Effect of Robots on Crowd Motion

7.1 Introduction

This chapter presents a collaborative research between Inria and University College London (UCL) on "whether and how the presence of a robot affects pedestrians and crowd dynamics and whether this influence varies across different robots?" To answer this question, a crowd-robot gate-crossing experiment was conducted at the PAMELA facilities in London in August 2019. The study involved 28 participants and two distinct robot representatives: A smart wheelchair and a Pepper humanoid robot. Collected data includes video recordings, robot and participant trajectories, and participants' responses to post-interaction questionnaires.

A quantitative analysis was performed on the trajectories of the robot and participants. This analysis suggests that the robot affects crowd dynamics in terms of trajectory regularity and interaction complexity. Also, qualitative results indicate that pedestrians tend to be more conservative and follow "social rules" while passing a wheelchair compared to a humanoid robot. These insights may be useful for the design of social navigation strategies that encourage natural interaction by taking account of the robot's effects on the crowd dynamics.

7.2 Contributions

We conducted a crowd-robot gate-crossing experiment with the two aforementioned robots and measured the effect each robot exerted on the crowds macroscopically (i.e. as one moving body of people) and on groups of individuals microscopically (i.e. pedestrians in close proximity and far away from the robot). This study makes the following contributions:

1. The first controlled crowd-robot experiment with recorded pedestrian trajectory dataset in the presence of a robot, which presents novel results.

2. An understanding of how pedestrian and crowd dynamics is affected by a robot. More specifically, we use both local and global metrics to explore further the effect of the robot motion on participants at the closest proximity of the robot. It will inform the design of a more natural crowd prediction method and a more realistic crowd simulation scenario.
3. An understanding of how the type of robot affects pedestrian behavior, which highlights considerations for designing robotic motion planning algorithms that also take into account the effect of the robot on the crowd.

Specifically, my contribution to this research was two-fold::

1. Collaboration on data collection, including the setup of recording equipment, estimation of camera calibration parameters, visual tracking of participants, and post-processing of the tracks.
2. Collaboration in the design of the quantitative metrics and in the analysis of the results of experiments.

Also the contributions of *Bingqing Zhang*, the main contributor to this experiment affiliated with the UCL, was two-fold:

1. Design and set up of the experiment
2. Analyzing and evaluation of the quantitative data (tracks) as well as qualitative data (questionnaire)

In the following sections, we discuss the detail of the research.

7.3 Related Work

The use of robots within pedestrian spaces is becoming increasingly common. Mobile robots with various shapes, sizes and functions have been applied in areas such as logistics, transportation and healthcare. For example, humanoid robots such as Pepper have been used to assist customers in train stations and restaurants, autonomous vehicles have been increasingly observed on the road and smart wheelchairs have been developed and tested in clinical trials.

In many of these environments, the robot must interact with pedestrians in a safe and potentially social way, requiring an understanding of pedestrian dynamics in response to different robots. However, state-of-the-art approaches normally model pedestrian dynamics using simulation or data collected in human-only experiments [HM95, HD05]. Few works have explored pedestrian dynamics in a robot-populated environment and they either studied it with a specific type of robot or limited number of pedestrians [CJG18, MHM+19, VOS+17]. Although human perception and interaction with different types of robot have been studied in many areas, it remains to be explored in a crowd-robot navigation scenario.

7.3.1 Pedestrian Robot Interaction

The study of Human-robot interaction has been an emerging area over the past few years. People's perception and reaction towards a robot is different from that to another human, and is greatly affected by factors such as demographics [MHB+17], appearance and size of the robot [BA01], perceived likeability and aggressiveness of the robot [MM11], and personal experience with pets or robots [TP09]. Despite this wide range of study, human-robot interaction in navigation is still relatively unexplored.

Recently, a small number of works have been focused on pedestrian-robot interaction and its effect on pedestrian dynamics. Vassallo et al. conducted a study which investigated a gate crossing scenario between one pedestrian and one robot: 10 participants and 279 trials were performed in total [VOS+18], with experimental results indicating that there is no difference in terms of the crossing order between human-human and the human-robot case, while the experiment in [VOS+17] reported that pedestrians prefer to give way to a passive robot. Marvrogiannis et al. conducted a within-subjects study to investigate the effect of distinct robot navigation algorithms on pedestrians' behavior [MHM+19]. Among 105 participants, three of them interacted with a lab-built robot in each trial. Unlike these studies where the pedestrian dynamics is analysed on individuals or a small groups of people, Chen et al. studied crowd-robot interaction by performing a corridor exiting experiment with 11 participants and 1 robot [CJG18]. Six test cases with different robot or pedestrian speeds have been explored. The result indicated that pedestrians' overall speed was affected by the presence of the robot.

Although these experiments demonstrated valuable results in pedestrian-robot interaction, most of them focused on an individual or small groups of pedestrians, which limits the applicability to the crowd-robot navigation scenario. In addition, these works concerned with one specific type of lab-made robot, which restricts the result from being generalized to other robots. This leads to a question: Does the existence and the type of the robot affect crowd dynamics, and what would this effect be?

7.3.2 Crowd Prediction and Robot Navigation

A human is capable of navigating through crowds by predicting the motion trajectories of surrounding pedestrians and taking them into account when planning his or her own movement. To achieve safe and human-like navigation for robots, it is crucial to mimic this decision-making process by taking pedestrians' trajectories into consideration. Early efforts in this area include the 'social force model' [HM95] which used 'attractive' and 'repulsive' force to model pedestrian-obstacles and pedestrian-destination interactions. Several extensions are proposed to this model [YKS14, KHvBO09]. Yamaguchi et al. (2011) proposed to take the grouping behavior, smoothness of movements and preferred speed of the pedestrian into account [YBOB11]. The main

concern with these models is that the hand-crafted rules may not perfectly reflect the realistic behaviors of humans.

Data-driven approaches are then proposed to resolve this problem. They allow the natural human-human interaction to be captured and learned directly using real-world data. Machine learning and deep learning methods such as Long Short-Term Memory (LSTM) [AGR+16] have been applied to predict individual trajectories with the pairwise pedestrian interactions being learned via a social-pooling layer. Generative Adversarial Networks [GJFF+18, KSMM+19], Transformer models [GHCG20], etc., have also been proposed for this task. However, the majority of these models are trained and validated on human trajectory datasets that either only contain pedestrian trajectories (e.g. ETH dataset [PESVG09], UCY dataset [LCL07], Grand Central dataset [ZWT12]), or contain other non-robot road-users (e.g. Stanford drone dataset [RSAS16] that also contains trajectories of bikers, skateboarders, cars, buses, and golf carts, recorded in a University campus). Moreover, as reported in [AZC+20], many of these datasets only cover low-to-medium-density crowd activities. A model trained on such data might fail to make correct predictions in new situations.

State-of-the-art work also presents approaches that incorporate pedestrian trajectory prediction into robot navigation. Kerfs (2017) adopted an improved version of social-LSTM to predict trajectories distributions and then used a dynamic A* for robot route planning [Ker17]. Similarly, Pradhan et al. (2011) predicted pedestrians' position and used a potential function based path planner for robot navigation in crowds [PBB11]. These approaches achieved robot navigation in human populated environments considering the human-human interaction and how human behavior would affect the robot's decision, but ignored the potential influence that the robot would exert on the pedestrians.

In order to consider this mutual effect, some works combined the prediction and planning process together. Trautman et al. (2013) addressed this mutual interaction by reasoning the robot and pedestrians' future trajectories jointly [TMMK13]. Their solution was evaluated on the ETH pedestrian dataset [PESVG09]. Similarly, Kuderer et al. (2012) also treated the navigation problem as jointly planning for robot and pedestrians. Differently from [TMMK13], they learned natural pedestrian behaviours features from their own lab-collected pedestrian data using the idea of maximum entropy [KKSB12]. However, the data was recorded without the presence of a robot, and it has been pointed out by the authors that pedestrians may react differently to robots than to other humans.

While the advances in trajectory prediction and robot navigation are often of great significance, they inevitably leave a gap in the validation of the suitability of pedestrian trajectory dataset in robot navigation which requires consideration of the potential effect created by the robot. This gap boils down to understanding the interaction between the crowd and the robot,

which we aim to address in this study.

7.4 Proposed Method

In this work, we conducted a preliminary study on crowd-robot interaction focusing on how two specific type of "robot" (Pepper and a shared-control wheelchair) affect such interaction. This research aims to explore three main experimental questions:

1. How does the presence of a Pepper/wheelchair influence crowd motion?
2. Does the presence of a Pepper/wheelchair influence crowd-dynamics only at a local level, or is the effect global?
3. If the robot does influence crowd behavior how does the response vary between Pepper and the wheelchair?

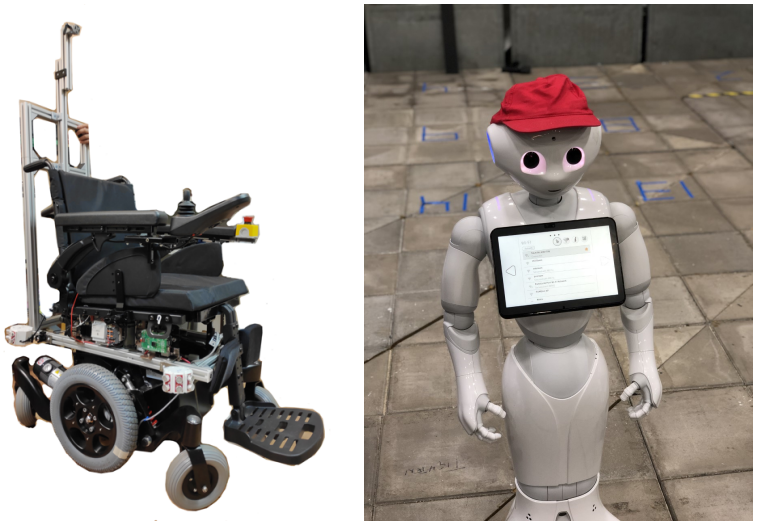
7.4.1 Gate-Crossing Experiment

In order to answer these questions, we conducted a crowd-robot gate-crossing experiment in an indoor environment. The experiment took place in an indoor pedestrian accessibility lab. The lab environment allows us to have significant control over multiple variables. The place consists of a platform which is constructed by 6×10 movable modules, with the size of each module being $1.2m \times 1.2m$. We used $6m \times 12m$ of the platform and constructed our gate using movable panels.

7.4.2 Choice of Robots

While a number of related works investigated pedestrian-robot interaction using a lab-made fully autonomous robot platform [CJG18, MHM+19], we decided to use one commercial humanoid robot (Pepper) and one shared-control wheelchair as shown in Fig. 7.1. A shared-control wheelchair is built on a standard powered wheelchair and has a collection of sensors for perception and navigation purpose. A user can express his or her driving intention through an interface (eg. Joystick), and the wheelchair's movement will be the result of a negotiation between the user input and the motion planner. It provides people who have mobility impairment and are considered unsafe to drive a traditional wheelchair with a mobility solution.

The reasons for choosing these robots are two-fold. Firstly, there is practical value to investigate this problem for robots that have been/would potentially be used in the public space. Secondly, these two types of robot offer us a great range of differences in terms of appearance, size, and dynamic constraints. The most distinct factor is the presence of a human 'driver' on the shared-control wheelchair. State-of-the-art works that explore human-aware navigation are



(a) Smart-wheelchair (b) Pepper

Figure 7.1 – Our smart wheelchair and the humanoid robot Pepper

mainly designed for fully autonomous robots, while such navigation strategy remains to be explored for semi-autonomous robot such as the shared-control wheelchair where a human driver can be seen by the pedestrians. It is suggested by Bingqing et al. that in contrast to a standalone fully-autonomous robot, an additional interaction channel between the wheelchair user and the surrounding pedestrians should be considered for a shared-control wheelchair [ZHCH19]. The fact that a human driver can be seen would potentially change pedestrians' perceptions and behaviors compared to the one with a standalone humanoid robot. Consequently, we include the shared-control wheelchair and Pepper in our study, and took the first step to collect such interaction data, with the aim to explore the impact that the system exert on the crowds and understand how that may differ across robots.

It should be noted that in this study, we do not consider the potential influence on pedestrians' dynamics caused by the different navigation algorithms. In addition, we assume that the robots are equipped with a human-like navigation strategy. As a result, a Wizard-of-Oz [Rie12] method was adopted in our experiment – the wheelchair was driven by an expert driver and the humanoid robot was tele-operated by an experienced operator.

Some may argue that the wheelchair is not actually 'shared-controlled' in this case and cannot be considered as a robot. Indeed, with no navigation algorithm, we may get similar results as those from the situation where a user is driving a standard powered wheelchair. The reason we called it 'shared-control wheelchair' is to emphasis that such system differs from other types of mobile robot by involving a human driver who can be seen by the pedestrians in daily use cases, and also differs from fully-autonomous wheelchair where the users are seen as a passenger from

the pedestrians' view. Most importantly, this is used to emphasis that such system would have a navigation strategy (and thus can be considered as a robot) and this study is used to inform the design of its navigation strategy. For simplicity, we will call it wheelchair in the rest of the chapter.

7.4.3 Participants

28 participants (15 females, 13 males) from different age groups (M=33, SD=8.8 years old) were recruited from UCL university and a participant pool. None had a mobility impairment. All of them had normal sight and hearing. The participants were given a copy of the information sheet and time to sign the consent form prior to the experiment. To prevent any potential bias, the actual purpose of the experiment was not revealed to them during the introduction.

7.4.4 Task

In contrast with previous work which studied interaction between one robot and one pedestrian [VOS+17] or small groups of people [MHM+19], we designed a robot-in-crowd gate-crossing task. The gate-crossing task has been widely used in analyzing crowd dynamics, in situations such as entering train stations and evacuation [TDLH+12]. During the experiment, each participant was asked to wear a colored hat for detection and tracking purposes (see Fig. 7.2).

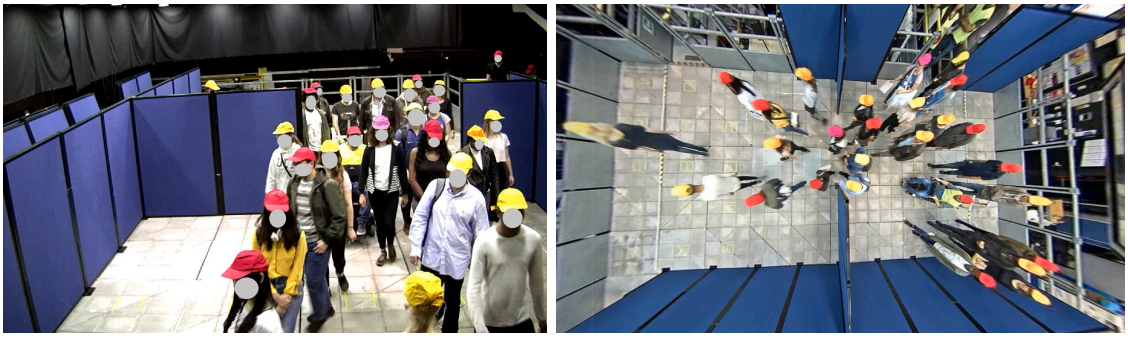


Figure 7.2 – Overview of the experiment from side and top camera

In each run, 28 people and the robot were randomly given an initial starting position number which represents one cube on the platform. In addition, we made sure that the starting positions assigned to the people in proximity to the "robot" were not the same in each run, so as to reduce the learning effect and potential bias caused by individual behavior. Furthermore, for more valid comparison, we let the participants keep their starting position across difference scenarios. For example, the starting position in S2 run 1 is different from S2 run 2 but is the same as S3 run 1. All pedestrians were instructed to walk together from one side of the platform to the other

side by crossing through a $2.2m$ wide gate (See Fig. 7.3). A vocal command was used to inform pedestrians of the start of each trial and the completion was achieved when all the pedestrians crossed a destination line at the end of the platform.

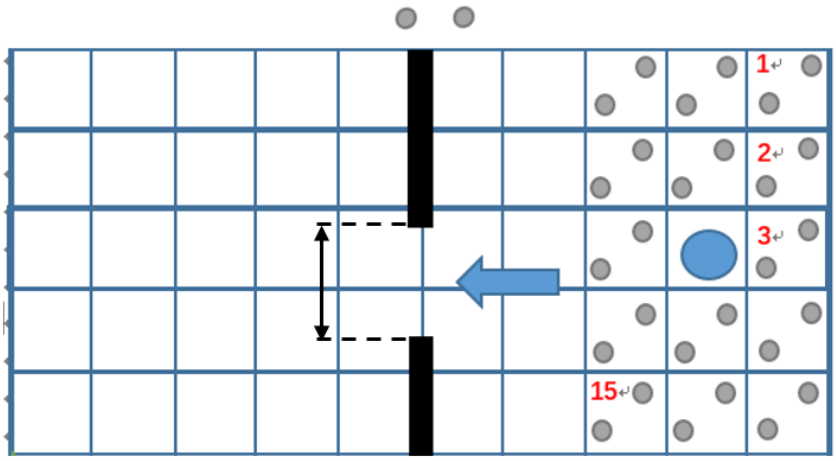


Figure 7.3 – Overview of the experimental plan

7.4.5 Experimental scenarios

We designed 4 testing scenarios with 3 independent variables, robot occurrence, robot type and robot speed (see Table 7.1). Due to the inherent differences in speed limit, we set the low speed around $0.5m/s$ for the wheelchair and Pepper, while the high speed for the wheelchair is about $1m/s$, which is comparable to normal pedestrian walking speed. Each scenario was repeated 5 times.

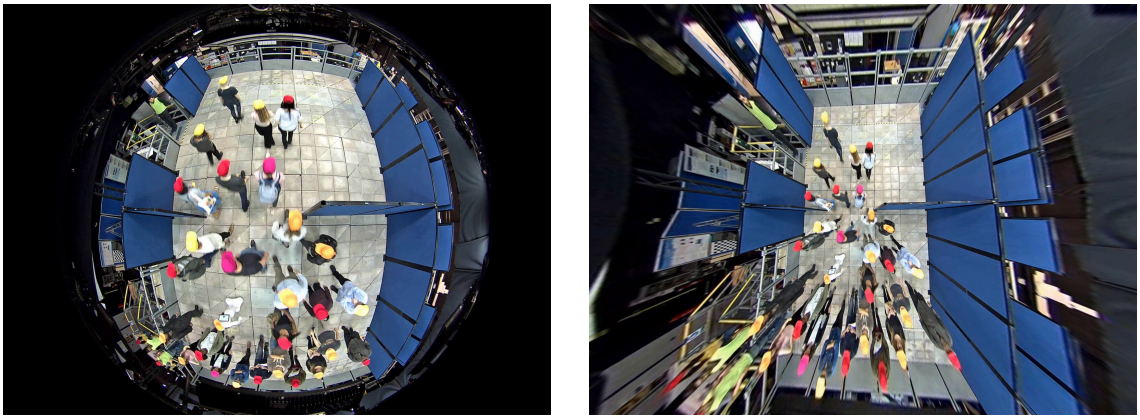
Table 7.1 – Five Experimental Scenarios

Experimental Scenarios		
Scene	Robot	Max Speed
S1	No robot	N/A
S2	Wheelchair	Low
S3	Wheelchair	High
S4	Pepper	Low

7.4.6 Data Collection

The experiment was recorded by an overhead fish-eye IP video camera (Axis M3037) at 12.5 frames per second. An additional IP video camera was set up from the side with the

aim to observe pedestrian behaviors qualitatively. In order to calibrate the camera, we used the chessboard method. We recorded a video with the camera, while a person was holding the chessboard toward the camera, moving around and rotating it. We extracted 22 different frames from this video to cover different positions and rotations of the chessboard in the image. We used Matlab Camera Calibrator toolbox [SMS06b] for estimating the calibration parameters of the fish-eye camera. It allowed us to undistort the videos (see Fig. 7.4).



(a) distorted frame (b) undistorted frame

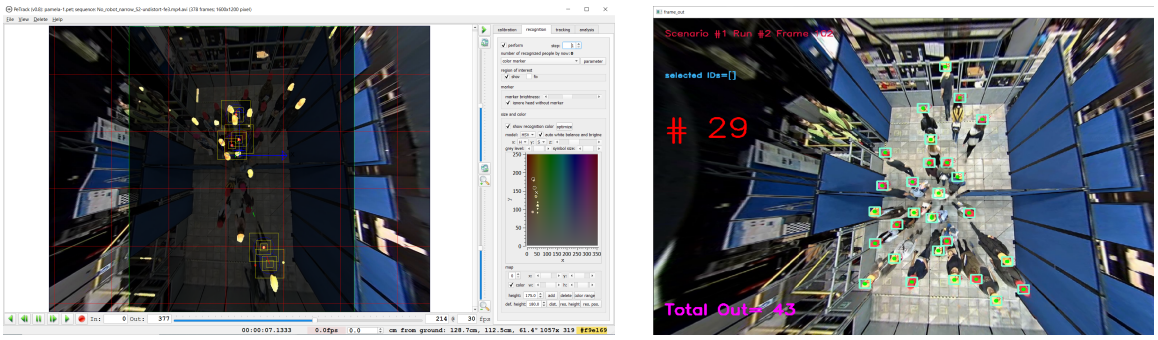
Figure 7.4 – Estimating camera calibration parameters, and un-distorting the recorded videos

We then used a special-purpose software called PeTrack to extract the pedestrian trajectories. (See Fig. 7.4.6 (a)) With PeTrack we were able to track the colorful helmets of pedestrians and the robot (or wheel-chair user). We then fixed the problems like ID switches and drifts in the detections by semi-automatic processes. Finally, we used a linear projection to map the head of the pedestrian to a point on the ground, which represents the leg of the person. For the humans we assumed an average height of 170 cm. For Pepper and the wheelchair we used heights of 120cm and 140 cm, respectively (See Fig. 7.4.6 (b)).

In order to filter out the high frequency jerks from the trajectories we applied a Kalman Filter (KF) with Rauch–Tung–Striebel (RTS) smoothing backward-pass [RTS65]. In total, 20 trials (for four scenarios, five runs each scenario) were performed with valid interactions between the robot and the crowds. 546 trajectories of pedestrians and the robots were extracted.

7.5 Analysis

To better evaluate the effect of presence of the robot in crowd dynamics in both global and local levels, we measured macroscopic and microscopic features, and reported the result quantitatively. Macroscopic features describe high-level crowd characteristics while microscopic features take individuals' properties into account. In order to quantify the effect of the presence



(a) PeTrack Software (b) Refining tracks by user

Figure 7.5 – Tracking pedestrians using PeTrack

of a robot and its effect on pedestrian dynamics, we analyzed the extracted trajectories both macroscopically and microscopically based on some common metrics have been used in previous works [MHM+19, VOS+17, PCBS11]. In general, we categorized the applied metrics to measure 'trajectory regularity' and 'interaction complexity'.

7.5.1 Preprocessing

Before applying metrics to the trajectories, we defined the notion of region-of-interests (ROI) as 2m before the gate and 0.5m after the gate. We observed that this was the region where most interaction happens. In addition, for more valid comparisons, we splitted all the trajectories extracted from the ROI into sub-trajectories with each having a fixed time length of 10 frames (=0.8s).

7.5.2 Trajectory Regularity

We evaluated the geometrical and physical properties of the sub-trajectories in order to reflect their irregularities and deviations from simple linear trajectory models. For this purpose, we used three metrics: average speed, average acceleration and path efficiency. Path efficiency is normally calculated as the ratio of distance between two terminals ($\vec{x}_{end}$ and $\vec{x}_{start}$) of the trajectory segment over the actual length of the segment [MHM+19]. However, in our experiment, the existence of the gate inherently affects pedestrians' path efficiency. This issue is addressed by dividing the sub-trajectories which cross the gate into 'before gate' and 'after gate', with the sub-goal ($\vec{x}_{sub}$) being introduced as the point on the sub-trajectory at the gate. As a result, path efficiency η for a sub-trajectory $\mathbf{X}^k$ is defined as:

$$\eta(\mathbf{X}^k) = \frac{\left\|\mathbf{x}^k_{sub} - \mathbf{x}^k_{start}\right\| + \left\|\mathbf{x}^k_{end} - \mathbf{x}^k_{sub}\right\|}{\sum_t \left\|\mathbf{x}^k_{t+1} - \mathbf{x}^k_t\right\|} \tag{7.1}$$

where t ranges from start to end.

7.5.3 Interaction Complexity

While the above metrics indicate the trajectory regularity and motion complexity of each pedestrian, they do not imply the interaction between pedestrian-pedestrian and pedestrian-robot. Consequently, we applied another three metrics to evaluate the interaction complexity in each scene. They are evacuation time, local density, and pass order inversion.
Evacuation time has been used to assess pedestrian dynamics in emergency situations. Here it is defined as the time elapsed from when the first pedestrian passes the gate to the time when the last pedestrian passes the gate. This quantity is further normalized by the number of pedestrians. In terms of local density, Helbing et.al proposed a formula based on the idea that each person occupies a fixed radius of area [HJAA07]. In this work, we adopted the notion proposed by Plaue et al. where a nearest neighbor Gaussian kernel estimator is used, which allows the difference of each pedestrian occupied area being taken into account [PCBS11]. For a point x_t, the local density $p(x_t)$ is defined as,

$$p(\mathbf{x}_t) = \frac{1}{2\pi} \sum_{i=1}^{K_t} \frac{1}{(\lambda d_t^i)^2} \exp\left(-\frac{\left\| \mathbf{x}_t^i - \mathbf{x}_t \right\|^2}{2(\lambda d_t^i)^2} \right) \tag{7.2}$$

where $d_t^i = min_{j \neq i} \|\mathbf{x}_t^i - \mathbf{x}_t\|$ is the Euclidean distance from agent i to its nearest neighbor and $\lambda > 0$ is a smoothing parameter.

In the daily life, a human always adapts to others when there is a risk of collision. To analyze whether such adaption exists in the human-robot navigation scenario and how it differs across different types of robot, we used a signed definition of the minimum predicted distance (SMPD) to analyze the adaption behavior [VOS+17]. As detailed in [OMCP12], minimal predicted distance (MPD) estimates the risk of future collision by calculating the distance to the closest approach (DCA) between the robot and the pedestrian at each time step, assuming they keep a constant velocity.

$$\begin{aligned} \hat{\mathbf{x}}(t,u) &= \mathbf{x}(t) + (u-t)\mathbf{v}(t) \\ MPD(t) &= \arg\min_u \|\hat{\mathbf{x}}_h(t,u) - \hat{\mathbf{x}}_r(t,u)\| \end{aligned} \tag{7.3}$$

where u is a future time parameter, and $\hat{\mathbf{x}}_h(t,u)$ and $\hat{\mathbf{x}}_r(t,u)$ are future positions of the human and robot.

By adding a sign to this metric, we can estimate whether the robot or the pedestrian is predicted to be ahead. In our study, we define t_{enter} as when the robot entered the ROI and t_{pass} as when either the pedestrian or the robot passed the gate. Consequently, we computed

SMPD(t_{enter}) and SMPD(t_{pass}) for each pedestrian-robot pair.

Considering the human perception capability, we only considered pedestrians who are behind the robot at t_{enter}. We define positive SMPD if the robot should pass first and negative SMPD otherwise. As a result, a change of sign of SMPD means that the future crossing order between the robot and the participant is switched, and thus implies the adaption in the pass order. In general, we define four pass order groups based on the sign of SMPD at t_{enter} and t_{pass}, namely: PosPos, NegNeg, PosNeg and NegPos. We classify 'PosPos' and 'NegNeg' as pedestrians who keep their pass order, 'PosNeg' represents pedestrians who overtake the robot while 'NegPos' implies pedestrians give the way to the robot.

7.6 Statistics

In order to assess the effect generated by the robot and whether it varies with robot type, detailed comparisons were made within and across scenarios. To guarantee valid data comparison, normality was assessed with the Kolmogorov-Smirnov test. It was indicated by the test result that statistics have a non-Gaussian distribution for some metrics. As a result, we used Wilcoxon ranked sum tests to determine differences and significant level. All effects were reported at $p < 0.05$. All the figures indicate the significant level with '*', where '*' stands for $p < 0.05$.

7.7 Experimental Results

7.7.1 Average Speed and Average Acceleration

Overall, our results indicate a significant difference between the human-only case (S1) and human-robot case (S2, S3, S4) in terms of average pedestrian speed and acceleration.

In order to further investigate the local effects, we grouped all pedestrian trajectories into two categories based on their spatial relationship to the robot. According to Hall's personal space model [Hal09], we evaluated each pedestrian's distance to the robot at each time stamp. For a pedestrian sub-trajectory, only those with the median of Euclidean distance less than the robot's close social space ($< 2.1m$) were considered as in proximity with the robot. This gives us 819 and 1067 pedestrian sub-trajectory segments near and far away from the robot.

Fig. 7.6a depicts the average pedestrian speed categorized by its proximity to the robot. The average value obtained in S1 is used as a baseline for comparison. It can be observed that in most scenarios (except S2_farSpeed), the average speed for pedestrians near and far away from the robot are significantly different from the baseline (S1, no robot case). Additionally, within the same scenario, pedestrians' average speed differs greatly based on their proximity to

the wheelchair (S2, S3) or Pepper (S4). In terms of average pedestrian acceleration, a similar difference exist between no robot and robot cases, while the local effect is less obvious.

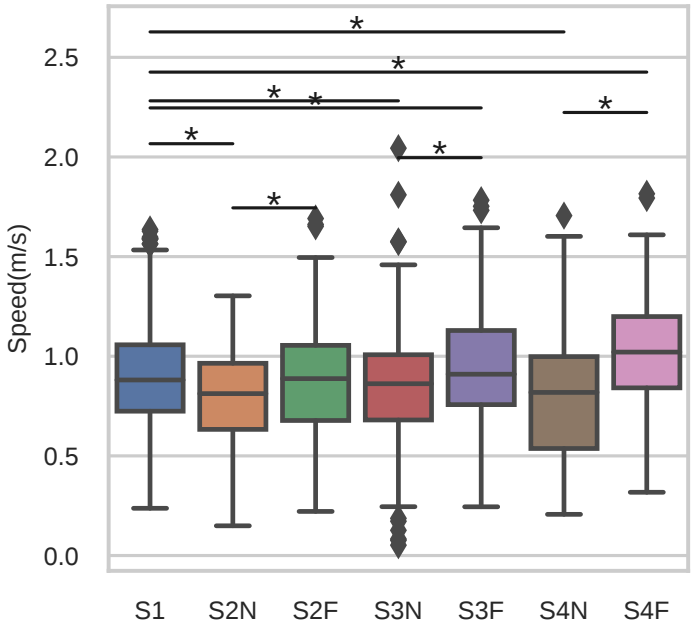


Figure 7.6 – Average pedestrian speed categorized by its proximity to the robot. 'N' and 'F' stand for the speed for pedestrians near the robot and far away from the robot. Pedestrian speeds in robot scenarios are significantly different from those in the no-robot scenario (S1). Within the same scenario, pedestrians that are in close proximity with the robot have lower speed compared to pedestrians far away from the robot.

7.7.2 Path Efficiency

In general, high path efficiency ($> 85\%$) in all scenarios was observed. While comparing the path efficiency of sub-trajectories categorized by their proximity to the robot within the same scenario, pedestrians who are in close proximity with the robot in S3 and S4 have slightly lower path efficiency compared with those who are distinct from the robot.

7.7.3 Evacuation time

Fig. 7.7 shows the evacuation time per person for all four scenes. By comparing S1 (no robot case) with S2-S4 (robot case), it can be observed that evacuation time significantly increased ($p < 0.05$) when a wheelchair or Pepper is involved. In addition, a significant difference is observed between the wheelchair case and the Pepper case regardless of the robot speed.

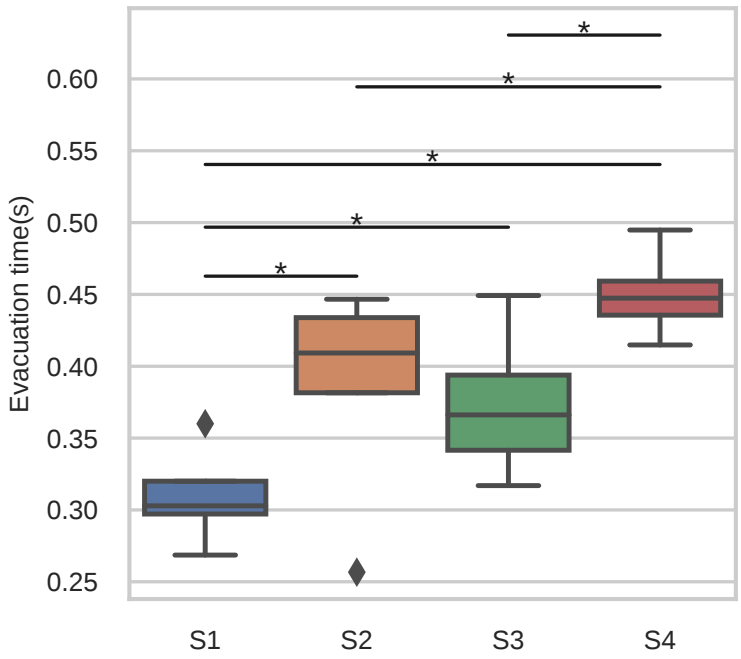


Figure 7.7 – Evacuation time per pedestrian. A significant increase can be observed when a robot is added to the crowd. Pedestrians need longer evacuation time when walking with Pepper compared to when walking with a smart wheelchair.

7.7.4 Local density

Average local density for each pedestrian or the robot sub-trajectory is illustrated in Fig. 7.8. Significant difference can be observed between no robot (S1) and robot (S2,S3,S4) cases, as well as between the wheelchair (S2,S3) and Pepper (S4). When the wheelchair was involved, the wheelchair speed also showed influence on average pedestrian the local density. In terms of the robot, local density around Pepper is significantly higher than the one around the wheelchair.

7.7.5 Pass Order

Fig. 7.9 provides a summary plot for the human-robot pass order in all scenarios. We can observe that about 50% of pedestrians kept their pass order when they walk with a wheelchair or Pepper. Among the pedestrians who adapt their behaviour, less than 20% of them overtake the wheelchair while this number is over 80% in the case of Pepper. On the contrary, less than 20% of pedestrians give way to the Pepper while over 80% of them let the wheelchair pass first regardless of the wheelchair speed setting. This behaviour difference is visualized in Fig. 7.10.

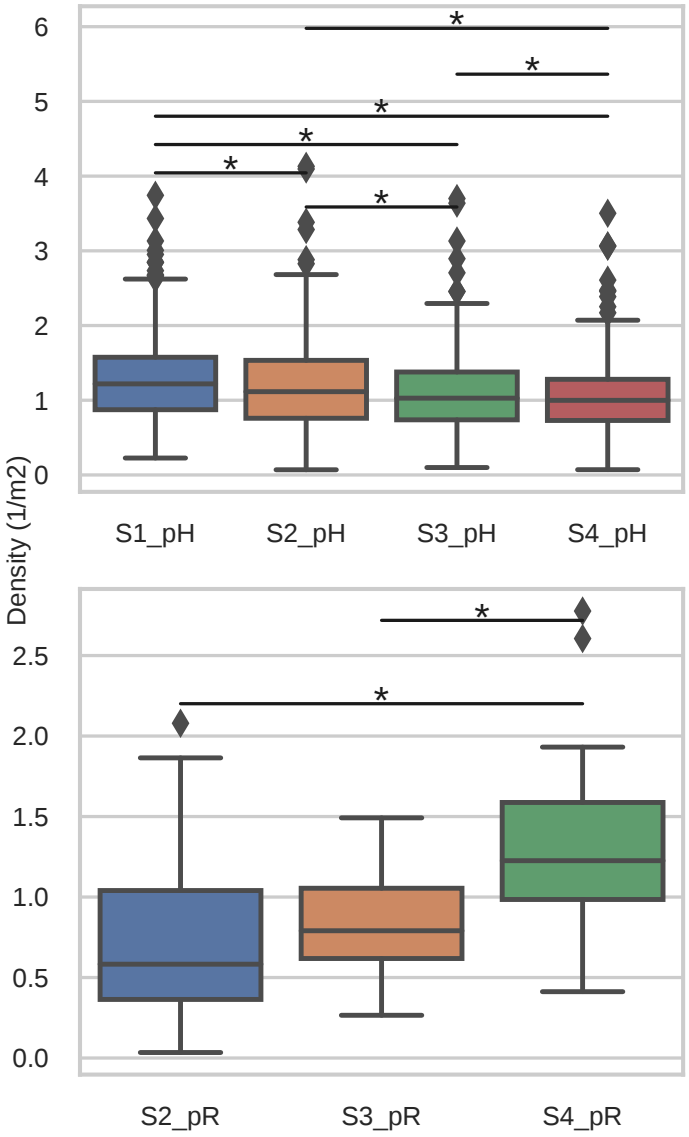


Figure 7.8 – Local density around the pedestrians (pH) and the robot (pR).When a robot is added to the crowd, the local density around pedestrians decreases. Higher local density can be seen around the Pepper robot compared with the smart wheelchair.

7.8 Discussion

We studied how pedestrian dynamics in terms of trajectory regularity and interaction complexity are affected by the occurrence of a wheelchair or a humanoid robot Pepper. In general, the influence on all pedestrians' average speed, acceleration and path efficiency is negligible. However, when we categorized the trajectories based on the spatial relationship to the robot,

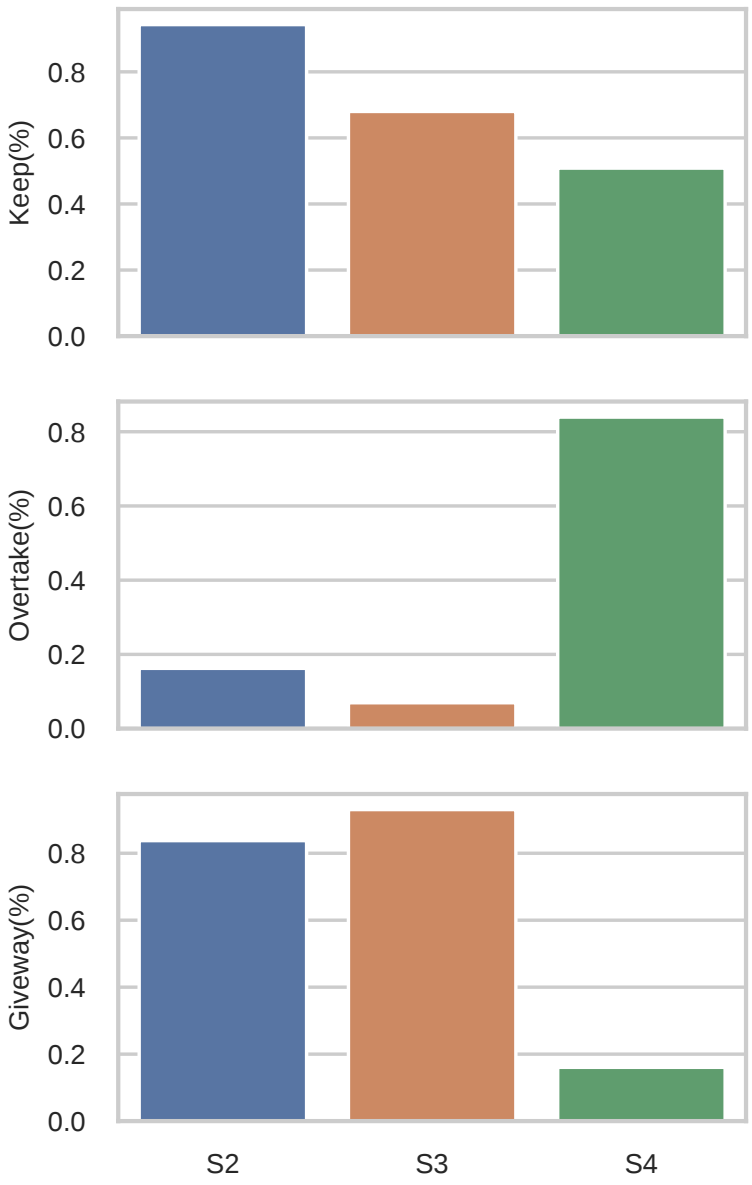
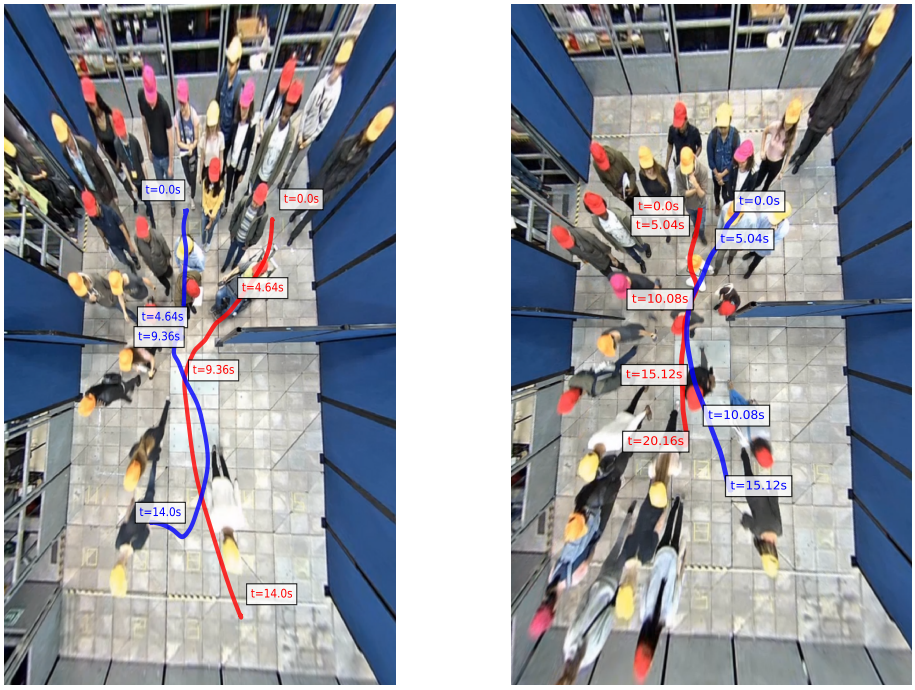


Figure 7.9 – Pass order inversion. In all scenarios, over 50% pedestrians kept their passing order. Among those who adapted their passing order for Pepper, over 80% overtook while less than 20% gave the way to the robot. The situation is the opposite in the wheelchair case.

we have observed significant differences between the 'near the robot' and 'far away from the robot' group in terms of average speed and path efficiency. Pedestrians who are near the robot tend to move slower with lower path efficiency compared to those far away from the robot. No significant evidence has been observed on how the result is affected by the robot speed.

On the other hand, more disparity has been observed across the scenarios in term of the



(a) (b)

Figure 7.10 – Sample robot and pedestrian trajectories. (a): The pedestrian (blue) gave way to the wheelchair (red). Right: The pedestrian (blue) overtake Pepper (red).

interaction complexity. Temporally, the evacuation time per person has increased significantly when a wheelchair or Pepper is involved. Besides, the crowds tend to spend more time in evacuation with the Pepper (S4) than with the wheelchair (S2, S3). In addition, a negative relationship between the robot speed and crowd evacuation time is observed. When looking at individual's adaption behavior, most pedestrians make adaption by overtaking Pepper while giving the way to the wheelchair. A potential explanation for this result is that the pedestrians overtaking decision is affected when a human (the wheelchair driver) is involved. This finding also suggests that pedestrians tend to follow 'social rules' and being polite by not overtaking other pedestrians, while this rule is not observed when the overtaking target is replaced by a humanoid robot.

Spatially, a similar value of about 1.5 $person/m^2$ for local density around pedestrians has been observed for all scenarios. However, this value is significantly lower around the wheelchair and the Pepper. In addition, wheelchair was surrounded by fewer people compared to Pepper.

These results suggest that the pedestrians' local adaption behavior and crowds interaction complexity are affected by the occurrence of a robot, and would vary across the tested two robot platforms.

Consequently, we draw the following implication from our study:

1. It would be important to consider the effect the robot exert on the surrounding pedestrians while planning for its next motion – which means prediction and planning should be considered together to capture the nature interaction in complex environment.
2. In order to achieve social robot navigation using data driven method, pure pedestrian data recorded from a no-robot environment may be insufficient, and it is better to obtain the pedestrian data where the specific robot is involved, as the occurrence and type of the robot affect pedestrian's trajectory regularity as well as crowd interaction complexity. In addition, by modeling the difference in pedestrian dynamics in a crowd simulator, it is potentially possible to achieve more realistic interaction between pedestrians and the robot, and thus providing us a more powerful tool to validate the developed navigation algorithm.
3. The social navigation strategy should potentially be developed differently for the shared-control wheelchair and the fully autonomous humanoid robot. In this study, we did not explore the influence of the existence of a human driver on the pedestrian's behavior, but rather considered the wheelchair +driver as a whole. However, when it comes to the social shared-control navigation, the role of the driver and his/her interaction with surrounding pedestrians would need to be further investigated.

We acknowledge the existence of limitations in our study. Firstly, the result is inherently limited to the certain scenarios as with any HRI study. Furthermore, the fact that human interaction with a robot is associated with its perception which may change over time, experience and environment.

7.9 Conclusion

In this chapter, we presented the first crowd-robot crossing experiment with collected trajectory dataset in the presence of two robot representatives: a smart wheelchair and a Pepper humanoid robot. Quantitative analysis implies the presence of the wheelchair and the Pepper affect crowd dynamics both locally and globally. Besides, the influence varies across the robot type. In general, the effect is reflected in the individual trajectory regularity and the interaction complexity. Qualitative results further supported the idea that pedestrians tend to behave more conservatively around the wheelchair compared to the Pepper, potentially due to the perception of a human driver. These results suggest that the influence of the robot on crowds should be taken into consideration when designing the pedestrian model in simulation and the navigation strategy for different kinds of robot.

In the future, this work could be extended to explore the effect of different types of robot on pedestrian dynamics in bi-directional or even more complex scenarios. In addition, the human

factors such as age, gender, familiarity with the robot which would potentially affect crowd dynamics in social navigation could be further investigated.

Chapter 8

Conclusions and Future Work

In this thesis, new insights into human motion prediction are highlighted and studied. We proposed our contributions through Chapters 3 to 7. In this chapter we briefly review those contributions, comment a few other results we obtained in this thesis and also discuss their limitations and -some- possible future work.

8.1 Benchmarking Human Trajectories Datasets

8.1.1 Contributions

In Chapter 3 we presented a benchmarking framework for assessing prediction complexity in human trajectories datasets. We proposed a series of indicators for gaining insight into the intrinsic complexity of Human Trajectory Prediction datasets. We divided these indicators into three folds:

— **Trajectory Predictability**: We proposed to fit a GMM to the samples of a trajectory dataset and find the number of clusters, which can be an indicator of multi-modality of a dataset. We also proposed formulations to measure the overall and conditional entropies, which can reflect the predictability of dataset samples.

— **Trajectory Regularity**: We considered very basic statistics of motion properties such as average, maximum or range of speed and acceleration of pedestrians. We also defined two indicators for measuring the non-linearity of trajectories: path efficiency, and angular deviation.

— **Context Complexity**: We defined indicators to quantify the interaction strength between pedestrians. We also considered global- and local-density to describe trajectory datasets.

We then applied these indicators on a range of different HTP datasets, including very common datasets such as ETH [PESVG09] and UCY [LCL07] that are massively used in the literature. We showed that these datasets exhibit different characteristics, in the light of our indicators. In particular, it may explain why some prediction techniques that do not use explicit modeling of social interactions, and consider trajectories as independent processes, may be rather successful on datasets where e.g., most trajectories have low collision energy; it may also indicate that

some of the more recent datasets with higher levels of density and interaction between agents could provide more reliable information on the quality of the prediction algorithm.

We also proposed more **technical contributions**: We defined *trajlets*: sub-trajectories with a fixed duration, to decompose each dataset to primitive elements. We created a toolkit containing the indicators, parsers for a variety of trajectory dataset templates, and pre-processing functionalities. The toolkit is implemented with Python and the source codes are published on Github [1].

8.1.2 Future Work

The proposed framework, actually comes with a lot of limitations. One major direction to improve this framework is to design other context-complexity indicators: what is the frequency of other types of social interactions (grouping behavior, leader-follower behavior, and etc) in different datasets. Also the semantic information in the map should be studied, to see how the prediction can be complex around different areas or objects in the map. Moreover, it will be interesting to find statistical correlation between indicators. This can help to find and remove redundant indicators reach a smaller combination of them that cover the desired criteria.

More recent datasets need to be considered in the future studies. Bigger datasets such as *Waymo* [SKD+19] and *NuScenes* [CBL+20] are becoming new benchmarks for prediction systems in self-driving cars. Moreover, these data are being used to train prediction models. Then, it is more important than ever to study these datasets to ensure that they cover a wide variety of samples.

8.2 Short-term Motion Prediction using Crowd Models

This section discusses the work we have done on applying crowd modeling algorithms to the human trajectory prediction problem. In this study, we developed a prediction system, based on the "Crowd Model Optimization framework" proposed by Wolinski et al. [WJGO+14]. We re-designed the framework to work in online mode, and be applicable for real-time trajectory prediction on mobile robots. The experimental results however are not satisfying. The approach is not successful in achieving good prediction performance, compared to very simple baseline. Hence, we decided to exclude this work from main body of this thesis, and discuss it in this concluding chapter.

By Crowd Model we mean "algorithms that model the motion of multiple agents, possibly many, taking into account the interactions with other agents and with static obstacles in the environment". Such algorithm requires the following inputs:

1. github.com/amiryanj/OpenTraj

(a) environment layout, $\mathbf{L}$ including the obstacles and the borders of the environment. This information should define the walkable/non-walkable areas,

(b) agents and their parameters: including N, the number of agents, and $\mathbf{P}$, the set of algorithm parameters. Usually $\mathbf{P}$ is composed of $\mathbf{p}^i$, the parameters corresponding to each agent, with $\mathbf{p}^i = \{p_1^i \; .. \; p_m^i\}$ $(1 \leq i \leq N)$, where m is the number of parameters for one agent. The parameters depend on the crowd algorithm. *Agent radius* and *preferred speed* are two common parameters, used in most of the algorithms. But generally any continuous or categoric parameter can be considered in $\mathbf{P}$.

(c) initial locations and velocities and their destinations. The destinations can be given in the form of a location point/area or a direction vector.

The crowd simulation function can be formulated by the following equation:

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{v}_{k+1} \end{bmatrix} = f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{z}, \mathbf{L}, \mathbf{P}), \tag{8.1}$$

where $\mathbf{x}, \mathbf{v}$ and $\mathbf{z}$ represent the set of locations, velocities and destinations of the agents, respectively, and $\mathbf{L}$ is the layout information. Also, k denotes the time index. The function returns the next (future) location and velocity vectors of the agents. We used the optimization framework, proposed by Wolinski et al. [WJGO+14], that finds the set of parameters $\mathbf{P}$ of model, such that it can replicate a given crowd experiment as 'best' as possible. Considering reference data as $\mathcal{H}$, the framework tries to solve the following optimization problem:

$$\mathbf{P}^* = \arg\min_{\mathbf{P}} \mathbb{E}_{(\mathbf{x},\mathbf{v},\mathbf{z})\sim\mathcal{H}}[d(\mathbf{x}_{k+1}, f(\mathbf{x}_k, \mathbf{v}_k, \mathbf{z}, \mathbf{L}, \mathbf{P}))], \tag{8.2}$$

where d is a distance function. We then applied the following modifications on this framework. We formulated the prediction problem as solving Eq. (8.2) for observed trajectories of agents, and use $\mathbf{P}^*$ to predict the future location of agents. By iteratively applying f on the last observations (or estimations), we can predict trajectories for each agent. To solve this problem, we also need to estimate the agents' destinations $\mathbf{z}$. We considered two different approaches for agents' destinations: (1) estimation agent's goal by extrapolating its instant motion vector, (2) using ground truth goal, for the sake of comparison.

We tested different crowd simulation algorithms: Helbing (Social-Forces) [HM95], PowerLaw [KSG14], RVO [vdBGLM11], different optimization methods: Greedy, Genetic Algorithm, and Simulated Annealing (all explained in [WJGO+14]). We predicted the location of agents for the next 1 sec for three different datasets: ETH-Univ, ETH-Hotel [PESVG09] and high-density crowd Bottleneck dataset [SPS+09]. In the experiments, we observed lower prediction error for RVO, compared to Helbing and PowerLaw, and the best optimization results using the Genetic Algorithm. The normalized prediction errors are shown in Fig. 8.1, only for RVO model:

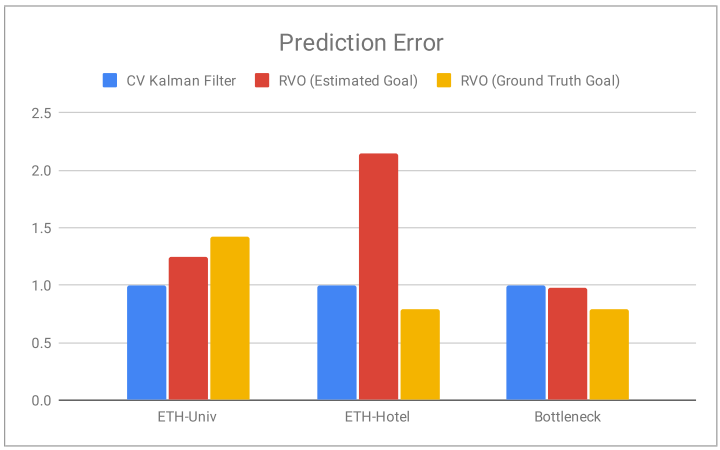


Figure 8.1 – Normalized prediction results using RVO crowd model, with and without goal estimation (red and yellow bars respectively) vs. Constant-Velocity Kalman Filter (blue bar).

As we can see, we almost did not achieve any improved motion predictions using this approach, compared to the baseline, which is a Constant-Velocity Kalman Filter. Only for Bottleneck dataset, the prediction error was slightly lower than the Kalman Filter baseline. Also the prediction error when using ground truth goals was lower for ETH-Hotel and Bottleneck datasets.

We attribute this behavior to the fact that we only used a very narrow window of the observed trajectories to find the optimum parameters $\mathbf{P}^*$. For instance, in short chunks of ETH datasets, there is not sufficient amount of interaction between agents, and the parameters are obtained sub-optimally. On the other hand, in the Bottleneck dataset, which contains high density crowd activities, the approach has provided slightly better predictions. Further, the goal estimation is not handled in an intelligent way, in this setup, and there is room to improve this aspect, for example by using statistical inference.

Our **technical contributions** include developing the mentioned system in C++, and creating a Qt-based graphical user interface to run and test crowd simulation algorithms on recorded datasets. A snapshot of our program is shown in Fig. 8.2.

8.2.1 Future Work

While the experiments in this section, does not show any benefit of using crowd models for prediction of human trajectories, however the intrinsic properties of this algorithm can still be useful in the domain of trajectory prediction. This family of algorithms, provide very simple and interpretable models of interactions between agents, and can be used in combination with

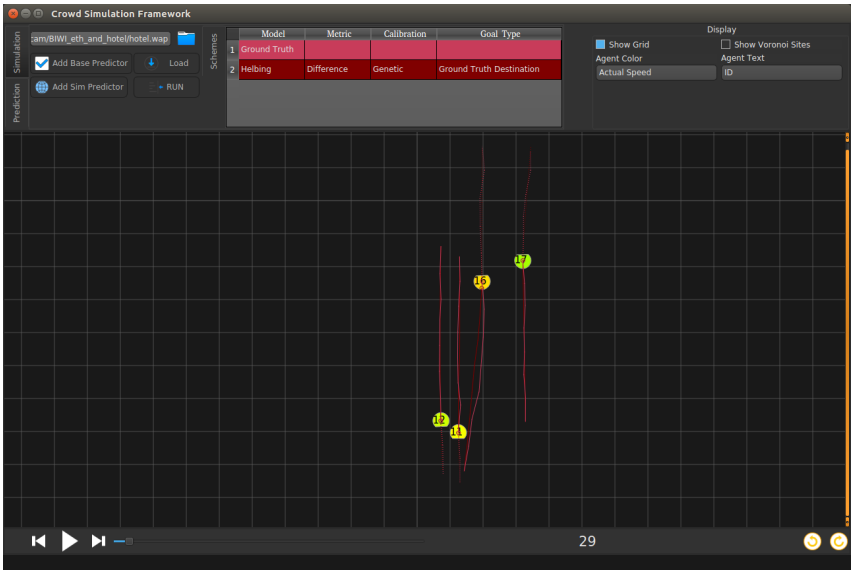


Figure 8.2 – Screen-shot of the user interface of the crowd prediction framework. The main window contains a scene to display the world in 2D, a menu at the top and some control buttons at the bottom. In the scene, the agents are shown by yellow/green circles and the ground truth predictions are shown in punch color, the results of Helbing model are shown in red.

pattern-based models. Another direction is to use these models for generating synthetic crowd, which can improve the training in data-driven models.

8.3 Multi-Modal Pedestrian Trajectory Prediction

In Chapter 4 we presented a novel approach for prediction of pedestrians trajectories. The proposed model is based on Generative Adversarial Networks.

8.3.1 Contributions

In the proposed architecture we used a few hand-designed interaction features inspired from the neuroscience/bio-mechanics literature. We used an attention pooling to process the social features of pedestrians. We also proposed to use the infoGAN approach, to alleviate the common *mode collapsing* effect observed in generative models. We showed that the proposed loss function, can help to better preserve the modes of predictive distributions.

We proposed a specifically designed toy trajectory dataset to use as an evaluation benchmark for comparing different approaches and baselines. We presented two new metrics to measure the quality of the prediction distributions:

1. **1-Nearest Neighbor classifier** assesses the distinguishability between real and fake (generated) samples,

2. **Earth-Mover's Distance** estimates the distance between the distributions of real and fake predicted trajectories.

We showed through evaluations on commonly used datasets that our approach partly improves the prediction accuracy of state-of-the-art methods on the datasets where the predictive distributions have the largest variances.

We implemented the proposed model, using Pytorch machine learning library and in Python language. The source codes are published on a Github repository [2].

8.3.2 Discussion and Future Work

We are aware that there is still room for improving the current generative models in pedestrian motion prediction and, above all, for exploiting these models in decision making.

Learning disentangled representations: To evaluate the benefits of using the information loss and the associated latent codes $\mathbf{c}$, we trained a GAN with information loss on a toy dataset and conducted generation experiments with two continuous latent codes $\mathbf{c} \in \mathbb{R}^2$. Then we studied qualitatively the generated trajectories when varying $\mathbf{c}$, for a given set of past observations. The results appear in Fig. 8.3 indicate that the variations in the first component seem to result in speed change in the trajectory, while the second component seems to control the steering angle of the trajectory.

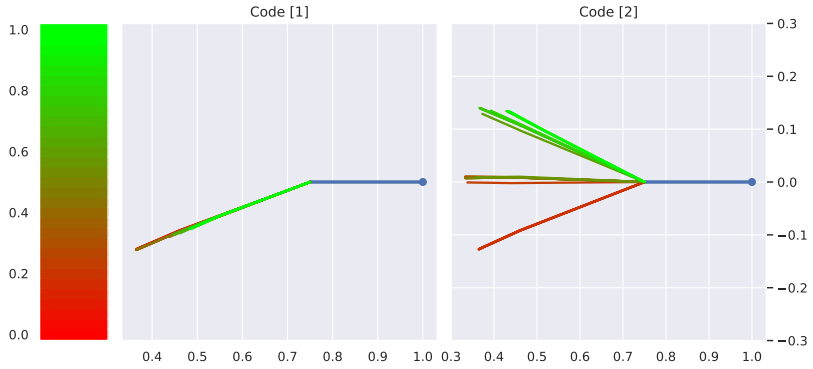


Figure 8.3 – Effect of varying the latent codes on the generated trajectories. The past observations appear in blue. On the left, variations in the first latent code generate trajectories aiming to the same direction but with different speed (length); On the right, variations in the second latent code generate trajectories with approximately the same speed but different steering angles.

We proceeded to test with different numbers of latent codes. However, we did not observe meaningful changes in the generated trajectories, after the second code. This result is however derived from a limited test on a small toy dataset, which is itself in a low-dimensional space.

2. github.com/amiryanj/socialways

Learning disentangled representations for human trajectories on real and large-scale datasets can be studied, in the future. It is also interesting to see if agents' interactions can be learned via disentangled representations.

Very recently, Kothari et al. [KSA21] have addressed the *interpretability* issues in trajectory prediction models using Discrete Choice Modelling (DCM). The model which has similarities with traditional hand-crafted algorithms for human motion prediction [RABC09, YBOB11], while leveraging neural networks to model complex and possibly subtle social interactions, successfully outperforms the state-of-the-art trajectory prediction models on the TrajNet++ benchmark [SICP20].

Sequence-to-sequence learning: In the neural-network system we presented in Chapter 4, we used recurrent networks cells for both encoding and decoding trajectories. This type of networks, however, suffer from vanishing (or exploding) gradient problem. This issue relates to back-propagating the training error through a long sequence of inputs (i.e., location of agents). The derivatives corresponding to consequent inputs are multiplied, hence this last expression tends to vanish more as the sequence gets larger.

Even though Long-Short Term Memory networks (LSTMs) are proposed to diminish vanishing gradient, the problem however is not completely solved. Therefore the trained network can be sub-optimal and unable to encode trajectory inputs effectively.

Transformer is an encoder-decoder architecture that is proposed for sequence-to-sequence problems without using recurrent networks and only by leveraging attention mechanism. The encoder and decoder each consists of a set of layers that sequentially compute the *self-attention* between the layer inputs and pass them through a feed-forward layer. The authors of [GHCG20] showed surprisingly that transformers can give competitive results in the trajectory prediction task as well. This model, however has not taken into account the social interactions between the agents. Then a very interesting research direction is to embed social-features and other auxiliary information in this model.

Generative models: As we discussed earlier, Generative Adversarial Networks, suffer from few intrinsic issues:

— **Non-convergence**: GANs are difficult to train. They are highly sensitive to the hyper-parameter selections, and small changes in the parameters may upset the equilibrium between the *Generator* and *Discriminator*. This can then cause one adversary (mostly the Discriminator) to train faster than other one. The result is that the gradient will diminish substantially and the training will not converge or it leads to over-fitting.

— **Mode-collapse**: we addressed this problem in our solution. Using information loss, we showed that the modes of the predictive distribution will better be preserved, compared to other baseline GANs, including vanilla-GAN (trained by *adversarial loss*), variety loss, and so on. However, maximizing the mutual information between the input and output of the Generator, solves this problem only partially, and this approach does not guarantee to learn all the modes of the data. Another problem arises from the sampling process, where we draw a set of independent Gaussian latent codes and convert them into predicted trajectories. This independent sampling, however, may fail to provide diverse outputs.

The above problems can be relieved or solved through future work. Some recent work has already applied better ideas and has shown improved results, while some ideas remain to be explored. Yuan and Kitani introduced Diversity Latent Flows (DLow) to produce a diverse set of samples from a pre-trained deep generative model, in the task of human pose prediction [YK20]. Instead of using independent sampling as in GANs, they sample a single random variable and map it with a set of learnable mapping functions to a set of correlated latent codes. These codes are then decoded into a set of correlated prediction outputs, which can improve the diversity of the samples. This concept can also be applied to predicting human trajectories.

Also, other generative models can be applied to the problem of trajectory prediction. Some recent models, based on conditional VAEs show promising and more stable results. The authors of *Trajectron* [IP19] achieved this by training a graph network using Conditional Variational Auto-Encoding (CVAE). A follow-up work, coined as *Trajectron++* [SICP20], obtained better prediction results compared to GAN-based counterparts such as *Social-Ways* (our proposed model), *Social-GAN* [GJFF+18], *Sophie* [SKS+18] and *Social-BiGAT* [KSMM+19].

However, a recent argument was raised by researchers at Leuphana University of Lüneburg that "variational auto-encoding does not contribute statistically significantly to empirical performance in modeling low-dimensional trajectories [RBD20] of agents." They performed multiple ablation tests, between models with/without a variational unit and suggested that the latter perform on par or better than former systems. They disabled the CVAE component of the *Trajectron* to obtain two model variants: one by interchanging the CVAE with a Gaussian Stochastic Neural Network (GSNN) component [SLY15]. And another by removing the latent variable component altogether. They showed that there is no significant difference in the model variants with respect to their log-likelihoods on five tested datasets.

There are also other generative models that explicitly learn the probability density function of data, such as Flow-based models [DKB14, KD18]. Some attempts are done to use normalizing flows in trajectory prediction problem. Bhat et al. [BFO20] proposed a model based on Transformers and Normalizing flows for Autonomous driving. Also, Fadel et al. [FMTB21] developed a movement model for prediction of trajectories of soccer players. Those models do not consider

the interactions of agents for prediction. Hence, there is room to improve these models and apply them on social robots.

Evaluation of trajectory prediction models: The metrics we proposed in Chapter 4 are useful tools for evaluation of multi multi-modal prediction models. However in our work, we only used them on the toy dataset, where for every observed trajectory we have access to a set (or probability distribution) of the future trajectories. Both metrics, also need to compare two sets with equal number of elements. In order to use those metrics on real datasets, some modifications are required. For example, a clustering mechanism can be used to treat trajectories whose observed parts are similar, as one set. Also, one can *up-sample* these clusters to obtain sets of equal sizes.

It is also important to reconsider the k-ADE and k-FDE metrics when evaluating multimodal models. Kothari et al. [KKA21] suggest using a small k, like 3 instead of 20 (which we used in Chapter 4). Because a model that produces uniform-spaced predictions, regardless of the input observation, can still result in low k-ADE and k-FDE when k is too large.

8.4 Occluded Crowd Prediction from Robot Sensing

8.4.1 Contributions

In Chapter 5 we proposed a new approach to impute the structure of a crowd in which a robot is performing navigation with its limited sensing. We leverage several new concepts that we have introduced to describe crowd patterns around the robot (strong and absent ties, communities and territories) to form a generative model for crowd patterns and use it to samples of imputed occupation maps, based on what is observed through the robot perception. We have shown on real crowd datasets that the proposed indicators reflect the nature of the typical pairwise relation within crowds, and we have obtained competitive prediction results, in particular on datasets with well-structured pedestrian flows.

8.4.2 Future Work

According to our best knowledge, the prediction of occluded crowds using the inter-pedestrian patterns has never been studied before. Despite that, it has been shown that the model's accuracy is lacking in some environments, particularly in low-density spaces. However, we believe that this study can open up a lot of interesting questions to the crowd prediction area. There are so many research questions to be explored further.

Deep Generative Models can be used to learn the more complex interactions between pedestrians. These models may be used to handle joint sequences of locations rather than individual

frames. Further, they can leverage the *Kernel trick* to transform these interactions to higher dimensional spaces, which allows for disclosing hidden patterns in data. It is also possible to improve the concept of social ties. In our study we proposed a binary definition, that classifies interactions into strong and absent. However, a fuzzy or continuous formulation could be more natural. Further, we can use a non-interpretable and high-dimensional variable to represent social interactions.

In the future, *domain adaptation* techniques can be used to transfer knowledge from one domain or environment to another. We showed that different distributions of interactions can be associated with different types of environments or various crowd densities. As such, domain adaptation can be useful in making more generalized models.

The proposed system can be integrated with a Human Trajectory Prediction module, to complement each other and make predictions in both time and space domains.

8.5 Data-driven Crowd Simulation

8.5.1 Contributions

In Chapter 6 we presented a data-driven crowd simulation method that uses GANs to learn the properties of input trajectories and then generate new trajectories with similar properties. Combined with flexible route following that takes temporal information into account, the trajectories can be used in a real-time crowd simulation. Our system can be used, for example, to create variants of a scenario with different densities. It can easily be combined with other simulation methods, and it allows interactive applications.

8.5.2 Future Work

In the future, we will perform a thorough analysis of the trajectories produced by our system, and compare them to other algorithms. We will also investigate the exact requirements for reliable training. Furthermore, our system generates trajectories for individuals, assuming that agents do not influence each other's choices. As such, it cannot yet model group behavior, and it performs worse in high-density scenarios where agents cannot act independently. We would like to handle these limitations in future work. In addition, many of the discussions we made about the future of the 'Social-Ways' work (in Sec. 8.3.2) are applicable for this problem as well.

8.6 Crowd-Robot Interaction Study

8.6.1 Contributions

In Chapter 7 we presented the first crowd-robot crossing experiment with collected trajectory dataset in the presence of two robot representatives: a smart wheelchair and a Pepper humanoid robot. Our quantitative analysis implies that the presence of the wheelchair and the Pepper affect crowd dynamics both locally and globally. Besides, the influence varies across the robot type. In general, the effect is reflected in the individual trajectory regularity and the interaction complexity. Qualitative results further supported the idea that pedestrians tend to behave more conservatively around the wheelchair compared to the Pepper, potentially due to the perception of a human driver. These results suggest the influence of the robot on crowds should be taken into consideration when designing the pedestrian model in simulation and the navigation strategy for different kinds of robot.

8.6.2 Future Work

In the future, this work could be extended to explore the effect of different types of robot on pedestrian dynamics in bi-directional or even more complex scenarios. In addition, the human factors such as age, gender, familiarity with the robot which would potentially affect crowd dynamics in social navigation could be further investigated.

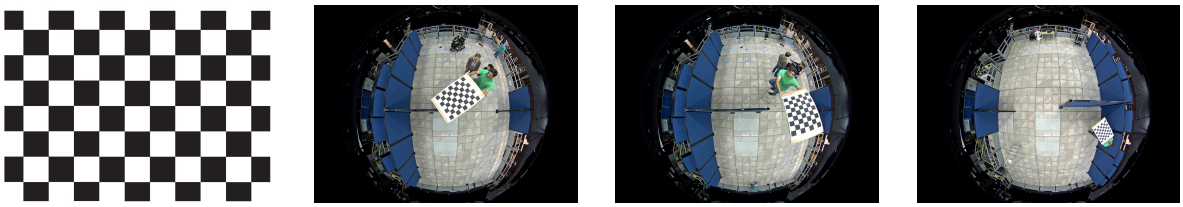
Appendix A

Appendix

A.1 Data Collection of Crowd-Robot Experiment

A.1.1 Camera Calibration

The experiment was recorded by an overhead fish-eye IP video camera (Axis M3037) at 12.5 frame per second. In order to calibrate the camera, we used the chessboard method. We recorded a video with the camera, while a person was holding the chessboard toward the camera, moving around and rotating it. We extracted 22 different frames from this video to cover different positions and rotations of the chessboard in the image (such as the ones in Fig. A.1)



(a) (b) (c) (d)

Figure A.1 – (a): Chessboard pattern for camera calibration. (b), (c), (d): chessboard in different positions and orientations

We used a pinhole model to calibrate the camera, in which, the parameters of calibration can be divided into two groups: the *extrinsic* and *intrinsic* parameters. The extrinsic parameters consist of a rotation, R, and a translation, T. The origin of the camera's coordinate system is at its optical center and its x- and y-axis define the image plane. Using a pinhole camera model we have the following equation:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \tag{A.1}$$

where $[X_w, Y_w, Z_w]^T$ is a three-dimensional point in world coordinate, and $[X_c, Y_c, Z_c]^T$ is the

projected point on the camera plane. For estimating the intrinsic parameters, we used the Scaramuzza's Omnidirectional camera model [SMS06a]:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \\ a_0 + a_2\rho^2 + a_3\rho^3 + a_4\rho^4 \end{bmatrix}, \tag{A.2}$$

where, λ represents a scaling factor, a_0, a_2, a_3, a_4 are the polynomial coefficients described by the Scaramuzza model, with a_1 being zero, (u, v) is the ideal image projections of the real-world points and $\rho = \sqrt{u^2 + v^2}$, is the distance of this point from the image center.

Finally the relation between the real distorted coordinates (u'', v'') and the ideal distorted coordinates (u, v) can be written as:

$$\begin{bmatrix} u'' \\ v'' \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} c'_x \\ c'_y \end{bmatrix} \tag{A.3}$$

where c, d, e describes a matrix that performs a *stretch* operation, and $[c'_x c'_y]^T$ is the distortion center. We used Matlab Camera Calibrator toolbox [SMS06b] for estimating the above coefficients which then allowed us to undistort the videos (see Fig. A.2).

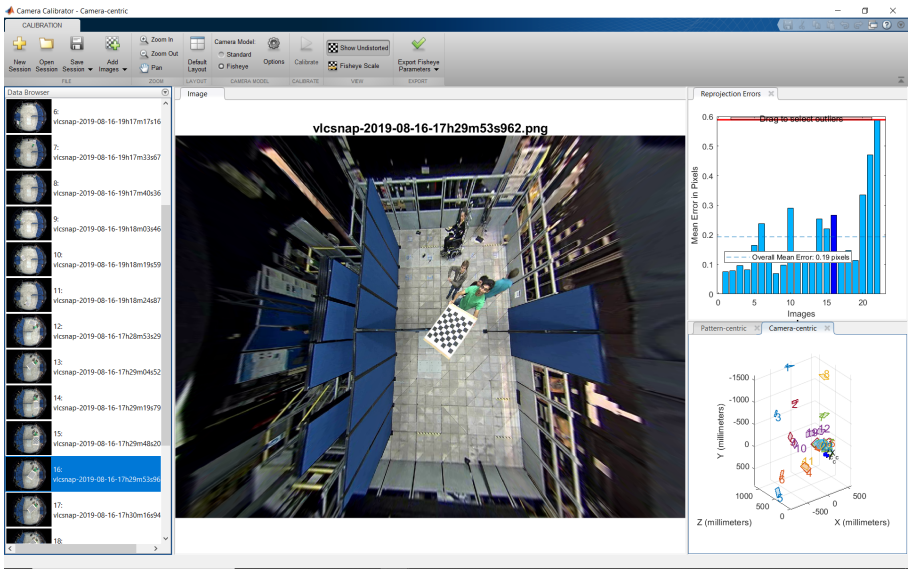
A.1.2 Tracking Participants

We then used a special-purpose software called PeTrack to extract the pedestrian trajectories. With PeTrack we were able to track the colorful helmets of pedestrians and the robot (or wheelchair user). We then fixed the problems like ID switches and drifts in the detections by semi-automatic processes. Finally, we used a linear projection to map the head of the pedestrian to a point on the ground, which represents the leg of the person. For the humans we assumed an average height of 170 cm. For Pepper and the wheelchair we used heights of 120cm and 140 cm, respectively.

In order to filter out the high frequency jerks from the trajectories we applied a Kalman Filter (KF) with Rauch–Tung–Striebel (RTS) smoothing backward-pass. Considering the transition model of the standard KF, without the control input, we can write:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}\ \mathbf{x}_{k-1} + \mathbf{w} \\ \mathbf{w} &\sim \mathcal{N}(0, \mathbf{Q}), \end{aligned} \tag{A.4}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}$ is a zero-mean Gaussian random variable with



(a) screenshot of the Matlab Camera Calibrator toolbox

(b) distorted frame

(c) undistorted frame

Figure A.2 – Estimating Scaramuzza's camera calibration parameters, and un-distorting the recorded videos

covariance matrix $\mathbf{Q}$, called *process noise*. Also the observation model is:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{H}\ \mathbf{x}_k + \mathbf{v} \\ \mathbf{v} &\sim \mathcal{N}(0, \mathbf{R}), \end{aligned} \tag{A.5}$$

where $\mathbf{H}$ is the observation matrix and $\mathbf{v}$ is another zero-mean Gaussian with covariance matrix $\mathbf{R}$, called *observation noise*. We use a constant-acceleration model, for $\mathbf{x}$, where the transition model, for one dimension (x) is given by:

$$\mathbf{F}^x = \begin{bmatrix} 1 & dt & \frac{1}{2}dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix} \tag{A.6}$$

We configured the model with $dt = \frac{1}{12.5}$, $\mathbf{Q} = 10\ \mathbb{I}_{2\times 2}$ and also $\mathbf{R}$ and $\mathbf{H}$ with identity matrices. In total, 20 trials (for four scenarios, five runs each scenario) were performed with valid interactions between the robot and the crowds. 546 trajectories of pedestrians and the robots were extracted.

List of publications

International Journals

— Bingqing Zhang, **Javad Amirian**, Harry Eberle, Julien Pettré, Catherine Holloway, and Tom Carlson, "**From HRI to CRI: Crowd Robot Interaction - Understanding the Effect of Robots on Crowd Motion**", International Journal of Social Robotics, 2021.

International Conferences and Workshops

— **Javad Amirian**, Jean-Bernard Hayet, and Julien Pettré. "**Social Ways: Learning Multi-modal Distributions of Pedestrian Trajectories with GANs.**" In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

— **Javad Amirian**, Wouter van-Toll, Jean-Bernard Hayet and Julien Pettré. "**Data-driven Crowd Simulation with Generative Adversarial Networks**." In Proceedings of the 32nd International Conference on Computer Animation and Social Agents (CASA), 2019.

— **Javad Amirian**, Bingqing Zhang, Francisco Valente Castro, Juan Jose Baldelomar, Jean-Bernard Hayet, and Julien Pettré. "**Opentraj: Assessing prediction complexity in human trajectories datasets.**" In Proceedings of the Asian Conference on Computer Vision (ACCV), 2020.

— Wouter van Toll, Fabien Grzeskowiak, Axel López, **Javad Amirian**, Florian Berton, Julien Bruneau, Beatriz Cabrero, Alberto Jovane, and Julien Pettré. "**Generalized Microscropic Crowd Simulation using Costs in Velocity Space.**" In Symposium on Interactive 3D Graphics and Games (i3D), 2020.

— **Javad Amirian**, Jean-Bernard Hayet and Julien Pettré. "**What we see and What we don't see: Imputing Occluded Crowd Structures from Robot Sensing**." In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021. (Under Review)

List of Figures

List of Tables

List of acronyms

ADE	Average Displacement Error
ANN	Artificial Neural Network
CNN	Convolution Neural Network
DCA	Distance to Closest Approach
EMD	Earth Mover's Distance
FDE	Final Displacement Error
HTP	Human Trajectory Prediction
HMM	Hidden Markov Model
IRL	Inverse Reinforcement Learning
KDE	Kernel Density Estimation
KNN	K-Nearest Neighbors
GA	Genetic Algorithms
GAIL	Generative Adversarial Imitation Learning
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GP	Gaussian Process
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memories
LTA	Linear Trajectory Avoidance
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
PCF	Pair Correlation Function
PDF	Probability Density Function
PMF	Probability Mass Function
PRM	Probabilistic Road-Map
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
RVO	Reciprocal Velocity Obstacles
TTC	Time To Collision
SFM	Social Forces Model
VAE	Variational Auto-Encoders
VRU	Vulnerable Road Users

BIBLIOGRAPHY

[AGR+16] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, June 2016.

[AHP19] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[AHTZ20] Jorge Agnese, Jonathan Herrera, Haicheng Tao, and Xingquan Zhu. A survey and taxonomy of adversarial neural networks for text-to-image synthesis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1345, 2020.

[ARFF14] Alexandre Alahi, Vignesh Ramanathan, and Li Fei-Fei. Socially-aware large-scale crowd forecasting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2203–2210, 2014.

[AZC+20] Javad Amirian, Bingqing Zhang, Francisco Valente Castro, Juan Jose Baldelomar, Jean-Bernard Hayet, and Julien Pettré. Opentraj: Assessing prediction complexity in human trajectories datasets. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[BA01] JohnTravis Butler and Arvin Agah. Psychological effects of behavior patterns of a mobile personal robot. *Autonomous Robots*, 10:185–202, 03 2001.

[Bas50] Frédéric Bastiat. Ce qu'on voit et ce qu'on ne voit pas. `http://bastiat.org/fr/cqovecqonvp.html`, 1850.

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[BCC+20] Alessia Bertugli, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. Ac-vrnn: Attentive conditional-vrnn for multi-future trajectory prediction. *arXiv preprint arXiv:2005.08307*, 2020.

[BDMFK91] Yoshua Bengio, Renato De Mori, Giovanni Flammia, and Ralf Kompe. Neural network-gaussian mixture hybrid for speech recognition or density estimation. *Advances in Neural Information Processing Systems*, 4:175–182, 1991.

[BF15] Graeme Best and Robert Fitch. Bayesian intention inference for trajectory prediction with an unknown goal destination. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5817–5823. IEEE, 2015.

[BFO20] Manoj Bhat, Jonathan Francis, and Jean Oh. Trajformer: Trajectory prediction with local self-attentive contexts for autonomous driving. *arXiv preprint arXiv:2011.14910*, 2020.

[BHHA18] Stefan Becker, Ronny Hug, Wolfgang Hübner, and Michael Arens. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. *arXiv preprint*, abs/1805.07663, 2018.

[BKIM13] D. Brscic, T. Kanda, T. Ikeda, and T. Miyashita. Person position and body direction tracking in large public spaces using 3d range sensors. *IEEE Transactions on Human-Machine Systems*, 43(6):522–534, 2013.

[BKM+19] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. *arXiv preprint*, abs/1911.07692, 2019.

[BKR+16] Aniket Bera, Sujeong Kim, Tanmay Randhavane, Srihari Pratapa, and Dinesh Manocha. Glmp-realtime pedestrian path prediction using global and local movement patterns. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5528–5535. IEEE, 2016.

[BLM19] Ulf Brefeld, Jan Lasek, and Sebastian Mair. Probabilistic movement models and zones of control. *Machine Learning*, 108(1):127–147, 2019.

[BR09] Ben Benfold and Ian Reid. Guiding visual surveillance by tracking human attention. In *Proc. of the British Machine Vision Conference (BMVC)*, 2009.

[BRM17] Aniket Bera, Tanmay Randhavane, and Dinesh Manocha. Aggressive, tense, or shy? Identifying personality traits from crowd videos. *IJCAI International Joint Conference on Artificial Intelligence*, 0:112–118, 2017.

[Buc10] Randy L Buckner. The role of the hippocampus in prediction and imagination. *Annual review of psychology*, 61:27–48, 2010.

[BZC18] Niccoló Bisagno, Bo Zhang, and Nicola Conci. Group lstm: Group trajectory prediction in crowded scenarios. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[BZH+18] M. Bieshaar, S. Zernetsch, A. Hubert, B. Sick, and K. Doll. Cooperative starting movement detection of cyclists using convolutional neural networks and a boosted stacking ensemble. *arXiv preprint*, abs/1803.03487, 2018.

[BZM+20] Huikun Bi, Ruisi Zhang, Tianlu Mao, Zhigang Deng, and Zhaoqi Wang. How can i see my future? fvtraj: Using first-person view for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 576–593. Springer, 2020.

[CBB+18] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur Bagautdinov, Louis Lettry, Pascal Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5030–5039, June 2018.

[CBCB14] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*, 2014.

[CBL+20] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, June 2020.

[CDH+16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[CH08] Gerda Claeskens and Nils Lid Hjort. *The Bayesian information criterion*, page 70–98. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2008.

[CH10] Shu-Yun Chung and Han-Pang Huang. A mobile robot that understands pedestrian spatial behaviors. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5861–5866. IEEE, 2010.

[CJG18] Zhuo Chen, Chao Jiang, and Yi Guo. Pedestrian-robot interaction experiments in an exit corridor. In *2018 15th International Conference on Ubiquitous Robots (UR)*, pages 29–34. IEEE, 2018.

[CKD+15] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *arXiv preprint arXiv:1506.02216*, 2015.

[CLEH17] Yu Fan Chen, Miao Liu, Michael Everett, and Jonathan P How. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 285–292. IEEE, 2017.

[CLKA19] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6015–6022. IEEE, 2019.

[CLS+19] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8748–8757, 2019.

[CSC09] Ujjal Chattaraj, Armin Seyfried, and Partha Chakroborty. Comparison of pedestrian fundamental diagram across cultures. *Advances in complex systems*, 12(03):393–405, 2009.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[DBR09] António Eduardo De Barros Ruano. *Artificial Neural Networks*. Faro, Portugal: University of Algarve, 2009.

[DDFMR13] Arnaud Doucet, Nando De Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. *arXiv preprint arXiv:1301.3853*, 2013.

[Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[DKB14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[DOLT20] 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.

[DT18] Nachiket Deo and Mohan M Trivedi. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs. In *IEEE Intelligent Vehicles Symposium, Proceedings*, volume 2018-June, pages 1179–1184, 2018.

[DTL17] Camille Dupont, Luis Tobias, and Bertrand Luvison. Crowd-11: A dataset for fine grained crowd behaviour analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–16, 2017.

[ELVG07] Andreas Ess, Bastian Leibe, and Luc Van Gool. Depth and appearance for mobile scene analysis. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1–8, 2007.

[ENMGC19] Pierre Ecormier-Nocca, Pooran Memari, James Gain, and Marie-Paule Cani. Accurate synthesis of multi-class disk distributions. In *Computer Graphics Forum*, volume 38, pages 157–168. Wiley Online Library, 2019.

[ESR09] 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.

[FMTB21] Samuel G Fadel, Sebastian Mair, Silva Torres, and Ulf Brefeld. Contextual Movement Models based on Normalizing Flows. 2021.

[For20] Piaggio Fast Forward. Gita. `https://mygita.com/`, 2020.

[FS98] Paolo Fiorini and Zvi Shiller. Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772, 1998.

[FS09] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *Proc. of the IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, pages 1–6, 2009.

[Gau14] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, 2014(5):2, 2014.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, volume 1. 2016.

[GCC+10] Stephen J Guy, Jatin Chhugani, Sean Curtis, Pradeep Dubey, Ming Lin, and Dinesh Manocha. PLEdestrians: A least-effort approach to crowd simulation. In *Computer Animation 2010 - ACM SIGGRAPH / Eurographics Symposium Proceedings, SCA 2010*, pages 119–128, 2010.

[GDB+14] Michael Goldhammer, Konrad Doll, Ulrich Brunsmann, Andre Gensler, and Bernhard Sick. Pedestrian's trajectory forecast in public traffic with artificial neural networks. In *2014 22nd International Conference on Pattern Recognition*, pages 4110–4115, Stockholm, Sweden, August 2014. IEEE.

[GHCG20] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. *ArXiv preprint*, abs/2003.08111, 2020.

[GJ14] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. of the Int. Conf. on Machine Learning (ICML)*, pages II–1764–II–1772. JMLR.org, 2014.

[GJFF+18] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[GLSU13] Andreas Geiger, P Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: the KITTI dataset. *The International Journal of Robotics Research*, 32:1231–1237, 2013.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[Gra73] Mark S Granovetter. The strength of weak ties. *American journal of sociology*, 78(6):1360–1380, 1973.

[GSLS11] Haifeng Gong, Jack Sim, Maxim Likhachev, and Jianbo Shi. Multi-hypothesis motion planning for visual object tracking. In *2011 International Conference on Computer Vision*, pages 619–626. IEEE, 2011.

[Hal09] Edward Hall. A system of notation of proxemic behavior. *American Anthropologist*, 65:1003 – 1026, 10 2009.

[HCGR11] Emili Hernández, Marc Carreras, Enric Galceran, and Pere Ridao. Path planning with homotopy class constraints on bathymetric maps. In *OCEANS 2011 IEEE-Spain*, pages 1–6. IEEE, 2011.

[HD05] Serge P Hoogendoorn and Winnie Daamen. Pedestrian behavior at bottlenecks. *Transportation science*, 39(2):147–159, 2005.

[HE16] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.

[HJAA07] Dirk Helbing, Anders Johansson, and Habib Al-Abideen. Dynamics of crowd disasters: An empirical study. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 75:046109, 05 2007.

[HLK16] Mark Harmon, Patrick Lucey, and Diego Klabjan. Predicting shot making in basketball learnt from adversarial multiagent trajectories. *arXiv preprint*, abs/1609.04849, 2016.

[HM95] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. In *Physical review E*, 1995.

[HNR68] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[HST+18] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco Cristani. Mx-lstm: mixing tracklets and vislets to jointly forecast trajectories and head poses. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[HZC21] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

[int16] SAE international. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE International,(J3016)*, 2016.

[IP19] 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.

[IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[JCIG13] Norman Jaklin, Atlas Cook IV, and Roland Geraerts. Real-time path planning in heterogeneous environments. *Computer Animation and Virtual Worlds*, 24(3):285–295, 2013.

[JHL20] Dan Jia, Alexander Hermans, and Bastian Leibe. DR-SPAAM: A Spatial-Attention and Auto-regressive Model for Person Detection in 2D Range Data. In *International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[KAHS16] Vasiliy Karasev, Alper Ayvaci, Bernd Heisele, and Stefano Soatto. Intent-aware long-term prediction of pedestrian motion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2543–2549. IEEE, 2016.

[Kal60] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[Kam11] Gal Kaminka. The impact of cultural differences on crowd dynamics in pedestrian and evacuation domains. Technical report, BAR ILAN UNIV RAMAT GAN (ISRAEL) DEPT OF COMPUTER SCIENCE, 2011.

[KD18] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

[Ker17] Jeremy Kerfs. Models for pedestrian trajectory prediction and navigation in dynamic environments. 05 2017.

[KG14] Christoph Gustav Keller and Dariu Gavrila. Will the pedestrian cross? a study on pedestrian path prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15:494–506, 2014.

[KGL+15] Sujeong Kim, Stephen J Guy, Wenxi Liu, David Wilkie, Rynson WH Lau, Ming C Lin, and Dinesh Manocha. Brvo: Predicting pedestrian trajectories using velocity-space reasoning. *The International Journal of Robotics Research*, 34(2):201–217, 2015.

[KGM+20] Kapil Katyal, Yuxiang Gao, Jared Markowitz, I Wang, Chien-Ming Huang, et al. Group-aware robot navigation in crowded environments. *arXiv preprint arXiv:2012.12291*, 2020.

[KHvBO09] Ioannis Karamouzas, Peter Heil, Pascal van Beek, and Mark H. Overmars. A predictive collision avoidance model for pedestrian simulation. In Arjan Egges, Roland Geraerts, and Mark Overmars, editors, *Motion in Games*, pages 41–52, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[KKA21] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[KKSB12] Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard. Feature-based prediction of trajectories for socially compliant navigation. 07 2012.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[KLSK96] Timo Koskela, Mikko Lehtokangas, Jukka Saarinen, and Kimmo Kaski. Time series prediction with multilayer perceptron, fir and elman neural networks. In *Proceedings of the World Congress on Neural Networks*, pages 491–496. Citeseer, 1996.

[KPP+19] Kapil Katyal, Katie Popek, Chris Paxton, Phil Burlina, and Gregory D Hager. Uncertainty-aware occupancy map prediction using generative networks for robot navigation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5453–5459. IEEE, 2019.

[KSA21] Parth Kothari, Brian Sifringer, and Alexandre Alahi. Interpretable Social Anchors for Human Trajectory Forecasting in Crowds. 2021.

[KSFG14] Julian F. P. Kooij, Nicolas Schneider, Fabian Flohr, and Dariu Gavrila. Context-based pedestrian path prediction. In *Proc. of ECCV*, 2014.

[KSG14] Ioannis Karamouzas, Brian Skinner, and Stephen J. Guy. Universal power law governing pedestrian interactions. *Phys. Rev. Lett.*, 113:238701, 2014.

[KSLO96] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE transactions on Robotics and Automation*, 12(4):566–580, 1996.

[KSMM+19] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, S Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *arXiv preprint arXiv:1907.03395*, 2019.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[KZBH12] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012.

[Lab16] Lowe's Innovation Labs. Lowebot. `https://pal-robotics.com/robots/reem-c/`, 2016.

[LC18] Uri Lavi and Lior Cohen. Ride vision. `https://ride.vision/`, 2018.

[LCHL07] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group behavior from video: A data-driven approach to crowd simulation. In *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation*, pages 109–118, 2007.

[LCL07] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007.

[LCV+17] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H. S. Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Li17] Yuke Li. Pedestrian path forecasting in crowd: A deep spatio-temporal perspective. In *Proc. of the ACM Conference on Multimedia*, 2017.

[Lik20] Scott Likens. The essential eight. `https://www.pwc.com/gx/en/issues/technology/essential-eight-technologies.html`, 2020.

[LJM+20] 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.

[LTH+17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[LTMR+15] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint*, abs/1504.01942, 2015.

[LWFZ16] Ming Li, Rene Westerholt, Hongchao Fan, and Alexander Zipf. Assessing spatiotemporal predictability of lbsn: A case study of three foursquare datasets. *GeoInformatica*, 22(3), 11 2016.

[Ma02] Qing Ma. Natural language processing with neural networks. In *Language engineering conference, 2002. proceedings*, pages 45–56. IEEE, 2002.

[Maj09] Barbara Majecka. Statistical models of pedestrian behaviour in the forum. Master's thesis, School of Informatics, University of Edinburgh, 2009.

[MBG+13] Marie Claude Montel, Thierry Brenac, Marie-Axelle Granié, Marine Millot, and Cécile Coquelet. Urban environments, pedestrian-friendliness and crossing decisions. 2013.

[MHB+17] David May, Kristie Holler, Cindy Bethel, Lesley Strawderman, Daniel Carruth, and John Usher. Survey of factors for the prediction of human comfort with a non-anthropomorphic robot in public spaces. *International Journal of Social Robotics*, 9, 01 2017.

[MHLK17] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M. Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[MHM+19] Christoforos Mavrogiannis, Alena Hutchinson, John Macdonald, Patrícia Alves-Oliveira, and Ross Knepper. Effects of distinct robot navigation strategies on human behavior in a crowded environment. 03 2019.

[MJS+17] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. *arXiv preprint arXiv:1705.09368*, 2017.

[MM11] Jonathan Mumm and Bilge Mutlu. Human-robot proxemics: physical and psychological distancing in human-robot interaction. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 331–338, 2011.

[MPG+10] Mehdi Moussaïd, Niriaska Perozo, Simon Garnier, Dirk Helbing, and Guy Theraulaz. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. *PloS one*, 5(4):e10047, 2010.

[MPPSD17] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163, 2017.

[Mur99] John J Murphy. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999.

[NENMC20] Baptiste Nicolet, Pierre Ecormier-Nocca, Pooran Memari, and Marie-Paule Cani. Pair correlation functions with free-form boundaries for distribution inpainting and decomposition. In *Eurographics*, 2020.

[NM19] Nishant Nikhil and Brendan Tran Morris. Convolutional neural network for trajectory prediction. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, volume 11131, pages 186–196. Springer International Publishing, 2019. Series Title: Lecture Notes in Computer Science.

[NR+00] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[ODZ+16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[ÖG12] A Cengiz Öztireli and Markus Gross. Analysis and synthesis of point distributions based on pair correlation. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[OHP+11] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, J.K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xiaoyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, , and Mita Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *Proc. of the Int.*

Conf. on Computer Vision and Pattern Recognition (CVPR), pages 3153–3160, 2011.

[OMC+13] Anne-Hélène Olivier, Antoine Marin, Armel Crétual, Alain Berthoz, and Julien Pettré. Collision avoidance between two walkers: Role-dependent strategies. *Gait & posture*, 38(4):751–756, 2013.

[OMCP12] Anne-Hélène Olivier, Antoine Marin, Armel Crétual, and Julien Pettré. Minimal predicted distance: A common metric for collision avoidance during pairwise interactions between walkers. *Gait and posture*, 36(3):399–404, 2012.

[PBB11] Ninad Pradhan, Timothy Burg, and Stan Birchfield. Robot crowd navigation using predictive position fields in the potential function framework. In *Proceedings of the 2011 American control conference*, pages 4628–4633. IEEE, 2011.

[PCBS11] Matthias Plaue, Minjie Chen, Günter Bärwolff, and Hartmut Schwandt. Trajectory extraction and density analysis of intersecting pedestrian flows from video recordings. In *Proc. of the ISPRS Conf. on Photogrammetric Image Analysis*, pages 285–296, 2011.

[PESVG09] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268. IEEE, 2009.

[PEVG10] 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.

[PGSVG12] Stefano Pellegrini, Juergen Gall, Leonid Sigal, and Luc Van Gool. Destination flow for crowd simulation. In *European Conference on Computer Vision*, pages 162–171. Springer, 2012.

[PPD07] Sébastien Paris, Julien Pettré, and Stéphane Donikian. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *Computer Graphics Forum*, volume 26, pages 665–674. Wiley Online Library, 2007.

[PPS+18] Mark Pfeiffer, Giuseppe Paolo, Hannes Sommer, Juan Nieto, Rol Siegwart, and Cesar Cadena. A data-driven model for interaction-aware pedestrian motion prediction in object cluttered environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[PS10] Anita Pal and Dayashankar Singh. Handwritten english character recognition using neural network. *International Journal of Computer Science & Communication*, 1(2):141–144, 2010.

[PS20] Pranav Padalkar and Abhay Singh. Autonomous last mile delivery market. `https://www.alliedmarketresearch.com/autonomous-last-mile-delivery-market`, 2020.

[PSS+16] Mark Pfeiffer, Ulrich Schwesinger, Hannes Sommer, Enric Galceran, and Roland Siegwart. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2096–2101. IEEE, 2016.

[RABC09] Th Robin, Gianluca Antonini, Michel Bierlaire, and Javier Cruz. Specification, estimation and validation of a pedestrian walking behavior model. *Transportation Research Part B: Methodological*, 43(1):36–56, 2009.

[RBD20] Yannick Rudolph, Ulf Brefeld, and Uwe Dick. Graph conditional variational models: Too complex for multiagent trajectories? In *"I Can't Believe It's Not Better!"NeurIPS 2020 workshop*, 2020.

[RER+18] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning*, pages 4364–4373. PMLR, 2018.

[Rie12] Laurel D. Riek. Wizard of oz studies in hri: A systematic review and new reporting guidelines. 1(1), 2012.

[Rob05] PAL Robotics. Reem (robot). `https://pal-robotics.com/robots/reem-c/`, 2005.

[Rob14] Softbank Robotics. Pepper (robot). `https://us.softbankrobotics.com/pepper`, 2014.

[RPA18] Andrey Rudenko, Luigi Palmieri, and Kai O Arras. Joint long-term prediction of human motion using a planning-based social force approach. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[RPH+20] 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.

[RSAS16] 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.

[RTS65] Herbert E Rauch, F Tung, and Charlotte T Striebel. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.

[SALK19] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. The simpler the better: Constant velocity for pedestrian motion prediction. *arXiv preprint arXiv:1903.07933*, 2019.

[SB98] Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[SG13] 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.

[SICP20] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020.

[SKD+19] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *arXiv preprint*, abs/1912.04838, 2019.

[SKG+18] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and Alexandre Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, abs/1805.07663, 2018.

[SKS+18] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. *arXiv preprint arXiv:1806.01482*, 2018.

[SLY15] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

[SMS06a] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pages 45–45. IEEE, 2006.

[SMS06b] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. 10 2006.

[SMS+14] Elias Strigel, Daniel Meissner, Florian Seeliger, Benjamin Wilking, and Klaus Dietmayer. The ko-per intersection laserscanner and video dataset. In *Proc. of the IEEE Conf. on Intelligent Transportation Systems (ITSC)*, pages 1900–1901, 2014.

[SPS+09] Armin Seyfried, Oliver Passon, Bernhard Steffen, Maik Boltes, Tobias Rupprecht, and Wolfram Klingsch. New insights into pedestrian flow through bottlenecks. *Transportation Science*, 43(3):395–406, 2009.

[Sta19] Statista. By 2030, one in 10 vehicles will be self-driving globally. `https://www.statista.com/press/p/autonomous_cars_2020/`, 2019.

[SZ09] Oliver Scherf and Stephan Zecha. Method for determining a probable movement area/location area of a living being and vehicle for carrying out said method. *Patent no. WO2*, 9:A3, 2009.

[SZDZ17] Hang Su, Jun Zhu, Yinpeng Dong, and Bo Zhang. Forecast the plausible paths in crowd scenes. In *Proc. of the Int. Joint Conference on Artificial Intelligence (IJCAI)*, pages 2772–2778. AAAI Press, 2017.

[TCP06] Adrien Treuille, Seth Cooper, and Zoran Popović. Continuum crowds. *ACM Transactions on Graphics (TOG)*, 25(3):1160–1168, 2006.

[TDLH+12] Yusuke Tamura, Phuoc Dai Le, Kentarou Hitomi, Naiwala P Chandrasiri, Takashi Bando, Atsushi Yamashita, and Hajime Asama. Development of pedestrian behavior model taking account of intention. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 382–387. IEEE, 2012.

[TK10] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803. IEEE, 2010.

[TLEH20] Jesus Tordesillas, Brett T Lopez, Michael Everett, and Jonathan P How. Faster: Fast and safe trajectory planner for flights in unknown environments. *arXiv preprint arXiv:2001.04420*, 2020.

[TLT21] 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.

[TMMK13] 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.

[TP09] Leila Takayama and Caroline Pantofaru. Influences on proxemic behaviors in human-robot interaction. IROS'09, page 5495–5502. IEEE Press, 2009.

[vdBGLM11] Jur van den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer, 2011.

[vdBLM08] Jur van den Berg, Ming C. Lin, and Dinesh Manocha. Reciprocal velocity obstacles for real-time multi-agent navigation. In *Proc. of the IEEE Int. Conf. on Robots and Automation (ICRA)*, 2008.

[vdHNWG19] Tessa van der Heiden, Naveen Shankar Nagaraja, Christian Weiss, and Efstratios Gavves. Safecritic: Collision-aware trajectory prediction. *arXiv preprint arXiv:1910.06673*, 2019.

[VK20] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[VMO17] Anirudh Vemula, Katharina Muelling, and Jean Oh. Modeling cooperative navigation in dense human crowds. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1685–1692. IEEE, 2017.

[VMO18] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA) 2018*, May 2018.

[VOS+17] Christian Vassallo, Anne-Hélène Olivier, Philippe Souères, Armel Crétual, Olivier Stasse, and Julien Pettré. How do walkers avoid a mobile robot crossing their way? *Gait & posture*, 51:97–103, 2017.

[VOS+18] Christian Vassallo, Anne-Hélène Olivier, Philippe Souères, Armel Crétual, Olivier Stasse, and Julien Pettré. How do walkers behave when crossing the way of a mobile robot that replicates human interaction rules? *Gait & posture*, 60:188–193, 2018.

[VPT16] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *arXiv preprint arXiv:1609.02612*, 2016.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[vTJG15] Wouter van Toll, Norman Jaklin, and Roland Geraerts. Towards believable crowds: A generic multi-level framework for agent navigation. In *ASCI.OPEN / ICT.OPEN (ASCI track)*, 2015.

[vTP21] Wouter van Toll and Julien Pettré. Algorithms for microscopic crowd simulation: Advancements in the 2010s. In *Computer Graphics Forum*, volume 40, 2021.

[VVS17] Pavan Vasishta, Dominique Vaufreydaz, and Anne Spalanzani. Natural vision based method for predicting pedestrian behaviour in urban environments. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[Wal06] Kenneth Wallis. A note on the calculation of entropy from histograms. 2006.

[Wal08] Johanna Wallén. *The history of the industrial robot*. Linköping University Electronic Press, 2008.

[WGD+12] Daniel Westhofen, Carolin Gründler, Konrad Doll, Ulrich Brunsmann, and Stephan Zecha. Transponder-and camera-based advanced driver assistance system. In *2012 IEEE Intelligent Vehicles Symposium*, pages 293–298. IEEE, 2012.

[WJF15] Zhan Wang, Patric Jensfelt, and John Folkesson. Modeling spatial-temporal dynamics of human movements for predicting future trajectories. In *Workshop at the Twenty-Ninth AAAI Conference on Artificial Intelligence," Knowledge, Skill, and Behavior Transfer in Autonomous Robots", AAAI Conference on Artificial Intelligence, Austin, USA, January 25, 2015*. Association for the advancement of Artificial Intelligence, 2015.

[WJGO+14] David Wolinski, S J. Guy, A-H Olivier, Ming Lin, Dinesh Manocha, and Julien Pettré. Parameter estimation and comparative evaluation of crowd simulations. In *Computer Graphics Forum*, volume 33, pages 303–312. Wiley Online Library, 2014.

[WOO16] He Wang, Jan Ondřej, and Carol O'Sullivan. Path patterns: Analyzing and comparing real and simulated crowds. In *Proc. 20th ACM SIGGRAPH Symp. Interactive 3D Graphics and Games*, pages 49–57, 2016.

[WYW+20] Lizi Wang, Hongkai Ye, Qianhao Wang, Yuman Gao, Chao Xu, and Fei Gao. Learning-based 3d occupancy prediction for autonomous navigation in occluded environments. *arXiv preprint arXiv:2011.03981*, 2020.

[WZX+16] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *arXiv preprint arXiv:1610.07584*, 2016.

[WZZH19] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. Gp-gan: Towards realistic high-resolution image blending. In *Proceedings of the 27th ACM international conference on multimedia*, pages 2487–2495, 2019.

[XHR18] Hao Xue, Du Q Huynh, and Mark Reynolds. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1194. IEEE, 2018.

[XHY+18] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018.

[XPG18] Yanyu Xu, Zhixin Piao, and Shenghua Gao. Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[XWF15] Shuang Xiao, Zhan Wang, and John Folkesson. Unsupervised robot learning to predict person motion. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June(June):691–696, 2015.

[YAJR+21] Yu Yao, Ella Atkins, Matthew Johnson-Roberson, Ram Vasudevan, and Xiaoxiao Du. Bitrap: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation. *IEEE Robotics and Automation Letters*, 2021.

[YBOB11] Kota Yamaguchi, Alexander C Berg, Luis E Ortiz, and Tamara L Berg. Who are you with and where are you going? In *Proc. of the IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1345–1352. IEEE, 2011.

[YDB17] Zhi Yan, Tom Duckett, and Nicola Bellotto. Online learning for human classification in 3d lidar-based tracking. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 864–871, September 2017.

[YK20] Ye Yuan and Kris Kitani. DLow: Diversifying Latent Flows for Diverse Human Motion Prediction. pages 1–25, 2020.

[YKS14] Xu Yan, Ioannis A Kakadiaris, and Shishir K Shah. Modeling local behavior for predicting social interactions towards human tracking. *Pattern Recognition*, 47(4):1626–1641, 2014.

[YLRO19] Dongfang Yang, Linhui Li, Keith Redmill, and Umit Ozguner. Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus. In *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, pages 899–904, 2019.

[YMPT09] Barbara Yersin, Jonathan Maïm, Julien Pettré, and Daniel Thalmann. Crowd patches: Populating large-scale virtual environments for real-time applications. In *Proc. Symp. Interactive 3D Graphics and Games*, pages 207–214, 2009.

[ZCLZ16] Jinghui Zhong, Wentong Cai, Linbo Luo, and Mingbi Zhao. Learning behavior patterns from video for agent-based crowd modeling and simulation. *Autonomous Agents and Multi-Agent Systems*, 30(5):990–1019, 2016.

[ZHCH19] Bingqing Zhang, Catherine Holloway, Tom Carlson, and R Herrera. Shared-control in wheelchairs – building interaction bridges. 05 2019.

[ZQRX19] Yanliang Zhu, Deheng Qian, Dongchun Ren, and Huaxia Xia. Starnet: Pedestrian trajectory prediction using deep neural network in star topology. *arXiv preprint arXiv:1906.01797*, 2019.

[ZSSZ18] Haosheng Zou, Hang Su, Shihong Song, and Jun Zhu. Understanding human behaviors in crowds by imitating the decision-making process. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[ZWT12] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors:learning a mixture model of dynamic pedestrian-agents. 06 2012.

[ZZP+17] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586*, 2017.

UNIVERSITÉ DE RENNES 1

Titre : Prédiction de la trajectoire du mouvement humain pour la navigation des robots

Mot clés : Prévision de trajectoires, prédiction de mouvements humains, simulation de foules, robotique, GANs

Résumé : Nos vies sont de plus en plus influencées par les robots. Ils ne se limitent plus à travailler dans les usines et apparaissent de plus en plus dans des espaces partagés avec les humains, pour livrer des biens et des colis, transporter des médicaments ou tenir compagnie à des personnes âgées. Par conséquent, ils doivent percevoir, analyser et prévoir le comportement des personnes qui les entourent et prendre des mesures sans collision et socialement acceptables des actions sans collision et socialement acceptables.

Dans cette thèse, nous abordons le problème de la prédiction de la trajectoire humaine (à court terme), afin de permettre aux robots mobiles, tels que Pepper, de naviguer dans des environnements bondés.

Nous proposons une nouvelle approche socialement consciente pour la prédiction de plusieurs piétons. Notre modèle est conçu et entraîné sur la base de réseaux adversariaux génératifs, qui apprennent la distribution multimodale des prédictions plausibles pour chaque piéton. De plus, nous utilisons une version modifiée de ce modèle pour effectuer une simulation de foule basée sur des données. La prédiction de l'emplacement des piétons occultés est un autre problème abordé dans cette thèse. Nous avons également réalisé une étude sur des jeux de données courants de trajectoires humaines. Une liste de métriques quantitatives est proposée pour évaluer la complexité de la prédiction dans ces jeux de données.

Title: Human Motion Trajectory Prediction for Robot Navigation

Keywords: Trajectory Forecasting, Human Motion Prediction, Crowd Simulation, Robotics, GANs

Abstract: Our lives are becoming increasingly influenced by robots. They are no longer limited to working in factories and increasingly appear in shared spaces with humans, to deliver goods and parcels, ferry medications, or give company to elderly people. Therefore, they need to perceive, analyze, and predict the behavior of surrounding people and take collision-free and socially-acceptable actions.

In this thesis, we address the problem of (short-term) human trajectory prediction, to enable mobile robots, such as Pepper, to navigate crowded environments.

We propose a novel socially-aware approach for prediction of multiple pedestrians. Our model is designed and trained based on Generative Adversarial Networks, which learn the multi-modal distribution of plausible predictions for each pedestrian. Additionally, we use a modified version of this model to perform data-driven crowd simulation. Predicting the location of occluded pedestrians is another problem discussed in this dissertation. Also, we carried out a study on common human trajectory datasets. A list of quantitative metrics is suggested to assess prediction complexity in those datasets.