



Doctoral Dissertation

Learning to Understand and Predict Heterogeneous
Trajectory Data

Tiago Rodrigues de Almeida
Technology

Learning to Understand and Predict Heterogeneous Trajectory Data

Örebro Studies in Technology 0



Tiago Rodrigues de Almeida

Learning to Understand and Predict
Heterogeneous Trajectory Data

© Tiago Rodrigues de Almeida, 2025

Title: Learning to Understand and Predict Heterogeneous Trajectory Data

Publisher: Örebro University, 2025
www.publications.oru.se

Printer: Printer Name

ISSN 1650-8580
ISBN 000-00-0000-000-0

Abstract

Robots and other intelligent systems operating in complex, dynamic environments must anticipate the current and future intentions, activities, and actions of surrounding agents to navigate efficiently and avoid collisions. Since the agents' motion trajectories can represent such intentions, trajectory prediction becomes critical in complex, dynamic environments. However, accurately predicting human motion remains challenging due to the multitude of environmental and agent-specific contextual factors that shape trajectory patterns. These include semantic information about the scene, agent roles or tasks, social interactions, physical constraints, and latent behavioral intentions. The complex interplay of these elements leads to heterogeneous trajectories characterized by variability in speed, direction, intent, and interaction patterns. Despite this, many state-of-the-art trajectory prediction models rely on simplifying assumptions, for instance, the absence of stopping behaviors or exclusively social navigation settings (i.e., multiple agents interacting) and training on homogeneous datasets with limited motion diversity. These limitations hinder their performance in more complex, real-world environments.

A key but underexplored source of trajectory heterogeneity lies in what we suggest referring to as trajectory classes: groupings of data samples sharing similar characteristics. These may be based on observable semantic attributes (e.g., agent type, activity, role) or data-driven latent features learned from the trajectory data itself. While observable classes can be inferred through visual perception systems, data-driven classes require learning directly from trajectory data. Both types can capture important motion diversity and enhance prediction accuracy when integrated effectively into predictors. Despite their relevance, existing work on trajectory classes lacks both dedicated datasets capturing heterogeneous motion patterns and methodological approaches addressing such heterogeneity. This thesis addresses the described gaps by systematically studying the phenomenon of heterogeneity in human motion, analyzing its sources, proposing methods to collect heterogeneous trajectory data, and incorporate trajectory classes (observable and data-driven) into trajectory prediction frameworks.

To answer the first research question – what types of datasets are needed to study trajectory classes and how they should be collected – we introduce

THÖR-MAGNI, a large-scale dataset recorded in a mock industrial environment. The dataset captures a wide range of agent activities and roles (e.g., box carriers, groups of people), which can be seen as observable classes, and provides detailed annotations for analyzing these classes in the context of human trajectory prediction. Its complexity and diversity also make it well-suited for learning and analyzing data-driven trajectory classes. We leverage THÖR-MAGNI to study the influence of trajectory classes, reflecting underlying human activities or roles in industrial contexts, on trajectory prediction.

This thesis primarily investigates both observable and data-driven trajectory classes as mechanisms to improve prediction accuracy. For observable classes, we ask: How can observable classes be leveraged to enhance trajectory prediction? We extend deep learning models to explicitly and efficiently incorporate observable classes. We evaluate their performance on THÖR-MAGNI and a state-of-the-art imbalanced outdoor dataset. Unlike previous approaches, our models do not require class-specific modules, making them inherently more scalable and memory-efficient. We also demonstrate that pattern-based approaches, such as Maps of Dynamics, outperform deep learning models in low-data and class-imbalanced regimes, which are present in robotics, particularly in cold-start settings where robots operate with minimal prior knowledge. However, observable classes can be ambiguous due to their static assignment of a single label to all trajectories of a given agent. This assumption is sometimes disregarded in real-world settings, where a single agent may perform diverse behavioral patterns. To address this limitation, we extend the THÖR-MAGNI dataset with fine-grained, frame-level action annotations, resulting in THÖR-MAGNI *Act*. By leveraging this enriched dataset, we demonstrate that frame-based action labels provide strong contextual cues. When integrated through direct conditioning or multi-task learning frameworks that jointly model trajectories and action sequences, actions help disambiguate static class assumptions and improve prediction accuracy. In particular, augmenting the state representation with frame-level action signals mitigates the limitations of static observable classes by capturing intra-agent behavioral variability.

For data-driven classes (those not directly observable), we first investigate how to learn them effectively from trajectory data in the context of prediction tasks. To this end, we propose a novel deep generative framework inspired by self-conditioning techniques from image modeling. Our Self-Conditioned generative model learns trajectory clusters that are intrinsically linked to the generative process itself, allowing these clusters to hold privileged information to guide and enhance the training of downstream predictors. Unlike traditional clustering methods, which often fail to capture minority patterns, our approach more effectively identifies less dominant classes, such as the stopping behavior, improving prediction accuracy across underrepresented trajectory modes. We further integrate these learned classes into a multi-stage prediction framework, where the trajectory classes explicitly condition generative models, leading to more accurate and probabilistically informed predictions.

In summary, this thesis provides a comprehensive investigation of the phenomenon of heterogeneity in human trajectory data. It presents methods to analyze natural motion variability, identify meaningful trajectory clusters, quantify their influence on prediction accuracy, and develop mechanisms to integrate this information into deep learning-based predictors. Together, these contributions support more accurate, robust, and context-aware prediction methods for robotics and intelligent systems operating in dynamic human environments.

Acknowledgements

This period would not have been possible without the support of some great people. I would like to express my gratitude to my supervisor, Professor Achim J. Lilienthal, for his invaluable guidance and support throughout this journey. I am also grateful to my co-supervisor, Professor Johannes Andreas Stork, for his insightful feedback and technical expertise. I would also like to thank my co-supervisor, Andrey Rudenko, for his continuous support, for being a great mentor, and for never dismissing any of my ideas, no matter how crazy they were. Andrey, I learned a lot from our discussions, and your ability to think outside the box has been a great source of inspiration for me. I hope you know that. Andrey's expertise and encouragement have been crucial in shaping my research.

I would also like to thank the other PhD students at AASS for creating such a stimulating and collaborative environment. Special thanks to Tim and Yufei. You have been wonderful collaborators, and I have learned a many things from you. Tim taught me how to be a better diplomat and person. You were also a great source of inspiration, and I am grateful for your support and friendship. Yufei taught me to be even more critical of my work, emphasizing that efficiency is essential for developing excellent products. Rishi, Suraj, and Shih-Min, it has been a pleasure getting to know you. Thanks for the discussions and beers. I hope you felt the same way and that we continue to see each other. To my other colleagues, thank you for the stimulating discussions and collaborative spirit that made this journey enjoyable. Matthias, Yuxuan, Simona, Fran, and the rest of the team, thank you for your support and for our discussions. Finally, to Per Sporrong, for always being there for anything we need.

Then, to the foundation. Edu, thank you for your support, for being the best friend, and for everything you have done for me. It's hard to believe that a PhD could offer so much. Edu is one of the best people I've ever met. But by far. He supported me not only in my work, but even more in my personal life. He always listened to my ideas, doubts, and concerns and encouraged me to keep going. I struggle to find the words to express my gratitude, but I hope you know how much you mean to me. David arrived later, yet he had such an impact on my life. With him, I learned all kinds of things I didn't know I

needed to learn, such as how often one should change a dish scrubber, how to change a bike tier in less than 30 seconds or even how to cook the best pizza. More important than that, David is a true friend who has been there for me. Edu and David, I am grateful for your support and friendship, and I'll cherish it for the rest of my life. Let's travel together again! I would also like to thank Angie and Ale. You are the best, and without you, we would have suffered a lot more. The Italian duo: Marchisio and Bertoglio. You are such characters that I can't even describe you. I laughed so much with you two! I am grateful for your support and friendship, and I hope we can continue sharing good times together.

Finally, I would like to thank my family and my portuguese people for their unwavering support and encouragement throughout this journey. *Pai, mãe, mana, Santi e Carol, a quem dedico este trabalho, obrigado por tudo. Foram fundamentais para que eu chegasse até aqui. Obrigado por me apoiarem em todos os momentos, mesmo quando as coisas estavam difíceis. São realmente especiais e sou-vos muito grato por fazerem parte da minha vida. Dizem que a família não se pode escolher, mas se pudesse, eu teria escolhido cada um de vocês por tudo o que representam para mim. Vocês são a minha base, o meu apoio e a minha inspiração para continuar a querer sempre mais e melhor. Carol, obrigado por seres a minha companheira de vida. A tua presença e apoio foram fundamentais para que eu pudesse concluir esta etapa. Eu sei que não foi fácil, mas foi uma aprendizagem para os dois. Espero que possamos continuar a crescer juntos e a enfrentar novos desafios, que, com certeza, virão, mas que serão mais fáceis de resolver, porque juntos conseguimos tudo. Aos meus amigos de sempre – J, Laura, Telma, Carlota, Gi, Zé, Cancela, Catarino, Mealha, Afonso e Lopes – agradeço o apoio e o facto de terem crescido comigo. Curado, Diogo, Miguel, António, Bernardo e Tavares obrigado também pelo apoio e por serem sempre uma fonte de inspiração. Todos vocês são uma parte importante da minha vida, e espero que possamos continuar a partilhar a vida juntos. A todos os outros amigos que me apoiaram ao longo deste percurso, o meu agradecimento. Desculpas a todos pela ausência de cinco anos mas estou de volta.*

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Notation and Terminology | 4 |
| 1.2 | Problem Formalization | 8 |
| 1.2.1 | Trajectory States Representation | 8 |
| 1.2.2 | Unimodal and Multimodal Trajectory Prediction | 10 |
| 1.2.3 | Performance Metrics | 11 |
| 1.3 | Research Questions and Contributions | 12 |
| 1.4 | Publications | 20 |
| 1.5 | Ethical Considerations | 23 |
| 1.6 | Thesis Outline | 24 |
| 2 | Literature Review | 27 |
| 2.1 | Heterogeneous Motion Trajectory Datasets | 28 |
| 2.1.1 | Outdoor Trajectory Datasets | 29 |
| 2.1.2 | Indoor Human Trajectory Datasets | 30 |
| 2.1.3 | Learning Trajectory Data Classes | 33 |
| 2.2 | Predicting Heterogeneous Trajectory Data | 33 |
| 2.2.1 | Observable Context for Trajectory Prediction | 34 |
| 2.2.2 | Data-driven Context for Trajectory Prediction | 35 |
| 3 | THÖR-MAGNI Data Collection | 39 |
| 3.1 | Introduction | 40 |
| 3.1.1 | Motivation and Contributions | 40 |
| 3.1.2 | Outline | 42 |
| 3.2 | THÖR-MAGNI Dataset | 42 |
| 3.2.1 | Environment Design and System Setup | 43 |
| 3.2.2 | Scenarios Design | 43 |
| 3.2.3 | Participants Background and Priming | 49 |
| 3.3 | Development Tools | 50 |
| 3.3.1 | Data Visualization | 50 |
| 3.3.2 | Data Filtering and Preprocessing with <i>thor-magni-tools</i> | 51 |
| 3.4 | Results and Analysis of Trajectory Data | 51 |

| | | |
|----------|--|-----------|
| 3.4.1 | Metrics | 52 |
| 3.4.2 | Comparison with State-of-the-Art Datasets | 54 |
| 3.5 | Conclusion | 56 |
| 4 | Prediction with Observable Classes | 59 |
| 4.1 | Introduction | 60 |
| 4.1.1 | Motivation and Contributions | 60 |
| 4.1.2 | Outline | 61 |
| 4.2 | Problem Formalization and Notation | 61 |
| 4.3 | Deep Learning Methods | 62 |
| 4.3.1 | Single-Output Methods: LSTMs and Transformers | 63 |
| 4.3.2 | Multiple-Output Methods: GANs and VAEs | 63 |
| 4.4 | Experiments | 65 |
| 4.4.1 | Datasets and Baselines | 66 |
| 4.4.2 | Implementation and Evaluation Details | 70 |
| 4.5 | Results | 71 |
| 4.5.1 | Accuracy Analysis Conditioned on Class Balance | 71 |
| 4.5.2 | Data Efficiency Analysis | 71 |
| 4.5.3 | Qualitative Results | 73 |
| 4.5.4 | Pitfalls of Observable Classes | 74 |
| 4.6 | Conclusions and Outlook | 76 |
| 5 | Fine-grained Human Actions | 87 |
| 5.1 | Introduction | 88 |
| 5.1.1 | Motivation and Contributions | 88 |
| 5.1.2 | Outline | 89 |
| 5.2 | THÖR-MAGNI <i>Act</i> Dataset | 90 |
| 5.2.1 | Action Annotations | 90 |
| 5.2.2 | Dataset Statistics | 92 |
| 5.3 | Trajectory and Action Prediction | 94 |
| 5.3.1 | Action-conditioned Trajectory Prediction | 94 |
| 5.3.2 | Trajectory and Action Prediction | 94 |
| 5.4 | Experiments | 95 |
| 5.4.1 | Target Scenarios | 95 |
| 5.4.2 | Evaluation Setup | 96 |
| 5.5 | Results | 96 |
| 5.5.1 | Quantitative Results | 96 |
| 5.5.2 | Qualitative Results | 98 |
| 5.6 | Conclusions and Outlook | 98 |

| | |
|--|------------|
| 6 Learning Data-driven Classes | 101 |
| 6.1 Introduction | 102 |
| 6.1.1 Motivation and Contributions | 102 |
| 6.1.2 Outline | 105 |
| 6.2 Clustering Trajectory Data | 105 |
| 6.2.1 Input Feature Representations | 106 |
| 6.2.2 Traditional Clustering Methods versus SC GAN | 108 |
| 6.3 Prediction Conditioned on Future Insights | 109 |
| 6.3.1 Formalization | 110 |
| 6.3.2 Performance Analysis | 110 |
| 6.4 SC GAN for Learning Trajectory Classes | 118 |
| 6.4.1 Overall Model Architecture | 120 |
| 6.4.2 Future-driven SC GAN | 121 |
| 6.4.3 Full-driven SC GAN | 123 |
| 6.4.4 Experiments | 124 |
| 6.5 Conclusions and Outlook | 130 |
| 7 Prediction with data-driven classes | 135 |
| 7.1 Introduction | 136 |
| 7.1.1 Motivation and Contributions | 136 |
| 7.1.2 Outline | 138 |
| 7.2 Future-driven Clusters for Prediction | 138 |
| 7.2.1 SC GAN for Diverse Prediction | 138 |
| 7.2.2 Experiments | 139 |
| 7.2.3 Quantitative Results | 140 |
| 7.2.4 Qualitative Results | 142 |
| 7.3 Full-driven Clusters for Prediction | 143 |
| 7.3.1 SC GAN for Probabilistic Prediction | 143 |
| 7.3.2 Experiments | 145 |
| 7.3.3 Quantitative Results | 149 |
| 7.3.4 Qualitative Results | 152 |
| 7.4 Conclusions | 152 |
| 8 Conclusions | 155 |
| 8.1 Thesis Contributions and Summary | 155 |
| 8.2 Future Work | 157 |
| 8.2.1 Hybrid Observable and Data-driven Class Conditioning | 158 |
| 8.2.2 Robustness of Trajectory Classes | 160 |
| 8.2.3 Time-Granularity in Data-driven Classes | 160 |
| References | 163 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Trajectory classes overview. | 2 |
| 1.2 | Integrating trajectory classes for trajectory prediction. | 5 |
| 1.3 | 3D tensor representation of a trajectory dataset. | 7 |
| 1.4 | Trajectory prediction tasks addressed in this thesis. | 9 |
| 1.5 | Joint Trajectory and Action Prediction (TAP) task. | 9 |
| 1.6 | Trajectory normalization steps. | 10 |
| 1.7 | Summary of the contributions of this thesis. | 21 |
| 2.1 | Trajectory prediction pipeline: training and inference. | 28 |
| 3.1 | Data modalities in THÖR-MAGNI. | 41 |
| 3.2 | Shared workplace environments in our dataset. | 44 |
| 3.3 | Robot used in THÖR-MAGNI (the “DARKO” robot). | 45 |
| 3.4 | Scenario definitions in the THÖR-MAGNI dataset. | 46 |
| 3.5 | Example trajectories in Scenarios 2–3. | 48 |
| 3.6 | Summary of trajectories in Scenario 3 during a single 4-minute run. | 48 |
| 3.7 | Visualization tool provided in <i>thor-magni-tools</i> . | 52 |
| 3.8 | Filtering methods in a 4-minute recording from Scenario 1. | 53 |
| 3.9 | Example of trajectory restoration for a 4-minute run for one participant in Scenario 1. | 53 |
| 3.10 | Tracking durations comparison. | 55 |
| 3.11 | Minimal distance between people comparison. | 55 |
| 3.12 | Motion speed comparison. | 56 |
| 3.13 | Path efficiency comparison. | 57 |
| 4.1 | Single-output unconditional and conditional methods. | 63 |
| 4.2 | GAN-based models. | 65 |
| 4.3 | VAE-based models. | 66 |
| 4.4 | Agent class distribution in SDD and THÖR-MAGNI datasets. | 67 |
| 4.5 | Example of trajectory data in SDD. | 69 |
| 4.6 | Comparison of motion patterns in THÖR-MAGNI, Scenario 2. | 74 |

| | |
|--|-----|
| 4.7 Motion patterns of <i>Carrier-Bucket</i> in THÖR-MAGNI, Scenario 2. | 78 |
| 4.8 General and class-conditioned CLiFF-maps in SDD. | 79 |
| 4.9 Top-1 ADE/FDE scores in THÖR-MAGNI scenarios. | 80 |
| 4.10 Top-3 ADE/FDE scores across THÖR-MAGNI scenarios. | 81 |
| 4.11 Top-1 and Top-3 ADE/FDE scores for SDD. | 82 |
| 4.12 Prediction examples in SDD. | 82 |
| 4.13 Prediction exmaples in THÖR-MAGNI dataset. | 83 |
| 4.14 The most representative clusters centroids for carriers. | 84 |
| 4.15 The most representative clusters centroids for visitors. | 84 |
| 4.16 ADE of class-conditioned trajectory prediction methods. | 85 |
| 4.17 Model selection for class-conditioned trajectory prediction. | 86 |
| 5.1 Example of action annotations for a 4-minute recording. | 89 |
| 5.2 Observable class-actions mapping and actions distribution. | 92 |
| 5.3 Acceleration, velocity, and navigation distance per class. | 93 |
| 5.4 Action-conditioned models and multi-task learning methods. | 95 |
| 5.5 Prediction examples for the evaluated methods. | 98 |
| 6.1 Trajectory clustering framework. | 106 |
| 6.2 Top-2 principal components in THÖR-MAGNI Scenario 2. | 107 |
| 6.3 Cluster type representations. | 108 |
| 6.4 Data-driven class conditioned trajectory predictors at inference time. | 111 |
| 6.5 Synthetic dataset. | 112 |
| 6.6 Trajectory predictions for THÖRMAGNI dataset with 2D flattened displacements as clustering inputs. | 115 |
| 6.7 Full-driven clusters in the synthetic dataset and occurrence matrix. | 117 |
| 6.8 Observation-driven clusters in the synthetic dataset. | 118 |
| 6.9 Future-driven clusters in the synthetic dataset. | 119 |
| 6.10 SC GAN architecture. | 122 |
| 6.11 Trajectories in the THÖR and Argoverse datasets. | 125 |
| 6.12 Trajectories in the ETH/UCY benchmark. | 125 |
| 6.13 Trajectories and directional statistics in the HOTEL dataset. | 128 |
| 6.14 Overlapping between ground truth and generated samples from FP GAN and FP SC GAN. | 129 |
| 6.15 Resulting clusters from each clustering method. | 129 |
| 6.16 Trajectory examples for the most and the least challenging clusters. | 131 |
| 6.17 Trajectory examples for two randomly sampled clusters. | 132 |
| 7.1 Data-driven classes for trajectory prediction. | 137 |
| 7.2 Proposed framework's overview based on future-driven trajectory classes. | 140 |

| | | |
|-----|--|-----|
| 7.3 | Trajectory forecasting examples. | 143 |
| 7.4 | Multi-stage prediction system overview. | 146 |
| 7.5 | Centroid-based method to rank predictions. | 147 |
| 7.6 | Top-3 predictions in test samples. | 153 |
| 8.1 | Hybrid clustering approach combining observable and data-driven classes. | 159 |
| 8.2 | Overview of trajectory prediction with temporally decomposed data-driven labels. | 162 |

List of Tables

| | | |
|-----|--|-----|
| 2.1 | Comparison of human trajectory datasets. | 32 |
| 2.2 | Comparison of trajectory prediction methods incorporating trajectory classes. | 37 |
| 4.1 | Data summary per role. | 68 |
| 4.2 | Top-1 ADE/FDE scores in THÖR-MAGNI Scenario 2 with a 90% train ratio. | 72 |
| 4.3 | Top-1 ADE/FDE scores in SDD with a 90% train ratio. | 73 |
| 4.4 | Cluster-observed class occurrence matrix in MAGNI-S2. | 76 |
| 5.1 | Eye-tracking and trajectory data recorded per observable class. | 90 |
| 5.2 | Action-conditioned trajectory prediction results for raw positions. | 97 |
| 5.3 | Comparative multi-task learning results. | 97 |
| 6.1 | Top-1 ADE/FDE for Transformer- and LSTM-based models on the synthetic and THÖR-MAGNI datasets. | 114 |
| 6.2 | Top-1 ADE/FDE average prediction scores per cluster id in the synthetic dataset for 1 fold. | 116 |
| 6.3 | Top-1 ADE/FDE for Transformer- and LSTM-based models with statistical-based features clustering. | 116 |
| 6.4 | Top-1 ADE/FDE for Transformer- and LSTM-based models with normalized trajectory clustering. | 120 |
| 6.5 | Number of clusters found in the training set of each dataset. | 126 |
| 6.6 | Top-1 ADE/FDE metrics in the test sets for our future-driven SC GAN | 127 |
| 6.7 | Top-3 ADE and FDE (\downarrow) metrics in the HOTEL test set. | 127 |
| 7.1 | Intra-observable classes ADE/FDE metrics in the test sets. | 141 |
| 7.2 | ADE/FDE metrics for 2 clusters of the test set. | 142 |
| 7.3 | ADE/FDE metrics in the test sets. | 142 |
| 7.4 | Top-3 ADE/FDE metrics in the test sets. | 150 |
| 7.5 | Top-1 ADE/FDE metrics in the test sets. | 150 |

| | | |
|-----|---|-----|
| 7.6 | Accuracy of the predictions ranking methods in the train/test split setting. | 151 |
| 7.7 | Accuracy of the predictions ranking methods in the leave-one-dataset-out setting. | 152 |

List of Algorithms

| | | |
|---|---|-----|
| 1 | Training Process of Self-Conditioned GAN | 123 |
| 2 | Multi-stage framework for probabilistic trajectory prediction . | 148 |

Chapter 1

Introduction

The future is not set. There is no fate but what we make for ourselves.

— John Connor, *Terminator 2*

Autonomous mobile robots have emerged as essential resources across several sectors of society, demonstrating substantial impact and transformative potential. In industrial environments, these systems facilitate efficient material transport and engage in complex collaborative tasks with human operators and other robotic units. In transportation, advanced driver-assistance systems harness intelligent features to enhance safety and improve the driving experience. In domestic settings, service robots are increasingly deployed to assist with activities of daily living while also addressing psychosocial needs, particularly among elderly populations.

Two fundamental aspects of these autonomous systems are their capabilities for navigation and human interaction within dynamic, complex, and anthropocentric environments. These capabilities require the integration of advanced perception and decision-making to ensure safety and effectiveness and leverage contextual awareness. Consequently, autonomous mobile robots must adapt to dynamic conditions, account for human unpredictability, and adhere to both explicit rules and implicit social norms, all while operating under real-time constraints.

Robots operate and share space with numerous dynamic agents in anthropocentric environments whose behaviors are shaped by a complex interplay between internal and external factors. Internal factors encompass an agent's intrinsic characteristics, such as activities, intentions, goals, and preferences. External factors, conversely, arise from the environment and include obstacles, semantically meaningful regions, and the presence and behavior of other agents. Autonomous mobile robots, equipped with advanced sensing and inference capabilities, can detect these external factors and infer certain internal factors, using this information to analyze the trajectory patterns of other

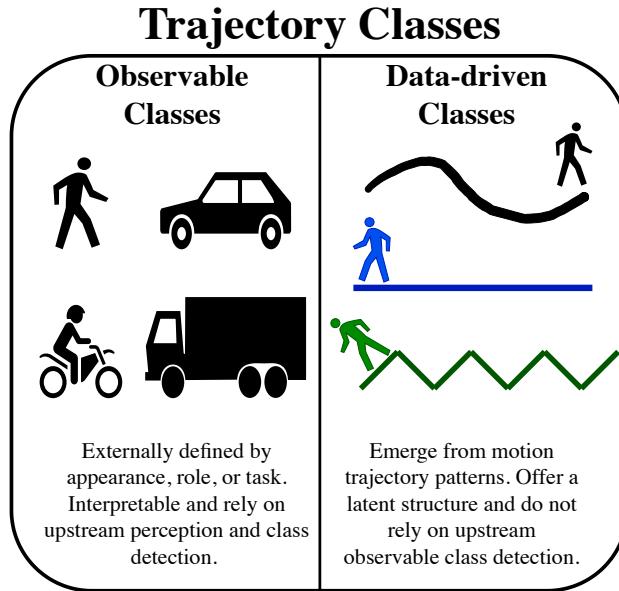


Figure 1.1: Trajectory classes can be derived from observable semantic attributes (**left**) and data-driven structures (**right**). The former can be estimated using external perception systems, while the latter can be found directly from the trajectory data.

agents. These factors influence measurable motion features, or *trajectory cues*, such as velocity, orientation, acceleration, and trajectory efficiency. Understanding and modeling the variability of these cues is crucial for anticipating future behaviors and ensuring safe and coordinated robot actions.

This thesis addresses the fundamental challenges of discovering and modeling trajectory heterogeneity, a complex phenomenon arising from the aforementioned influencing factors. Rather than treating all moving agents as homogeneous as in most prior works, this work emphasizes the importance of heterogeneity represented as *trajectory classes*, which group trajectories based on perceived appearance or trajectory cues. These classes can originate from two primary sources: *observable classes*, defined by human semantics and accessible via perception systems, and *data-driven classes*, which are automatically learned from the structure and dynamics of the trajectory data (see Fig. 1.1).

Observable classes are human-interpretable labels that can be inferred through the robot's perception stack using sensors such as cameras or lidars. Common examples include agent types like *pedestrian*, *cyclist*, or *car* in traf-

fic environments [97], or task-based roles such as *object carrier* or *visitor* in industrial contexts [86]. In some applications, such as airport environments, observable classes may include demographic categories like *child*, *elderly*, or *adult* [61]. The advantages of observable classes include interpretability, explainability, and the ability to support rule-based decisions, such as adapting a robot’s motion policy near explicitly defined vulnerable users. However, they also have limitations: semantic ambiguity (where the same class contains diverse trajectory patterns or different classes contain the same trajectory pattern), dependency on perception accuracy, and the high cost of manual annotation. It is also possible to perceive additional state information to mitigate semantic ambiguity, such as the observed per-frame actions of the agents. Actions describe the agent’s activity at each time step (e.g., picking up an object, carrying, stopping, or moving). Yet, these actions also depend on perception modules and may not always be reliable or available in all environments.

In contrast, data-driven classes offer a complementary approach by identifying structure directly from motion trajectories without relying on prior semantic labels. Data-driven classes group trajectories according to shared dynamics, such as displacement patterns, acceleration, curvature, or velocity profiles. Therefore, these classes are particularly valuable when semantic observable labels are unavailable, noisy, or fail to align with actual motion patterns. Moreover, they can capture subtle variations in behavior that may not be easily discernible through human-defined observable classes, such as the stopping behavior. Nevertheless, the main trade-off is interpretability: while such classes reflect meaningful movement patterns, they do not always correspond to intuitively understandable or semantically grounded concepts.

Understanding and predicting the dynamics of diverse moving agents is central to the safe navigation of autonomous mobile robots and effective human-robot interaction. In this thesis, we view human behavior through the lens of *trajectory patterns*, which are structured data representations that describe how agents move through space and time. By developing accurate and efficient models to classify and predict future trajectory patterns of other dynamic agents, autonomous systems can be integrated seamlessly into human-centric environments.

Studying such patterns in real-world scenarios requires access to heterogeneous trajectory datasets, which reflect the variability of human behavior and environmental complexity. To support such studies, this thesis introduces THÖR-MAGNI, a large-scale dataset comprising weakly-scripted scenarios in which humans and a mobile robot navigate, interact, and perform tasks in a mock industrial setting. To further enable action-aware prediction, we extend the dataset to include manually annotated, frame-level action labels from ego-centric video, resulting in THÖR-MAGNI *Act*. These resources allow us to study the impact of trajectory classes and augmented input states with fine-grained actions on trajectory prediction. To that end, we propose machine learning methods to both learn data-driven trajectory classes and integrate

trajectory classes, whether observable or learned, into predictive models as shown in Fig. 1.2.

Machine learning, particularly its subset deep learning, has revolutionized numerous domains in computer science and beyond. In computer vision, these techniques have enabled unprecedented advancements in object detection, recognition, and scene understanding, enhancing the capabilities of perception systems. Similarly, in natural language processing, deep learning has facilitated the development of applications such as neural machine translation, sentiment analysis, and context-aware text generation. These breakthroughs stem from the remarkable ability of deep learning architectures to automatically extract and learn hierarchical, complex representations from vast amounts of high-dimensional data. This capacity for representation learning has led to improvements in the performance and generalization of models across a broad spectrum of real-world tasks. Motivated by these successes, this thesis builds and compares deep learning-based frameworks for analyzing heterogeneous human trajectory datasets, enabling both the capture of underlying structures and the development of class-aware, accurate prediction models.

In summary, this thesis addresses a fundamental challenge in the trajectory prediction domain: modeling trajectory heterogeneity through observable and data-driven classes. It presents a comprehensive study spanning from data collection strategies designed to capture complex human motion patterns influenced by contextual factors to machine learning frameworks that infer trajectory classes directly from data. By critically revisiting the role of trajectory classes in the prior art, this work introduces novel methodologies and research directions, overcoming the limitations of prior approaches, particularly the lack of contextual cues related to the target agent and its motion trajectories. Ultimately, this thesis advances the understanding of how semantically grounded abstractions and data-driven representations can be leveraged to improve the accuracy and robustness of trajectory prediction in dynamic, human-centered environments.

1.1 Notation and Terminology

Throughout this thesis, the following general conventions for mathematical notation will be consistently applied.

- Scalars are denoted by lowercase letters a, b, c, d .
- Sets are denoted by uppercase calligraphic letters, e.g. $\mathcal{A}, \mathcal{B}, \mathcal{C}$. The cardinality of a set \mathcal{A} is denoted by $|\mathcal{A}|$. Indexed sets are concisely expressed as $\mathcal{A} = \{a_i\}_{i=1}^n$, and when the cardinality is either unspecified or irrelevant, the indexing is written as $\mathcal{A} = \{a_i\}_i$.
- Functions are denoted by letters and always shown with their arguments, e.g., $f(x)$. For families of functions, we use subscripts, e.g., f_θ , to indicate

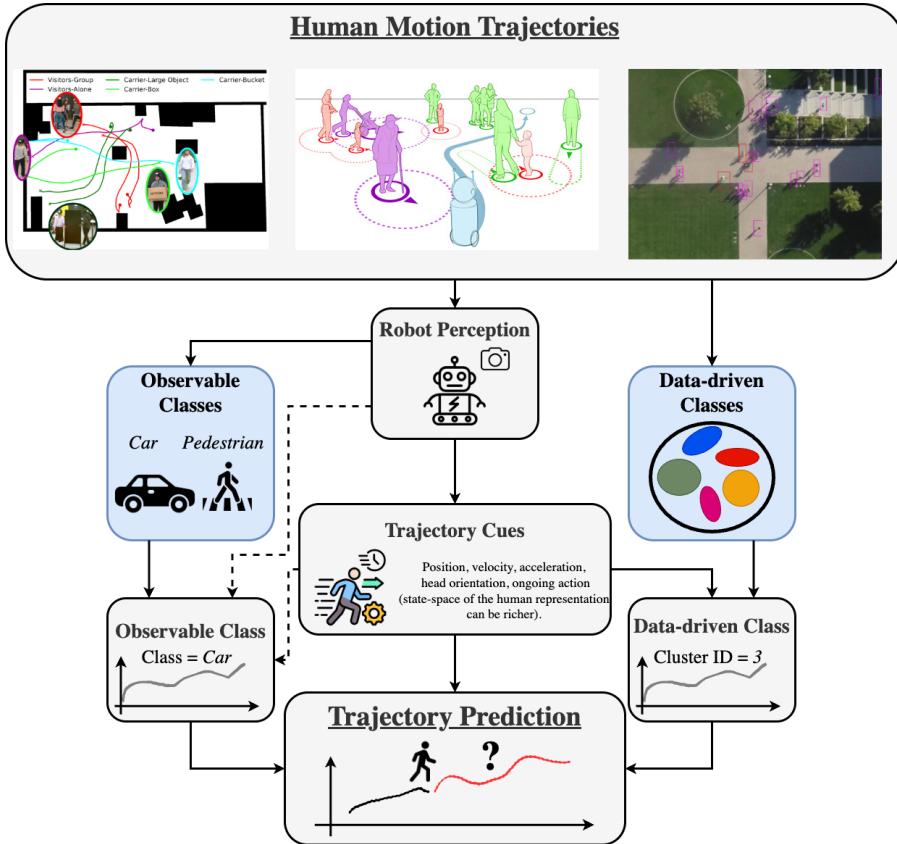


Figure 1.2: Integrating trajectory classes for trajectory prediction. **Top to bottom:** Human motion trajectories arise in diverse environments, such as road scenarios and industrial settings. Autonomous robot perception systems detect other agents and infer high-level semantic categories, resulting in observable classes (e.g., *Pedestrian*, *Car*). Alternatively or in parallel, *data-driven classes* are learned offline by clustering raw trajectory patterns based on their dynamic characteristics. Trajectory cues, such as position, velocity, acceleration, head orientation, and ongoing actions, are extracted from observed trajectories and can serve as inputs for class identification and trajectory prediction. Dashed arrows indicate observable classes detection steps, which we do not cover in this work. Integrating both observable and data-driven classes enriches the contextual understanding and improves the accuracy and robustness of future trajectory prediction.

parametrization or relationship with the subscript. We also use Newton's notation for derivatives with respect to time, where the first derivative of function f is represented by \dot{f} , while the second derivative is denoted by \ddot{f} .

- Vectors are denoted by bold lowercase letters, e.g., \mathbf{v} .
- Matrices are denoted by bold uppercase letters, e.g. \mathbf{M} .
- Matrix elements are indicated using bracket notation; for example, $[\mathbf{M}]_{i,j}$ denotes the element in the i -th row and j -th column of matrix \mathbf{M} .
- When defining vectors and matrices, their dimensions are often stated explicitly. For example, a matrix \mathbf{M} with n rows and m columns over the real numbers is denoted as $\mathbf{M} \in \mathbb{R}^{n \times m}$.
- We refer to the Frobenius or Euclidean norm of a vector \mathbf{v} as $\|\mathbf{v}\|_F$.
- We denote the prior probability of an event by $P(\cdot)$ and the probability of an event conditioned on a different event as $P(\cdot | \cdot)$.
- We denote distributions by lowercase Greek letters, e.g., ε .

In addition, the thesis uses consistent symbols for recurring concepts:

- Subscripts are frequently used to indicate relationships between different mathematical objects; for instance, $\mathbf{Y}_\mathbf{S}$ denotes a matrix \mathbf{Y} that is associated with the matrix \mathbf{S} in some way.
- We use the subscript t to indicate time steps, e.g. \mathbf{s}_t and \mathbf{s}_{t+1} .
- A 2D position at time step t of a trajectory is referred to by the bold lower case vector $\mathbf{p}_t = (x, y)$. Therefore, a 2D velocity vector at time step t of a trajectory is referred to by $\mathbf{v}_t = (\dot{x}, \dot{y})$.
- We denote latent vectors as \mathbf{z} .
- Predicted or estimated objects are denoted using $\hat{\cdot}$, e.g. $\hat{\mathbf{Y}}$.

To establish a foundation for the rest of this thesis, we now define key terminology and concepts. The term *training* or *validation trajectory dataset* refers to a collection of trajectories represented as a 3D tensor (i.e., all trajectories are of equal length), where the first axis corresponds to the number of trajectories, the second to the time steps, and the third to the state representation of a dynamic agent (see Fig. 1.3). A *trajectory* represents a dynamic agent's position profile, typically in a two-dimensional plane, over a given period. An *agent* refers to any observable dynamic object whose position is being tracked, such as humans, mobile robots, human-driven vehicles, autonomous

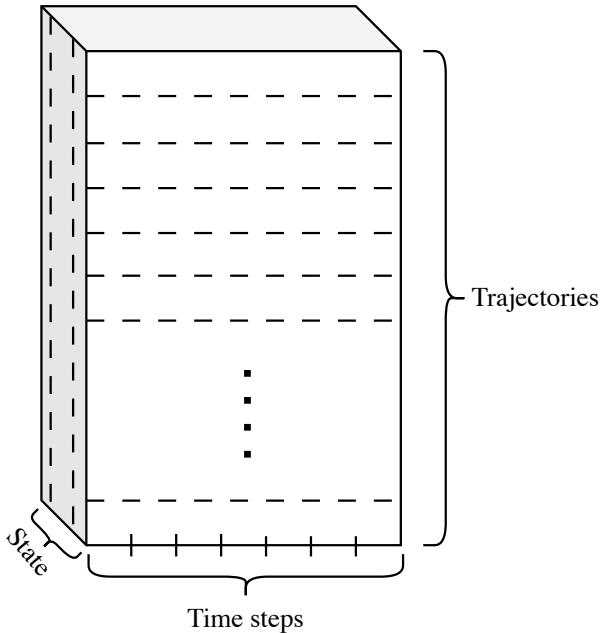


Figure 1.3: 3D tensor representation of a trajectory dataset, where the first axis indexes the individual trajectories, the second axis corresponds to the time steps, and the third axis represents the trajectory state dimensions.

vehicles, or cyclists, whose states can be computed. An agent's *states* are derived from its tracked trajectory cues, such as position and head orientation, and may include additional time-varying attributes like velocity, acceleration, or actions. Each agent may belong to an *observable class*, which characterizes the agent type (e.g., *pedestrian* or *car* in a road scenario [97]) or the agent's ongoing activities (e.g., transporting an object in an industrial setting [86]). In this case, all trajectories of an agent belong to the same class. Moreover, a trajectory may belong to a learnable class denoted by *data-driven class*, which relies on unsupervised trajectory cues processing [25]. An agent can also perform fine-grained *actions*, which may or may not be unique to its class and form part of its state as they vary over time.

Trajectory prediction or *forecasting* involves estimating future states, potentially with a different configuration from the observed states, based on past state observations and relevant contextual information, such as the locations of other agents or obstacle maps. The prediction spans a predefined *prediction horizon*, which is the period from the last observed time step to the final point in time for which predictions are made. We use the terms *observed trajec-*

tory and *tracklet* interchangeably to refer to the sequence of observed states. A tracklet spans a predefined *observation horizon*, which covers the period from the first time step to the last observed time step. *Joint trajectory and action prediction* involves predicting both future trajectories and the sequence of future actions simultaneously. *Class-conditioned trajectory prediction* or *forecasting* involves trajectory prediction conditioned on the corresponding class. *Action-conditioned trajectory prediction* or *forecasting* refers to trajectory prediction where the observed states include the observed sequence of fine-grained actions. *Multi-task trajectory and action prediction* or *forecasting* refers to the joint prediction of future trajectories and action sequences, where the observed inputs may optionally include the sequence of observed fine-grained actions.

1.2 Problem Formalization

Dynamic agents generate trajectory data, each denoted as A_i and associated with an observable class c_{A_i} . Agent trajectories are converted into tracklets of fixed-length $\mathbf{S} = (\mathbf{s}_t)_{t=1}^O$. The states \mathbf{s}_t , depending on the dataset, trajectory modeling task, and predictive model, may consist of various configurations: only 2D velocities $\mathbf{s}_t = (\dot{x}_t, \dot{y}_t)$; 2D positions and velocities, i.e., $\mathbf{s}_t = (x_t, y_t, \dot{x}_t, \dot{y}_t)$; or including the action a_t , i.e., $\mathbf{s}_t = (x_t, y_t, \dot{x}_t, \dot{y}_t, a_t)$. Action labels represent an agent's fine-grained actions at each time step from a predefined set of actions \mathcal{A} . In contrast to observable classes, which remain constant for all trajectories of an agent, action labels can vary at each time step, influencing human trajectory and capturing its heterogeneity. The *future* of an observed tracklet consists of 2D velocities, $\mathbf{Y}_{\mathbf{S}} = ((\dot{x}_t, \dot{y}_t))_{t=O+1}^{T_P}$ of length $L = T_P - O$, which are subsequently converted into future positions $\mathbf{P}_{\mathbf{S}}$. The future sequence of actions temporally aligned with $\mathbf{Y}_{\mathbf{S}}$ is denoted by $\mathbf{a}_{\mathbf{S}} = (a_t)_{t=O+1}^{T_P}$, $a_t \in \mathcal{A}$. This thesis builds on and extends these foundational notations across various trajectory modeling tasks, including:

- Canonical trajectory prediction (TP) [80].
- Class-conditioned trajectory prediction using either observable classes (O-TP) [27] or data-driven classes (D-TP).
- Action-conditioned trajectory prediction (A-TP) and joint trajectory and action prediction (TAP) [81].

Fig. 1.4 illustrates the different trajectory prediction tasks, while Fig. 1.5 depicts the joint trajectory and action prediction task.

1.2.1 Trajectory States Representation

As previously described, the agent state at time step t , denoted by \mathbf{s}_t , can be configured in various ways depending on the predictive model, the avail-

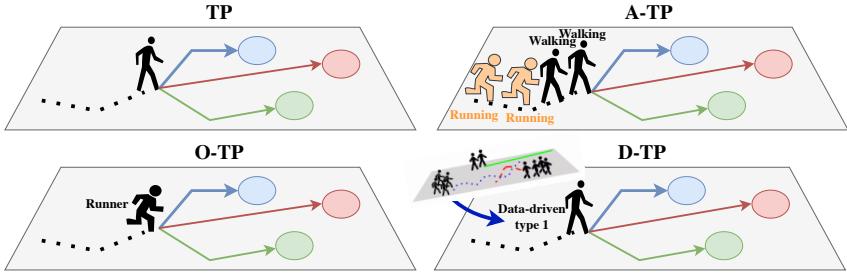


Figure 1.4: Trajectory prediction tasks, where black dots represent the observed trajectory, and colored arrows depict possible future trajectories. **Top left:** The canonical trajectory prediction task (TP) predicts the possible future(s) based solely on the observed trajectory. **Top right:** The action-conditioned trajectory prediction task (A-TP) incorporates observed actions at each time step into the input state to enhance predictions. **Bottom left:** The observable class-conditioned trajectory prediction task (O-TP) conditions the trajectory predictor on the agent’s observable class. **Bottom right:** The data-driven class-conditioned trajectory prediction task (D-TP) leverages learnable trajectory classes derived from data as conditioning inputs for more informed predictions.



Figure 1.5: Joint Trajectory and Action Prediction (TAP) task: given the observed trajectory, observable class, and fine-grained actions augmenting the observed trajectory, the task involves simultaneously predicting the future trajectory and the corresponding sequence of fine-grained actions.

able data modalities, or the application. These configurations can be broadly categorized into environment-agnostic and environment-aware representations.

Environment-agnostic features are derived solely from the agent’s trajectory and do not explicitly encode contextual information about the surrounding environment. In this thesis, we explore displacements, i.e., finite differ-

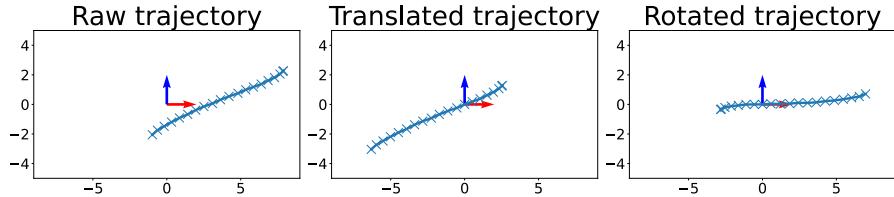


Figure 1.6: Trajectory normalization steps. **Left:** Raw trajectory. **Center:** Trajectory translated to the origin based on the pivot point at the 8th time step. **Right:** Translated trajectory rotated to align the first displacement vector with the X-axis.

ences of 2D positions, capturing local motion between successive time steps; velocities, which are the first derivatives of positions (\dot{x}_t, \dot{y}_t); and normalized trajectories, where each trajectory is translated to the origin and rotated to align with the X-axis (see Fig. 1.6).

In contrast, environment-aware features encode spatial and social context in which the agent operates, such as the absolute positions representing the agent’s spatial coordinates in a global or scene-specific coordinate frame (x_t, y_t). Alternatively, one can use semantic maps or visual inputs representing the spatial layout of the environment, which can be encoded as a 2D occupancy grid, 3D point cloud, or a semantic map. In addition, social distances or other agents’ relative positions can also be used to model the other agents in the environment. These are critical in dense or interactive scenarios, where surrounding agents shape an agent’s behavior. A typical formulation includes pairwise Euclidean distances to nearby agents or social pooling features summarizing their influence [51].

In summary, the design of s_t plays a central role in trajectory prediction models, as it drives the type and granularity of trajectory cues available for forecasting. While environment-agnostic features enable generalization across scenes and tasks, environment-aware features provide important context for interaction modeling and spatially grounded behavior. This thesis focuses primarily on environment-agnostic features while considering absolute spatial coordinates as the environment-aware representation.

1.2.2 Unimodal and Multimodal Trajectory Prediction

Trajectory prediction methods can be categorized as unimodal or multimodal, depending on how they model the inherent uncertainty of future trajectories.

Unimodal or single-output prediction methods produce a single most likely future $\hat{\mathbf{Y}}_S$ given the observed states S . These models assume that future tra-

jectories follow a single path, which may not always be the case, especially in semantically rich and dynamic environments where an agent may have multiple plausible futures. While effective in structured environments or short horizons, unimodal predictors fail to capture the stochasticity of human behavior, leading to overly smooth or averaged predictions.

Multimodal or multiple-output prediction methods address this limitation by generating multiple future trajectories $\hat{\mathbf{Y}}_{\mathbf{S}}^1, \hat{\mathbf{Y}}_{\mathbf{S}}^2, \dots, \hat{\mathbf{Y}}_{\mathbf{S}}^K$ or sample from a learned distribution $P(\mathbf{Y}_{\mathbf{S}}|\mathbf{S})$. Multimodality is typically achieved through stochastic generative models such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs). Generative frameworks model the distribution of future trajectories, allowing for the generation of diverse, plausible trajectories.

In this thesis, we investigate both single- and multiple-output predictors. Single-output approaches comprise Long Short-Term Memory (LSTM) [41] and Transformer [93] networks, and we consider them in the context of observable class- and action-conditioned trajectory prediction and multi-task learning models. We also study multiple-output frameworks, such as VAEs [48] and GANs [36], to capture the diversity of human motion patterns and support probabilistic inference for class-conditioned trajectory forecasting.

1.2.3 Performance Metrics

To evaluate trajectory predictions, we use the *Top-K Average* and *Final Displacement Errors*, as in [85, 51]. Top- K Average Displacement Error (Top- K ADE) measures the average ℓ_2 distance between the ground truth track and the closest prediction (out of K samples):

$$\text{Top-}K \text{ ADE} = \min_{k \in \{1, \dots, K\}} \frac{1}{L} \sum_{j=O+1}^{T_P} \|\mathbf{p}_j - \hat{\mathbf{p}}_j^{(k)}\|_2, \quad (1.1)$$

where \mathbf{p}_j is the ground truth position at time step j and $\hat{\mathbf{p}}_j^{(k)}$ the corresponding prediction for the k^{th} generated sample. Top- K Final Displacement Error (Top- K FDE) measures the distance between the last predicted position and the corresponding ground truth position:

$$\text{Top-}K \text{ FDE} = \min_{k \in \{1, \dots, K\}} \|\mathbf{p}_{T_P} - \hat{\mathbf{p}}_{T_P}^{(k)}\|_2, \quad (1.2)$$

where \mathbf{p}_{T_P} is the last ground truth position and $\hat{\mathbf{p}}_{T_P}$ the corresponding prediction for the k^{th} generated sample. When $K = 1$, we interchangeably refer to Top-1 ADE and Top-1 FDE as ADE and FDE, respectively.

Furthermore, we use accuracy (ACC) and F1 score (F1), both $\in [0, 1]$ for action prediction scores and trajectory classification. Accuracy represents

the proportion of correct action predictions relative to the total number of instances:

$$\text{ACC} = \frac{1}{L} \sum_{j=O+1}^{T_P} \mathbb{1}[\hat{a}_j = a_j], \quad (1.3)$$

where \hat{a}_j and a_j are the predicted and ground-truth action labels at time step j , and $\mathbb{1}[\cdot]$ is the indicator function. F1 score calculates the harmonic mean of precision and recall, providing a more balanced measure of the model's performance:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1.4)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

where TP, FP, and FN denote the number of true positives, false positives, and false negatives.

For any trajectory clustering task, we use the Davies-Bouldin Index (DBI) [24] to select the number of clusters. DBI evaluates clustering quality based on intra-cluster compactness and inter-cluster separation:

$$\text{DBI} = \frac{1}{K} \sum_{i=1}^K \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(\mu_i, \mu_j)} \right), \quad (1.5)$$

where σ_i is the average distance from points in cluster i to its centroid, and $d(\cdot, \cdot)$ is the euclidean distance between cluster centroids. The lower the average similarity is, the better the clusters are separated and the better the result of the clustering performed. To evaluate the separation between clusters, we compute the average similarity between cluster centroids. A lower value indicates better-separated clusters. Therefore, we search for the *optimal* number of clusters $\in [2, 50]$ and select the one with the lowest DBI.

1.3 Research Questions and Contributions

The central contribution of this thesis is to demonstrate how observable and data-driven classes can be effectively used to analyze and predict human trajectory data. To achieve this, we adapt and propose advanced deep learning frameworks based on state-of-the-art architectures, including Long Short-Term Memory [41], Transformer [93], Variational Autoencoder [48], and Generative Adversarial Network [36]. We run those methods in outdoor settings, road scenarios, and industrial environments. By integrating deep learning methods conditioned on trajectory classes, this thesis aims to show that these attributes are powerful cues for trajectory analysis. This overarching objective requires

developing and disseminating comprehensive, meaningful datasets of heterogeneous trajectory data, the creation of advanced trajectory predictors that incorporate both observable and data-driven class labels, and the implementation of robust evaluation and validation methodologies. These requirements collectively lead to the main research goal of this thesis:

Research Goal

How can observable and data-driven classes be effectively used to analyze and predict human trajectory data?

As discussed earlier, trajectory classes can be imposed through external perception, which may or may not be available (observable classes) or derived from trajectory data (data-driven classes). To conduct a comprehensive study of both sources, it is imperative to use datasets that contain complex, meaningful scenarios featuring heterogeneous agents with distinct trajectory patterns. In the domain of road scenarios, propelled by advancements in autonomous driving research, several datasets align with our requirements, such as the Stanford Drone Dataset (SDD) [79], Argoverse [97], and TITAN [69]. These datasets feature heterogeneous road agents (e.g., cars, pedestrians, cyclists) coexisting in shared spaces, each showing different trajectory patterns and labeled accordingly. In contrast, for industrial environments, the THÖR dataset [83] stands as the sole dataset potentially meeting our criteria. It provides accurate position tracking of heterogeneous agents in a mockup indoor industrial environment. However, THÖR's limited scope (approximately 60 minutes of trajectory data from 9 participants) constrains its applicability for training and generalizing data-intensive learning-based approaches [80]. This limitation underscores a gap in the literature: the scarcity of comprehensive, labeled, and heterogeneous trajectory datasets for industrial environments, which leads to the first research question:

Research Question 1

What datasets are needed to study heterogeneity in human trajectories, and how to collect them?

Building on the protocol established in [83], we propose the first contribution of this thesis:

Contribution 1

A comprehensive and labeled trajectory data collection of weakly scripted scenarios featuring both humans and a robot navigating and interacting within the environment called THÖR-MAGNI. A subset of this dataset is specifically designed to study the influence of observable classes on trajectory patterns. These classes encompass participants moving individually, in groups, and transporting various object types associated with distinct trajectory characteristics.

THÖR-MAGNI addresses the limitations of its predecessor (THÖR) by including over 3.5 hours of trajectory data from 40 participants. Additionally, it includes sensor data recorded by a mobile robot, eye-tracking videos, and gaze vectors aligned with the corresponding trajectories. A subset of the data collection, totaling 1.5 hours of trajectory data, is specifically designed to study the impact of observable classes on human trajectory prediction. In this portion of the dataset, participants assume two primary roles, visitors and industrial workers, while co-navigating with a mobile robot that may be static or moving in the environment. These roles are tailored to industrial tasks, such as navigating alone, moving in groups of varying sizes, and transporting different objects. This heterogeneous social setting provides a novel framework for analyzing the influence of specific industrial roles on human trajectories.

The role of observable classes in trajectory prediction is underexplored, particularly in the context of mobile robots operating in dynamic environments, as human trajectory datasets with class or activity labels remain rare, as demonstrated in the first contribution. THÖR-MAGNI now enables the study of trajectory prediction methods that incorporate observable classes, leading to the second research question:

Research Question 2

How can observable classes improve trajectory prediction?

Existing methods for trajectory prediction of heterogeneous agents, often tailored for autonomous driving, do not transfer well to robotics settings, as they depend on domain-specific contextual features [14]. Additionally, robotics applications pose unique challenges, such as the cold-start scenario, where a robot navigates previously unseen environments with limited data [23]. Moreover, both robotics and autonomous driving domains often face imbalanced data (i.e., non-uniform class distributions), which can degrade the performance of deep learning-based trajectory prediction methods [80]. Hence, it is important to understand whether class-conditioned prediction methods can offer benefits in applications with scarce or imbalanced data. Additionally, under-

standing the extent of these benefits and the conditions under which they are most effective forms the basis of the second contribution of this thesis:

Contribution 2

A comprehensive performance analysis of class-conditioned trajectory prediction methods on small and imbalanced datasets for heterogeneous agents. We propose a set of efficient deep learning baselines and evaluate their performance on both robotics and outdoor datasets (THÖR-MAGNI and the Stanford Drone Dataset) against pattern-based approaches based on Maps of Dynamics.

For this in-depth study of class-conditioned trajectory prediction methods under different conditions, we adapt several deep learning methods to include class labels. Unlike previous methods [67, 33, 72], our proposed deep learning approaches are both memory and energy-efficient, as they do not require training or running individual modules per class. We evaluate their performance across diverse training conditions, considering balanced and imbalanced datasets with uniform and non-uniform class distributions and varying amounts of training data. Moreover, we also compare the deep learning baselines against CLIFF-LHMP [107], which uses Maps of Dynamics [56] (MoD), and to its extension incorporating observable classes. Our experiments show that all methods benefit from including class labels, improving prediction accuracy in most settings. More importantly, we observe differences when learning from imbalanced datasets or in environments where sufficient data is not available. In particular, we find that deep learning methods perform better on balanced datasets. However, in applications with limited data, e.g., the cold start of a robot in a new environment or imbalanced classes, pattern-based methods may be preferable.

As demonstrated in the second contribution, observable classes provide valuable cues for enhancing trajectory prediction in autonomous driving and robotics environments. However, agents within the same class may engage in diverse activities, each influencing their trajectory patterns differently. As a result, a single agent class can encompass a wide range of motion behaviors, requiring fine-grained representations to mitigate such ambiguities. Fine-grained actions can offer more specific and detailed information about the dynamics of trajectory data. Furthermore, in addition to trajectory prediction, action prediction is crucial for ensuring safe and reliable human-robot interactions in dynamic mobile robots environments, such as those in THÖR-MAGNI. A mobile robot interacting with a human operator needs to not only predict the operator's future positions but also anticipate their future actions. Labeled frame-based actions are also a rare feature in existing human trajectory datasets, especially in dynamic environments involving mobile robots, leading to the third research question:

Research Question 3

How can frame-based actions improve trajectory prediction?

As described in the first contribution, the activity labels in THÖR-MAGNI are limited to the static roles of each participant (referred to as observable classes), representing a broad activity assigned to the participant for the duration of the experiment. To better describe activities and the associated trajectory patterns using fine-grained actions, we propose a substantial extension of THÖR-MAGNI as our third contribution:

Contribution 3

The THÖR-MAGNI *Act* dataset includes 8.3 hours of manually labeled participant actions derived from egocentric videos recorded via eye-tracking glasses. These actions, aligned with the trajectory cues provided in THÖR-MAGNI, enable the exploration of more prediction tasks, such as action-conditioned trajectory prediction and joint action and trajectory prediction.

Building on the existing observable classes in the THÖR-MAGNI dataset, THÖR-MAGNI *Act* defines a set of 14 unique action labels. Each agent class is associated with specific actions, while some actions are shared across different agent classes. Importantly, while an agent class remains constant across all trajectories of a particular agent, an action can vary at each time step. As a result, this data extension provides finer granularity in labeling internal factors, such as goal-driven actions, that can influence human trajectory. We demonstrate the utility of THÖR-MAGNI *Act* for two trajectory modeling tasks: action-conditioned trajectory prediction, and joint action and trajectory prediction. To address these tasks, we propose two deep learning-based models that outperform baseline class- and actions-unaware methods with minimal increases in the number of parameters.

While observable classes and fine-grained actions provide powerful cues for enhancing trajectory predictions, they depend on an advanced perception stack to detect these cues accurately. Also, observable classes from human annotators can be ineffective in representing similar groups of trajectories. In such cases, using observable class labels may be detrimental to the trajectory prediction task, mainly when different classes include similar trajectory patterns [27]. Alternatively, data-driven classes do not rely on upstream perception systems and are less ambiguous representations, as they are solely derived from the trajectory data itself. However, due to the inherent complexity of trajectory data, identifying meaningful groups of trajectories (clusters) that hold valuable

information for trajectory prediction presents a major challenge. Thus, we pose the fourth research question as follows:

Research Question 4

How to learn data-driven classes for trajectory prediction?

We first comprehensively study traditional clustering methods, such as K-means [66] and Time-Series K-means [91], and different input states to address the complexity of discovering meaningful data-driven classes within trajectory data for the trajectory prediction task. We find that different input states (e.g., displacements or normalized trajectories) and the part of the trajectory that is clustered (observation, future, or entire trajectories) lead to different cluster formations and, consequently, affect further trajectory predictions. Specifically, we find that clustering based on future or full trajectory states leads to more informative representations for the prediction task. Moreover, traditional clustering methods face algorithmic limitations, such as the Euclidean distance metric being unsuitable for flattened time-series data in the case of K-means or the high computational cost associated with Time-Series K-means. To overcome these limitations, we propose a novel approach that leverages a GAN to learn meaningful data-driven classes from a deep feature space and embeds this task within a trajectory generation framework. Consequently, we assess the quality of the derived clusters through their impact on prediction performance. To this end, we propose a GAN-based framework as our fourth contribution:

Contribution 4

A Self-Conditioned GAN (SC GAN) designed to produce clusters of embeddings that encapsulate similar trajectory patterns. The meaningfulness of these clusters is validated by leveraging their information to enhance the learning process of separate trajectory predictors.

Drawing from advancements in the computer vision domain, we adapt the Self-Conditioned GAN framework [64] to the trajectory generation task, enabling the learning of meaningful embeddings directly from trajectory data. GANs aim to reconstruct the generative process of the underlying data distribution through two primary neural networks: a generator and a discriminator. The generator's objective is to produce realistic samples, while the discriminator is tasked with distinguishing between real and generated samples. Self-Conditioned GAN clusters the discriminator's feature space, and the generator is conditioned on the corresponding cluster identifiers. To evaluate the quality of these clusters, we focus on the SC GAN's predictive performance conditioned on the corresponding generated clusters.

While Self-Conditioned GAN enhances the model’s exposure to trajectory diversity, it also introduces a novel research challenge: as the learned clusters are derived from future trajectories, they cannot be directly used during inference. We refer to predictors conditioned on future insights as the set of predictive approaches using data-driven classes that rely on future trajectory states. Although such predictors can not be used in practice, they contain privileged and insightful information that can be leveraged to train and improve other predictors, as demonstrated in [80]. This characteristic highlights a key opportunity for new predictive frameworks that exploit privileged information at training time but require mechanisms to infer or approximate cluster assignments at test time, which leads to the fifth research question:

Research Question 5

How can data-driven classes improve the prediction of trajectories?

For data-driven classes to be effectively integrated into trajectory prediction, these classes must either be based on the observed tracklet or inferred from future trajectory states. Using observed tracklets is a natural alternative to observable classes as they can be easily detected during inference, but they alone do not introduce novel meaningful information to the prediction task, as the trajectory predictor already processes the observed data. In contrast, future or entire trajectory-based clusters offer a forward-looking perspective. However, they can only be induced implicitly or may require a mechanism to assign clusters during inference, as the future trajectory is not available then. Focusing on the former case, SC GAN provides meaningful information, including the most representative cluster or associated cluster with the worst prediction performance. The intuition is that this information can guide better downstream predictors.

Furthermore, we also study efficient and effective mechanisms for including explicit cluster class conditioning in predictive systems instead of retraining an additional trajectory predictor from scratch based on assumptions from SC GAN’s clustering space. Hence, focusing on methods for implicit and explicit cluster class conditioning based on future or entire trajectory states, we propose the fifth contribution:

Contribution 5

Two deep learning frameworks that leverage future- and full-driven clusters for trajectory prediction. The first stands for training strategies that aim at enhancing the learning of a broader spectrum of future trajectories based on their complexity and representativeness in SC GAN’s clustering space. The second is a multi-stage probabilistic framework that conditions trajectory predictors on entire trajectory-based clusters and requires an additional mechanism to assign probabilities to the predictions during inference.

GANs often fail to model the whole space of the input data, suffering from the so-called *mode collapse* problem, where the models can only recover the data’s most representative modes (i.e., most common trajectory patterns). Since SC GAN’s clusters are based on future trajectories’ states, we have access to each cluster’s prediction errors and corresponding representativeness. Building on this information, we first introduce three training strategies based on the SC GAN’s clustering space that encourage a second GAN-based predictor to learn a broader distribution of behaviors from the input data. We penalize the generator’s loss function of a GAN-based predictor and sample more challenging samples to force the predictor to cover the most challenging modes (the ones related to the highest prediction errors). By doing so, we show that we are able to cover more modes from the input data distribution, reducing the mode collapse effect on a GAN-based forecaster. Second, we propose a multi-stage probabilistic approach for trajectory prediction, which involves clustering entire trajectory states, training cluster class-conditioned deep generative models for trajectory prediction, and ranking the corresponding predictions. To cluster entire trajectory states, we extend SC GAN to generate full trajectories, ensuring that the resulting clusters are formed based on complete trajectory information. The prediction framework is structured into three main stages:

1. **Trajectory states clustering:** We cluster the entire trajectory’s input states, comparing traditional clustering methods such as K-means [66] and its time-series extension, TS K-means [91], with our Self-Conditioned GAN, which clusters a deep feature space derived from entire trajectory states.
2. **Conditional deep generative modeling:** We train a conditional deep generative model, such as a conditional VAE (cVAE) or a conditional GAN (cGAN), conditioned on the clusters obtained from the first stage. Analogous to the fourth contribution, these clusters depend on future trajectory data, which is not available during inference.

3. **Predictions ranking:** To address the unavailability of future trajectory data during inference, we propose ranking mechanisms for the proposed future trajectories. These mechanisms assign probabilities to each prediction based on distance metrics, providing a more efficient and accurate approach than using an auxiliary neural network to map predictions to clusters, as proposed in previous works [90, 21, 45].

Our experiments show that the Self-Conditioned GAN handles distribution shifts more effectively than traditional clustering methods. Moreover, the overall system outperforms class-agnostic deep generative models and effectively captures static behaviors within the clustering space, which are often neglected by state-of-the-art trajectory prediction methods.

In conclusion, this thesis shows that classes in trajectory data, whether observable or data-driven, serve as contextual signals that, when coherent and unambiguous, offer privileged insights into the structure and semantics of human trajectory patterns. A task that can benefit from these insights is trajectory prediction, which is the main focus of this thesis. Trajectory prediction is inherently challenging due to many factors influencing the agent's behavior. However, by embracing heterogeneity rather than abstracting it away, we can develop models that are more accurate, adaptive, and context-aware. Observable classes provide interpretable cues, while data-driven classes encompass underlying motion structures beyond what is perceptible. To leverage them fully, we need rich datasets, appropriate models, and a deep understanding of trajectory patterns. This thesis contributes to all three as seen in Fig. 1.7.

1.4 Publications

The work described in this thesis has been published in peer-reviewed international conferences and journals.

- Tiago Rodrigues de Almeida, Eduardo Gutierrez Maestro and Oscar Martinez Mozos. Context-free Self-Conditioned GAN for Trajectory Forecasting. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1218-1223, 2022

I am the main author contributing to idea development, method design, software implementation, validation, results analysis, and manuscript writing. The remaining authors contributed to the manuscript preparation. This paper constitutes the main contribution described in Chapter 6 and the first part of Chapter 7.

- Tiago Rodrigues de Almeida and Oscar Martinez Mozos. Likely, Light, and Accurate Context-Free Clusters-based Trajectory Prediction. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1269-1276, 2023

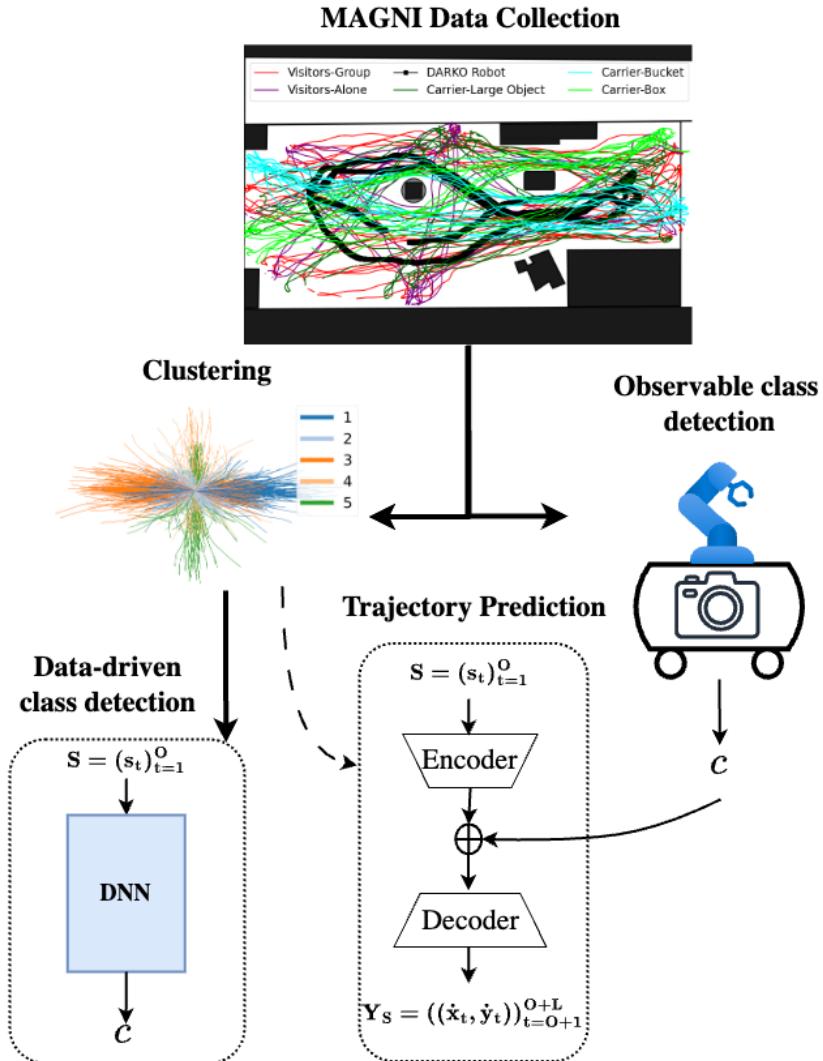


Figure 1.7: Summary of the contributions of this thesis. **Top:** THÖR-MAGNI dataset collection. **Right:** Observable classes used to condition trajectory prediction methods. **Left:** Data-driven classes extracted directly from trajectory data conditions the predictor during training and are inferred during inference. **Middle:** Class-conditioned trajectory prediction methods using an encoder-decoder architecture.

I am the main author contributing to idea development, method design, software implementation, validation, results analysis, and manuscript writing. This paper constitutes the second part of Chapter 7.

- Tiago Rodrigues de Almeida, Andrey Rudenko, Tim Schreiter, Yufei Zhu, Eduardo Gutierrez Maestro, Lucas Morillo-Mendez, Tomasz P. Kucner, Oscar Martinez Mozos, Martin Magnusson, Luigi Palmieri, Kai O. Arras, Achim J. Lilienthal. THÖR-MAGNI: Comparative Analysis of Deep Learning Models for Role-Conditioned Human Motion Prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2200-2209, 2023.

I am the main author contributing to idea development, method design, software implementation, validation, results analysis, and manuscript writing. Tim Schreiter and Eduardo Gutierrez Maestro helped with labeling the dataset and manuscript preparation. The remaining authors contributed to the manuscript preparation. This paper is part of Chapter 4.

- Tiago Rodrigues de Almeida*, Yufei Zhu*, Andrey Rudenko, Tomasz P. Kucner, Johannes A. Stork, Martin Magnusson, Achim J. Lilienthal. Trajectory Prediction for Heterogeneous Agents: A Performance Analysis on Small and Imbalanced Datasets. In *IEEE Robotics and Automation Letters*, 9(7), pages 6576-6583, 2024.

I am one of the main authors contributing to idea development, the design and implementation of deep learning methods, validation, results analysis, and manuscript writing. Yufei, the other main co-author, developed the method based on Maps of Dynamics and assisted with setting up the evaluation framework and co-authoring the manuscript. The remaining authors contributed to the manuscript preparation. This paper is the main part of Chapter 4.

- Tim Schreiter*, Tiago Rodrigues de Almeida*, Yufei Zhu, Eduardo Gutierrez Maestro, Lucas Morillo-Mendez, Andrey Rudenko, Luigi Palmieri, Tomasz P Kucner, Martin Magnusson, Achim J Lilienthal, THÖR-MAGNI: A Large-scale Indoor Motion Capture Recording of Human Movement and Robot Interaction. In *The International Journal of Robotics Research*, 44(4):568-591;2024.

I am one of the main authors contributing to idea development, the design and execution of data collection (focusing on Scenarios 1-3), validation, results analysis, and manuscript writing. Tim Schreiter, the other main co-author, made equal contributions focusing on Scenarios 4-5. Yufei Zhu, Eduardo Gutierrez Maestro, and Lucas Morillo-Mendez contributed to idea development, the design and execution of data collection, and manuscript writing. The other authors contributed to the

idea development, the design of data collection, and manuscript writing. This paper constitutes the main part of Chapter 3.

- Tiago Rodrigues de Almeida, Tim Schreiter, Andrey Rudenko, Luigi Palmieiri, Johannes A Stork, Achim J Lilienthal. THÖR-MAGNI *Act*: Actions for Human Motion Modeling in Robot-Shared Industrial Spaces. In *IEEE/ACM International Conference on Human-Robot Interaction (HRI), Short Contribution*, 2025.

I am the main author contributing to idea development, method design, software implementation, validation, results analysis, and manuscript writing. Tim Schreiter contributed by labeling the dataset and manuscript preparation, while the remaining authors supported the manuscript preparation. This paper constitutes the main part of Chapter 5.

This thesis does *not* report on the following publications, which are outside of its scope:

- Tiago Rodrigues de Almeida, Vitor Santos, Oscar Martinez Mozos, Bernardo Lourenço. Comparative Analysis of Deep Neural Networks for the Detection and Decoding of Data Matrix Landmarks in Cluttered Indoor Environments. In *Journal of Intelligent & Robotic Systems* 103, 13, 2021.
- Eduardo Gutierrez Maestro, Tiago Rodrigues de Almeida, Erik Schaffernicht, Oscar Martinez Mozos. Wearable-Based Intelligent Emotion Monitoring in Older Adults during Daily Life Activities. In *Applied Sciences*, 13(9), 5637, 2023.

1.5 Ethical Considerations

Research on trajectory prediction and modeling, particularly in dynamic and human-centered environments, entails important ethical considerations. The potential applications of this work, ranging from robotics and autonomous systems to surveillance and human behavior analysis, require a thoughtful evaluation of ethical principles to ensure responsible development and deployment.

The dataset collected during this research, THÖR-MAGNI, involves human participants whose trajectories and visual appearances are recorded and analyzed. All personally identifiable information has been excluded from the dataset used for model training to address privacy concerns. Moreover, informed consent was obtained from all participants for both data collection and its subsequent use in research, ensuring compliance with ethical standards.

Trajectory prediction systems must perform equitably across diverse demographic and environmental contexts to avoid unintended biases. Potential biases in datasets or models could disproportionately affect certain groups,

leading to concerns about fairness. During THÖR-MAGNI data collection, deliberate efforts were made to include participants of diverse genders, nationalities, and ethnicities, reducing the risk of overfitting to specific demographics or conditions (see Sec. 3.2.3).

Safety and reliability are also critical concerns, particularly in real-world applications of trajectory prediction, such as human-robot interaction. Addressing these concerns involves two key measures: (1) ensuring the interpretability of trajectory predictors to allow accountability in case of failures, and (2) integrating mechanisms to detect and manage anomalous predictions, particularly in dynamic and safety-critical environments. In line with these goals, we have developed a trajectory prediction system that assigns probabilities to predicted trajectories, enabling robots to make more informed decisions based on the likelihood of potential future states (see Sec. 6.4.3).

Finally, this thesis is grounded in the collaborative *ethos* of the scientific community, leveraging shared datasets, tools, and methods. To uphold the principles of open science, we have made our models, code, and results publicly accessible, promoting transparency, reproducibility, and further advancement in the field. This research seeks to contribute positively to the broader community by adhering to these ethical and scientific practices while ensuring its methods and outcomes align with societal values.

1.6 Thesis Outline

The rest of this thesis is structured as follows:

Chapter 2 reviews existing datasets capturing heterogeneous human motion, focusing on collections that span diverse agent types and behavioral variability. It also surveys methods for learning trajectory classes, including unsupervised and self-supervised approaches, and summarizes state-of-the-art trajectory prediction techniques for heterogeneous environments, discussing their strengths and limitations.

Chapter 3 introduces the THÖR-MAGNI dataset, a novel large-scale human trajectory dataset designed for mobile robot environments. It comprises 3.5 hours of data from 40 participants across five scenarios involving diverse spatial layouts, agent roles, and interaction patterns. Of particular interest is the annotation of human roles, which reflect activity-driven trajectory variations in industrial settings.

Chapter 4 proposes class-conditioned deep learning models for trajectory prediction and evaluates them alongside pattern-based alternatives such as Maps of Dynamics. The chapter investigates predictive performance under low-data and class-imbalanced regimes across both indoor (THÖR-MAGNI) and outdoor (SDD) settings, offering practical guidance for model selection based on deployment constraints.

Chapter 5 extends the THÖR-MAGNI dataset with frame-level action annotations derived from egocentric eye-tracking recordings, resulting in the THÖR-MAGNI *Act* dataset. These fine-grained labels capture intra-agent behavior variability at each time step and are used to enrich state representations for both action-conditioned and joint trajectory-action prediction models.

Chapter 6 investigates unsupervised data-driven trajectory classes. It first studies traditional clustering methods (e.g., K-means) under various feature representations. It then introduces the Self-Conditioned GAN, a deep learning framework that learns meaningful trajectory clusters by optimizing a clustering objective over a learned feature space. The chapter also explores predictors conditioned on future-driven clusters and evaluates their theoretical advantages.

Chapter 7 applies data-driven classes to two prediction frameworks. First, it uses trajectory clusters to enhance GAN-based predictors by encouraging coverage of rare future patterns. Second, it proposes a multi-stage probabilistic prediction system that integrates trajectory classes as conditioning inputs and uses cluster-based sampling and ranking to generate diverse and more accurate predictions.

Chapter 8 concludes the thesis by summarizing its main contributions and proposing directions for future work. These include: (1) hybrid class-conditioning strategies that combine observable and data-driven labels, (2) robustness analysis of prediction performance under class noise and sensor errors, and (3) fine-grained decomposition of data-driven classes to capture temporal dynamics of motion behavior.

Chapter 2

Literature Review

The beginning of knowledge is the discovery of something we do not understand.
— Frank Herbert, *Dune*

Trajectory prediction using deep learning methods involves a two-stage pipeline: (1) an offline training phase, where the model learns to align its predictions with ground truth by minimizing a defined objective function over a training dataset, and (2) an inference phase, where the trained model is deployed on unseen data to generate trajectory predictions based on the learned parameters. This thesis proposes using contextual observable and data-driven cues, specifically trajectory classes, to enhance the accuracy of trajectory prediction. Trajectory classes offer descriptive insights into the agent’s movement patterns, providing additional context to improve predictive accuracy.

Fig. 2.1 illustrates the trajectory prediction pipeline. During the training phase (top), including observable classes typically requires manual labeling of the trajectory dataset by a human annotator. Alternatively, data-driven classes can be identified through an automated learning process, which partitions the dataset based on inherent structures discovered within the trajectory data. Once the dataset is augmented with trajectory classes, the predictor’s weights are optimized using this enriched information. During the inference phase (bottom), the trained predictor uses the trajectory classes to produce more accurate predictions of future trajectories.

This chapter covers the existing work in every aspect of the described prediction pipeline. Sec. 2.1 begins with a comprehensive review of existing human trajectory data collections, highlighting key datasets and their respective contexts. In the same section, we discuss various clustering techniques for learning data-driven trajectory classes, which enable the augmentation of trajectory datasets with data-driven classes. Subsequently, Sec. 2.2 explores trajectory prediction approaches that explicitly or implicitly consider trajectory classes, emphasizing methods incorporating observable- and data-driven contextual information.

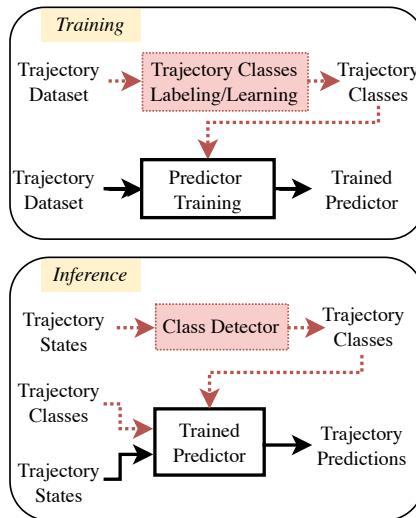


Figure 2.1: The trajectory prediction pipeline consists of two stages: offline training (**top**) and inference (**bottom**). Dotted arrows and red boxes highlight the steps for incorporating data heterogeneity. During training, either learned or predefined trajectory classes augment the original dataset and integrate the training phase of the prediction model. The model learns to map observed trajectory states and classes to future trajectory states. During inference, trajectory classes can be dynamically detected or assumed based on prior knowledge, guiding the trained model to generate accurate future trajectory predictions from the input trajectory states.

2.1 Heterogeneous Motion Trajectory Datasets

We consider a motion trajectory dataset heterogeneous when it captures distinct trajectory patterns produced by dynamic agents. In many existing datasets, particularly those designed for autonomous driving, observable classes – predefined categories such as vehicles, pedestrians, or cyclists – represent distinct trajectory patterns. Beyond these predefined classes, data-driven methods automatically identify trajectory classes, facilitating the extraction of meaningful entities from any heterogeneous dataset. This section reviews trajectory datasets that encompass heterogeneous trajectory data, with a focus on outdoor scenarios, which are widely studied, and indoor environments, where motion trajectory data collection remains comparatively underexplored.

2.1.1 Outdoor Trajectory Datasets

Outdoor datasets are primarily driven by the autonomous driving community, where trajectory prediction is a critical task [63]. The highD dataset, recorded using camera-equipped drones, captures trajectories of cars and trucks on German highways, providing a dataset of over 110,000 vehicles [53]. It lacks the diversity of other road agents, excluding, for instance, pedestrians and cyclists. Conversely, the PIE dataset focuses exclusively on pedestrian action estimation and trajectory prediction [78]. The inD dataset builds upon highD, shifting the focus from highways to intersection zones and incorporating five categories of road agents: pedestrians, bicycles, cars, trucks, and buses. The Argoverse datasets are designed for various autonomous driving tasks. The first version, Argoverse 1, features trajectories of autonomous vehicles, regular vehicles, and other road agents [17]. Argoverse 2 expands upon this, supporting a more diverse set of perception tasks, including semantic segmentation and 3D object detection [97].

In other contexts, such as university campuses and surveilled areas, human motion trajectory datasets have advanced our understanding of human movement in outdoor environments. The UCY dataset [60] consists of pedestrian trajectories in three public spaces from top-view acquisitions. Pellegrini, S. *et al.* extend this work by adding two additional outdoor scenes [76], forming the well-established ETH/UCY trajectory prediction benchmark. The Stanford Drone Dataset (SDD), recorded using a drone, provides trajectory data from eight unique scenes at the Stanford University campus [79]. It includes six observable classes: *Bicyclist*, *Pedestrian*, *Skateboarder*, *Cart*, *Car*, and *Bus*. However, the dataset’s utility is limited due to data inaccuracies and post-processing errors [2]. For instance, the observable classes *Bicyclist* and *Pedestrian* dominate the class representation in most scenes. In the remaining scenes, *Pedestrian* and *Car* are the most prevalent classes, though most cars are parked, contributing static rather than dynamic trajectories. The ViRAT dataset [74] is a comprehensive resource for event recognition in surveillance videos recorded by static cameras. It features 11 scenes with 23 event categories, grouped into three main types: single-person events (e.g., walking, running, standing), person-object interactions (e.g., entering/exiting vehicles, loading/unloading goods), and person-facility interactions (e.g., entering/exiting buildings). The UCLA dataset [89] captures aerial videos of picnic areas using drones. It offers diverse annotations, including semantic segmentation of scene layouts, human and object detection and tracking, group identification, role assignments, and event recognition. The JRDB dataset provides an extensive hybrid collection of indoor and outdoor trajectory data captured by sensors onboard a mobile robot [70]. Successive iterations of this dataset introduce additional features for understanding human movement in contextualized and social environments: JRDB-Act focuses on micro-actions and social group dynamics [30]; JRDB-Pose provides human pose estimation and tracking [94];

JRDB-Social includes text annotations describing human social interactions, capturing relationships between group body positions, salient scene contexts, venue locations, and group intentions or purposes [44].

In this thesis, we use the ETH/UCY dataset [60, 76], the SDD [79], and the Argoverse [17] to evaluate trajectory prediction methods in outdoor environments.

2.1.2 Indoor Human Trajectory Datasets

Indoor human trajectory datasets are characterized by constrained movement spaces, often influenced by static obstacles shaping how people navigate the environment. The Edinburgh dataset captures people walking through the Informatics Forum, the main building of the School of Informatics at the University of Edinburgh [68]. This dataset records pedestrian trajectories in social contexts near entrances and exit locations. The ATC dataset provides human motion trajectories in a shopping mall, collected over an extensive period of 41 days using multiple 3D range sensors [11]. The Kinetic Tracking Precision (KTP) dataset tracks individuals in an empty room using RGB-D sensors mounted on a mobile robot [73]. It includes four videos, each featuring distinct robot behaviors: static, unidirectional movement, rotation, and bidirectional movement, with interactions involving five individuals. The L-CAS dataset, recorded in the main building of the University of Lincoln, includes 3D scan frames acquired by a mobile robot platform [101]. It categorizes agents into two observable classes: *Pedestrian* and *Group*. The MoGaze dataset is the first to integrate full-body motion and gaze data, capturing a single individual performing various cleaning tasks at a table using a motion capture (mocap) system [54]. The Flobot dataset features data from an advanced autonomous floor-scrubbing robot equipped with a stereo camera, two RGB-D cameras, and a 3D lidar. This dataset spans four public locations: an airport, a warehouse, a supermarket, and a hospital, offering diverse scene perceptions. The THÖR dataset records accurate human motion trajectories in a mock industrial environment with a mocap system [83]. It includes nine participants interacting in three loosely scripted scenarios: a static mobile robot, a moving mobile robot, and the presence of obstacles. Finally, the Oxford Indoor Human Motion Dataset (Oxford-IHMM) also uses a mocap system to capture goal-oriented human trajectories in an indoor setting with one participant.

As detailed in Chapter 3, we extend the THÖR family of datasets with the introduction of THÖR-MAGNI [86]. The THÖR-MAGNI dataset offers extensive indoor human-robot interaction data using MoCap, 3D lidar, and RGB-D cameras to record motion and social interactions in various contexts. It enriches the field by including scenario-based interactions, making it ideal for analyzing human social navigation and collaboration. Our dataset explores human-robot co-navigation and robotic assistance in industrial settings, focusing on task efficiency and user experience in collaborative workflows, making THÖR-MAGNI

uniquely valuable for advancing our understanding of human-robot interaction. In comparison to the predecessor THÖR dataset, THÖR-MAGNI represents a substantial improvement, incorporating more extended data, exogenous factors such as lane markings, one-way passages, and human activities, and introducing specific HRI scenarios. In summary, the THÖR-MAGNI dataset contains 3.5 times more trajectory data than THÖR, therefore providing a broader range of situations for the analysis of human motion trajectories. In addition, THÖR-MAGNI includes sensor data recorded by a mobile robot and gaze vectors aligned with the corresponding trajectories, allowing simultaneous analysis of both modalities. Also, a key feature of THÖR-MAGNI is the inclusion of human roles, represented as observable classes, tailored to tasks and activities relevant to industrial settings (e.g., transporting objects). However, the activity labels in THÖR-MAGNI are limited to the static roles assigned to each participant, reflecting a single complex activity for the duration of the experiment. To address this limitation, we introduce THÖR-MAGNI *Act*, extending the original dataset by introducing fine-grained action labels (one action per time step), enabling a more detailed representation of sub-tasks within each activity [81].

Tab. 2.1 compares well-established and recent datasets thoroughly.

Table 2.1: Comparison of human trajectory datasets.

| Dataset | Environment | Sensors for Pose Estimation | Duration | Pose Frequency (Hz) | Pose Annotation | Social Interactions | Robot in the Scene | Intended for HRI | Goals | Map | Robot Data | Other Data |
|--|---|-----------------------------------|-----------|---------------------|-----------------|---------------------|--------------------|------------------|-------|-----|--|---|
| UCY [60] | Street (outdoor) | RGB camera | 20 min. | Continuous | Manual | ✓ | | | | | | |
| ETH [76] | University and Hotel | RGB camera | 25 min. | 2.5 | Manual | ✓ | | | ✓ | ✓ | | |
| Edinburgh [68] | Forum (outdoor) | RGB camera | 4 months | 6-10 | Automated | ✓ | | | | | | |
| Town Center [8] | Street (outdoor) | RGB camera | 5 min. | 25 | Manual | ✓ | | | | Raw | | |
| VIRAT [74] | Various outdoors | RGB camera | 29 h | 2, 5, 10 | Manual | ✓ | | | | Raw | | Human activities, agents types |
| Central station [105] | Train station | RGB camera | 34 min. | 24 | Automated | ✓ | | | | | | |
| ATC [11] | Shopping Centre | Several 3D range sensors | 41 days | 10-30 | Automatic | ✓ | | | | | | |
| NBA SportVU 2013 ¹ | Basketball court | RGB camera | 20 days | 25 | Automatic | | | | | | | Multi-agent human activities |
| KTP [73] | Empty Room | RGB-D camera | 4.7 min. | 30 | Manual | ✓ | ✓ | | | | RGB-D camera | Motion capture |
| KTH [29] | Lab | RGB-D camera and 2D laser scanner | 2.7 h | 25 | Automatic | ✓ | ✓ | | | | RGB-D camera and 2D laser scanner | |
| UCLA Aerial Event Dataset [89] | Outdoor spaces | RGB camera | 1.5 h | 60 | Automatic | ✓ | | | | Raw | | Human roles, small and large objects location |
| SDD [79] | University campus (outdoor) | RGB camera | 5 h | 30 | Manual | ✓ | | | | Raw | | Human activities |
| L-CAS [101] | Office | 3D lidar | 49 min. | 10 | Manual | ✓ | ✓ | | | | 3D lidar | Single-person, group labels |
| MoGaze [54] | Lab | Motion capture | 3 h | 120 | Ground truth | | | | | | | Human activities |
| Flobot [102] | Public spaces (i.e., airport, warehouse, supermarket) | 3D lidar and RGB-D camera | 27.5 min. | 10 | Automatic | ✓ | ✓ | | | | 2D and 3D lidars, RGB-D and stereo cameras | |
| THÖR [83] | Lab with various spatial layouts | Motion capture | 1 h | 100 | Ground truth | ✓ | ✓ | | ✓ | ✓ | 3D lidar | Aligned ET, Human activities |
| JRDB-Act [30] | University campus (indoor and outdoor) | Lidar and RGB camera | 1 h | 7.5 | Automatic | ✓ | ✓ | | | | Velodyne, several cameras (RGB and RGB-D) | Human activities |
| Oxford-IHM [32] | Lab/Office | Motion capture | 1 h | 100 | Ground truth | | ✓ | | ✓ | ✓ | RGB-D camera | Static RGB-D camera |
| THÖR-MAGNI (2024) THÖR-MAGNI Act (2025) | Lab with various spatial layouts | Motion capture | 3.5 h | 100 | Ground truth | ✓ | ✓ | ✓ | ✓ | ✓ | 3D lidar, RGB and RGB-D cameras | Aligned ET, Several human tasks and actions |

2.1.3 Learning Trajectory Data Classes

Most processes to automatically find classes in trajectory datasets rely on traditional clustering methods such as K-means [66] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [31, 12]. Among these, K-means is the most widely used clustering method for trajectory prediction due to its simplicity, scalability, and computational efficiency [13, 65, 90, 100, 98, 99, 4, 96, 5]. In trajectory prediction tasks, K-means is often employed to cluster future trajectory modalities, providing a means for trajectory decoders to produce diverse and multimodal predictions. DBSCAN, on the other hand, offers a density-based clustering approach that can automatically detect noise or outliers in the data. Unlike K-means, which assumes spherical cluster shapes, DBSCAN can capture clusters of arbitrary shapes, making it particularly suitable for more complex and irregular trajectory patterns. For instance, *CoMoGCN* leverages DBSCAN to detect groups of people in a scene, enhancing trajectory representations for more coherent predictions [22]. In another application, [58] employs DBSCAN for anomaly detection in indoor human trajectory data, where noise points correspond to irregular or anomalous movement patterns.

In addition to K-means and DBSCAN, more advanced clustering techniques have also been applied in trajectory prediction tasks. For instance, Spectral Clustering has been used to regularize the loss function while training graph-based trajectory prediction models [15]. This approach reduces the error margin in long-term prediction by transforming trajectory data into a graph representation and applying spectral methods. Furthermore, Learning Vector Quantization (LVQ) [49] has been explored to analyze the complexity of standard human trajectory datasets [42]. Unlike traditional clustering methods, LVQ adopts a prototype-based approach, mapping trajectories to representative codebook vectors. This method provides an interpretable and fine-grained analysis of the trajectory data structure while highlighting the diversity and complexity of human motion patterns. In [80], we propose a Self-Conditioned GAN framework to discover data-driven trajectory classes and extend this approach in [25]. This framework clusters the discriminator’s feature space during training, and the resulting cluster indices directly condition the generator. This process ensures that the quality of the learned clusters directly influences the prediction performance, leading to clusters that are informative to the trajectory prediction task.

2.2 Predicting Heterogeneous Trajectory Data

The prediction of heterogeneous trajectory data poses challenges due to the diversity and variability of trajectory patterns, requiring highly flexible and context-aware models. Informed prediction methods address this complexity

by incorporating additional contextual information derived from two sources: (1) the external perception and (2) the trajectory data.

External perception context pertains to observable characteristics of the agents, such as their classes (e.g., *Pedestrian* or *Car*) or characteristics (e.g., shape) and actions (e.g., picking up an object or walking), which provide prior perception-driven knowledge to guide trajectory predictions. In Sec. 2.2.1, we discuss methods that leverage the external context that is either predefined or detected dynamically.

Conversely, analyzing agents' trajectories solely depends on the trajectory data. These approaches often involve a preliminary step for detecting trajectory classes (see Fig. 2.1) or use the predictor's feature space to automatically learn meaningful data partitions. By eliminating the dependence on observable class labels, these methods enable the learning of class-like structures through unsupervised or self-supervised learning techniques. These trajectory-based methods are covered in detail in Sec. 2.2.2.

2.2.1 Observable Context for Trajectory Prediction

Observable contextual information is essential for addressing the challenges of trajectory prediction, as it allows models to be explicitly conditioned on contextual cues. In autonomous driving settings, methods like *Traphic* [14] leverage the shape of the agent obtained from perception modules, to differentiate between agents. This contextual information is subsequently used to condition and enhance trajectory predictions. However, this approach does not transfer well to human-centered scenarios where agents share similar physical appearances, rendering shape-based differentiation unreliable. *HAICU* [43] relies on class conditioning, where road agent classes are first detected, and the outputs of these detectors are used as representations for class labels. Although effective in structured environments, this approach is less suitable for human-centered settings, where trajectory cues rather than agent characteristics are the dominant source of heterogeneity [80]. Asghar *et al.* [3] explores dynamic Occupancy Grid Maps (OGMs) combined with agent classes to predict solely vehicle trajectories, assuming that class information is available. However, this method does not account for heterogeneous entities such as pedestrians and cyclists. Additionally, the resolution of OGMs presents limitations, as smaller entities like pedestrians and cyclists require finer representations, which increase computational complexity and restrict scalability. These limitations highlight the need for more versatile and scalable models applicable to diverse road users.

Alternative methods address heterogeneity by employing dedicated deep learning modules for each agent class [67, 72, 33]. *TafficPredict* [67] assumes that class information is available and that agents of the same class share similar dynamic properties, such as speed, acceleration, and reaction to other agents. A graph-based network is then used, where each class is modeled as a super node connected to its respective instances. *HEAT* is also a graph-

based deep learning framework that leverages ground truth class information in node representation for interaction and historical agent-state representation [72]. Maosi *et al.* propose a 3-channel hierarchical Transformer network, where each channel extracts trajectory features for one of three agent classes: vehicles, bicycles, and pedestrians [33]. While effective, these methods depend on class-specific nodes, encoders, or channels, which limits their scalability as the number of agent classes grows. Furthermore, such methods require an external mechanism to infer this contextual information during inference, for instance, an object detector [95]. To address these limitations, *ABC+* [38] employs supervised contrastive learning to group trajectory representations of samples belonging to the same action while pushing apart representations of different classes. This approach reduces dependency on external detectors during inference by learning observable contextual information solely during the offline training stage.

In human-centered environments, class-conditioned trajectory prediction remains underexplored. Unlike road agents, where distinct motion patterns (e.g., faster-moving vehicles versus slower-moving pedestrians) facilitate class-based differentiation, human trajectories often present subtle differences, making class-conditioned prediction more challenging. To address this gap, THÖRMAGNI [86] provides a comprehensive dataset of motion trajectories in an industrial-like environment, as described in Chapter 3. The dataset introduces agent classes tailored for industrial tasks that encapsulate groups of trajectories with distinct velocity profiles [26]. Building on this, Chapter 4 analyzes various trajectory prediction methods, including deep learning frameworks and a class-aware Maps of Dynamics method, in robotics and outdoor environments [27]. These initial efforts establish a foundation for understanding the impact of agent classes in trajectory prediction tasks within robotics and human-centered settings. In particular, we address challenges posed by class imbalance and limited training data, both critical barriers to effectively training deep learning models. Tab. 2.2 top overviews existing trajectory prediction methods where the observable contextual information guides forecasts.

2.2.2 Data-driven Context for Trajectory Prediction

As defined in Chapter 1, trajectory classes group similar trajectory samples based on shared characteristics. These groups can be identified either prior to training the predictor via clustering techniques (see Fig. 2.1) or during the training process itself. For instance, in generative models such as Variational Autoencoders (VAEs) [59], Generative Adversarial Networks (GANs) [37], or Flow-based models [20], the latent space is utilized to uncover trajectory groups by capturing the underlying distribution of the input data. Alternatively, contrastive learning techniques offer a self-supervised approach to discovering class-like representations, effectively grouping trajectories without requiring explicit labels [19]. Miao *et al.* propose a pre-processing step where the Eu-

clidean distance is used to identify the most similar observed trajectories, forming a candidate set assuming their corresponding futures are plausible [45]. A regression network then refines these candidates, while a scoring network evaluates and classifies their plausibility. This approach effectively represents the refined candidate set as a group of trajectories sharing similar future motion patterns akin to trajectory classes. *PCCSNet* identifies trajectory classes by clustering deep trajectory embeddings during the optimization process, ensuring diversity among predicted trajectories and assigning probabilities to multiple plausible futures [90]. Similarly, Chen *et al.* cluster future trajectory embeddings to learn trajectory classes [21]. A classifier then establishes a mapping between observed trajectories and the corresponding clusters of future trajectories in the latent space. Alternatively, we introduce a multi-stage prediction framework that explicitly conditions the predictor on data-driven future trajectory classes [25]. A distance-based mechanism assigns probabilities to predictions during inference, providing a more informed probabilistic output. The *Mixed Gaussian Flow* (MGF) model extends this concept by employing a Normalizing Flow framework to model probabilities over a set of plausible future trajectories [20]. This approach constrains the negative log-likelihood (NLL) to a sub-Gaussian distribution derived from clustered representations of future trajectories, improving prediction diversity and accuracy.

Contrastive learning has also been employed to force models to learn diverse motion behaviors in the embedding space [19, 96]. The *DisDis* method incorporates contrastive learning within a cVAE trajectory predictor, where the contrastive objective implicitly discriminates between similar and dissimilar motion patterns in the latent space [19]. Finally, *FEND* employs Prototypical Contrastive Learning (PCL) to construct a semantically hierarchical clustered feature space, effectively mitigating long-tailed errors and improving the representation of rare trajectory patterns [96]. Tab. 2.2 bottom overviews existing trajectory prediction methods, where the class-like representations emerge through learning without direct conditioning.

Table 2.2: Comparison of trajectory prediction methods considering observable- and data-driven contextual information. **Top:** Methods conditioned on observable characteristics, such as agent shape or type. **Bottom:** Methods leveraging data-driven contextual features, including trajectory clusters and learned trajectory embeddings.

| Method | Trajectory Class | Training | Inference |
|---------------------|-----------------------|-----------|-----------|
| Traphic [67] | agent shape | detected | detected |
| HAICU [43] | agent type | detected | detected |
| OGMs [3] | agent type | available | available |
| TrafficPredict [67] | agent type | available | available |
| HEAT [72] | agent type | available | available |
| 3-channel [33] | agent type | available | available |
| ABC+ [38] | agent type | available | |
| Ours [27] | agent type | available | available |
| PCCSNet [90] | trajectory cluster | available | detected |
| DisDis [19] | embeddings similarity | learned | |
| [45] | trajectory similarity | detected | detected |
| [21] | trajectory cluster | available | detected |
| Ours [25] | trajectory cluster | available | detected |
| FEND [96] | trajectory cluster | detected | |
| MGF [20] | trajectory cluster | available | detected |

Chapter 3

Data Collection of Heterogeneous Moving Agents

To deny our own impulses is to deny the very thing that makes us human.
— Mouse, *The Matrix*

In this chapter, we introduce THÖR-MAGNI, a large-scale indoor human and robot navigation and interaction dataset. Our dataset supports the understanding, modeling, and prediction of human motion, analyzing goal-oriented human-robot interactions, and investigating visual attention in social interaction contexts. THÖR-MAGNI addresses a critical gap in existing datasets for human motion analysis and human-robot interaction. This gap stems from the limited representation of the scene and target agent cues, which are essential for developing robust models that can accurately capture the interplay between contextual cues and human behavior across diverse scenarios. Unlike existing datasets, THÖR-MAGNI incorporates a broader range of contextual features and scenario variations, enabling controlled isolation of specific factors for analysis. The dataset includes many social human-human and HRI scenarios, rich context annotations, and multi-modal data, such as walking trajectories, gaze tracking data, and lidar and camera streams recorded from a mobile robot. We also provide tools for visualization and processing of the recorded data. THÖR-MAGNI is unique in the amount and diversity of sensor data collected in a contextualized and socially dynamic indoor environment, capturing natural human-robot interactions. For this thesis, THÖR-MAGNI introduces meaningful observable classes for dynamic agents tailored for industrial settings, enabling the study of human activities and motion patterns in complex, real-world scenarios.

3.1 Introduction

3.1.1 Motivation and Contributions

Modern approaches for modeling human motion require plentiful data recorded in diverse environments and settings to train on and for the evaluation [84]. Among the growing numbers of human trajectory datasets, most focus on capturing interactions between the moving agents in indoor [11], outdoor [79], and automated driving [10, 69, 97] settings. These datasets are designed to study how people interact and avoid collisions in social settings by describing their motion trajectories through position and velocity information. Further datasets attempt to capture full-body motion in various activities and human-object interactions in household settings [62, 54, 30].

As described in Chapter 1, human motion is influenced by many exogenous factors, which cumulatively amount to the *context* in which people move and interact. Among those are numerous external or environmental factors: motion and activities of other people and robots, locations of obstacles, semantic layout attributes such as points of common interest, direction signs, and special zones. Motion trajectory datasets should not only capture these factors to enable computational analysis of how people navigate but also vary them systematically to support factor isolation in various conditions. Datasets with rich context can better explain, model, and predict human motion.

Furthermore, beyond the spatial context, there are various aspects of the specific agent – internal factors – which are helpful in better understanding their intention, ongoing activity, attention, distraction, preferences, and abilities. These cues include head orientations, full body positions, gaze directions, social grouping, and past activity patterns. Multi-modal approaches for human motion modeling and prediction can provide more accurate results by combining these cues [27], and their development is subject to the availability of high-quality multi-modal data.

Existing datasets in human motion analysis often lack the comprehensive inclusion of the exogenous factors and the target agent cues necessary for holistic studies of human motion dynamics. This research gap hinders the development of robust models that capture the relationship between contextual cues and human behavior in different scenarios. To address this gap, we present a novel dataset incorporating a broader set of contextual features and multiple variations to support factor isolation. By integrating diverse modalities such as walking trajectories, eye-tracking data, and environmental sensory inputs captured by a mobile robot (see Fig. 3.1), our dataset fosters the exploration and analysis of human motion in various scenarios with increased fidelity and granularity. A key contribution of this dataset, particularly relevant to this thesis, is the inclusion of human roles. These roles, described by physical tasks in industrial scenarios, influence how individuals navigate the environment, pro-

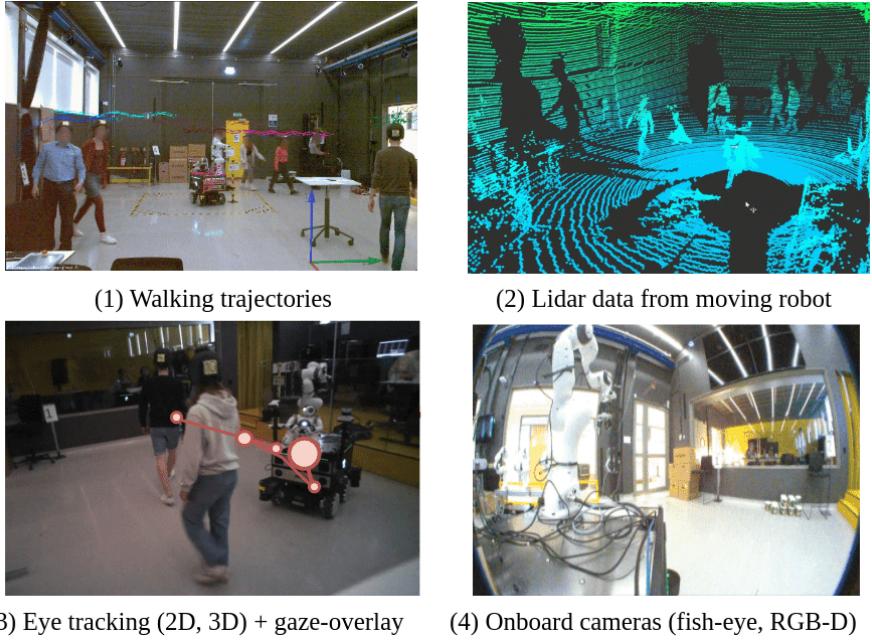


Figure 3.1: THÖR-MAGNI data modalities. (1) walking trajectories of participants in a workplace setting shared with other humans and robots; (2) lidar sweep recorded with a mobile robot; (3) snapshot from an eye tracker’s gaze overlay video; (4) fish-eye camera image from the mobile robot, showing object stashes and two goal points from our scenarios.

viding meaningful insights into the relationship between activities and motion patterns.

We propose a novel dataset, THÖR-MAGNI, of accurate human and robot navigation and interaction in diverse indoor contexts, building on the previous THÖR dataset [83]. The THÖR dataset established a foundation for collecting open-source data on human social navigation toward randomized targets in a controlled setting using motion capture technology with minimal scripting. The THÖR-MAGNI dataset represents an important advancement, enhancing data quality and features to provide rich insights into human motion and interactions within a larger room.

The THÖR-MAGNI data collection is designed around a systematic variation of exogenous factors to allow building cue-conditioned models of human motion and verifying hypotheses on factor impact. To that end, we propose five scenarios in which the participants, in addition to navigation, need to move objects, interact with each other and the robot, and respond to remote instructions. The dataset includes differential and omnidirectional robot nav-

igation, semantic zones, direction signs in the environment, and many other aspects. We provide position and head orientation for each moving agent, as well as robot sensor data and gaze tracking. Finally, we provide tools to visualize the dataset’s multiple modalities and preprocess the trajectory data. In total, THÖR-MAGNI captures 3.5 hours of motion of 40 participants over five days of recording, which is available for download¹. Furthermore, we note the continuity between the THÖR and THÖR-MAGNI recordings due to their shared environment (in diverse configurations), motion capture system, and complementary scenario composition.

3.1.2 Outline

The chapter is organized as follows: in Sec. 3.2, we provide detailed information about the data collection process, while Sec. 3.3 describes the tools available for visualizing and preprocessing the data. Finally, Sec. 3.4 presents a quantitative evaluation of the collected data, and the chapter concludes in Sec. 3.5.

3.2 THÖR-MAGNI Dataset

The THÖR-MAGNI dataset is a large-scale indoor motion capture recording of human movement and robot interaction. It consists of 52 four-minute recordings (runs) of participants performing various activities related to navigating alone and in groups, finding and transporting small and large objects, and interacting with robots. THÖR-MAGNI contains over 3.5 hours of motion data for 40 participants, including position, velocity, and head orientation across five scenarios. A total of 9.1 hours of raw eye-tracking data and 8.3 hours of egocentric video recordings were collected from 16 participants. In 24 runs, THÖR-MAGNI also includes the robot sensor data of 3D point clouds from an Ouster lidar. Additionally, videos recorded by an Azure Kinect camera and a Basler fish-eye camera onboard a mobile robot are available on request.

This thesis focuses on understanding the heterogeneity of motion trajectory patterns, which arise from factors such as the agents’ ongoing roles, activities, and motion patterns [27]. From the THÖR-MAGNI dataset, we concentrate on two scenarios where agent activities are the main feature of interest. This section provides an overview of the data recording environment (Sec. 3.2.1), summarizes the five scenarios with an emphasis on the two most relevant to this thesis and their associated agents (Sec. 3.2.2), and describes the participants’ backgrounds and priming protocols (Sec. 3.2.3).

¹<https://doi.org/10.5281/zenodo.10407223>

3.2.1 Environment Design and System Setup

The data acquisition was conducted in a laboratory at Örebro University, the same as in the THÖR dataset [83]. In THÖR-MAGNI, the laboratory features two distinct configurations:

1. **Semantically-rich industrial logistic setting:** This configuration was designed to emulate an industrial workspace to encourage frequent interactions between humans and robotic co-workers. It includes static obstacles, lane markings on the floor, and designated special zones (see Fig. 3.2, left). This layout features variations in the placement of obstacles, such as robotic manipulators and tables, to create diverse navigation paths and prevent straightforward paths between goal points.
2. **Human-robot interaction setting:** This compact, open-space configuration was optimized for human-robot interaction experiments, providing an environment to test collaboration and engagement in shared tasks (see Fig. 3.2, right).

Both configurations include seven goal positions strategically placed to encourage purposeful human navigation and promote frequent interactions in the central area (see (4) Fig. 3.2, left). Additionally, two robots were included in the environment: a static robotic arm positioned near the podium and an omnidirectional mobile robot equipped with a robotic arm and the NAO robot, particularly used for human-robot interactions, referred to as “DARKO”² (see Fig. 3.3).

We used a motion capture system from Qualisys³ with ten infrared cameras (Oqus 7+) positioned around the room to track moving agents. This setup provides broad coverage of the room’s volume, capturing data at 100 Hz with a spatial resolution of 1 mm. The system’s coordinate frame is at ground level in the center of the room. The participants and the mobile robot are represented as unique rigid bodies, identifiable through distinct patterns of passive reflective markers. These markers are arranged in six degrees of freedom (6DoF) on bicycle helmets to track the participants (see (3) in Fig. 3.2, left). For the mobile robot, the reflective markers were attached directly to its surface. This configuration precisely captures each participant’s 6DoF head position and orientation. We provided the participants with individualized helmets for the recording sessions.

3.2.2 Scenarios Design

To study the context of agent movement, we propose five distinct scenarios encompassing both human and robot dynamics. These scenarios were tailored

²<https://darko-project.eu/>

³<https://www.qualisys.com>

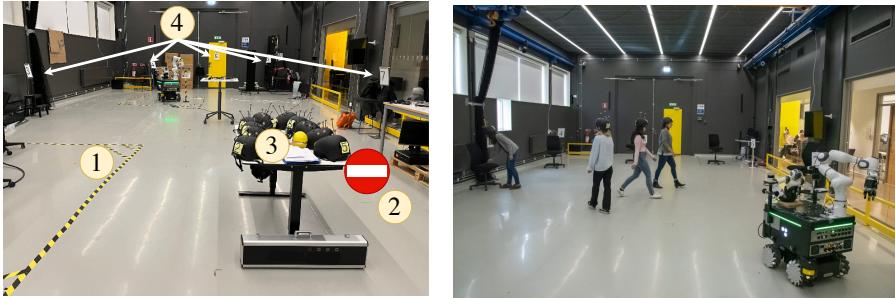


Figure 3.2: Our dataset comprehensively explores human-robot interaction in a shared workplace environment. **Left:** larger laboratory room layout highlighting several semantic features, such as static obstacles (e.g., tables and robots), floor markings (1), and a narrow corridor on the right, restricted by a no-entry sign (2). The table displays the motion-tracking helmets (3). Goal points are located around the room (4). **Right:** smaller laboratory room where participants navigate independently, collaborate in social groups, and interact with a mobile robot. Navigation between goal points is coordinated via card decks located at the goal points, which assign participants a new destination upon drawing a card, as shown on the far left.

to capture various aspects of human motion and human-robot interactions. Scenario 1 focuses on the influence of semantic environmental attributes on human motion and establishes a baseline for goal-directed social navigation. It includes two conditions: condition A, comprising a regular social behavior in a static environment without additional semantic cues, and condition B, including floor markings and a one-way passage to study their impact on navigation dynamics. Building on the layout of Scenario 1A, Scenario 2 introduces observable classes by assigning specific roles to participants. These roles represent industrial activities, such as transporting objects, and are helpful to study how role-specific tasks influence navigation patterns. Subsequently, Scenario 3 explores how the robot’s motion style affects the role-specific navigation patterns established in Scenario 2.

Transitioning to a smaller room configuration, we present two scenarios to explore human motion and intended human-robot interactions: Scenarios 4–5. In Scenario 4, participants engage in intermittent interaction with DARKO. We equip DARKO with the NAO robot (see Fig. 3.3), which communicates with participants using two distinct interaction styles: verbal (condition A) and multi-modal (condition B). These interactions are mediated to guide joint navigation toward goal points, emphasizing collaborative navigation behaviors. Scenario 5 introduces active collaboration between the robots and a human co-worker to transport small storage bins, simulating task-oriented human-robot teamwork. In summary, DARKO remains stationary in Scenarios 1–2

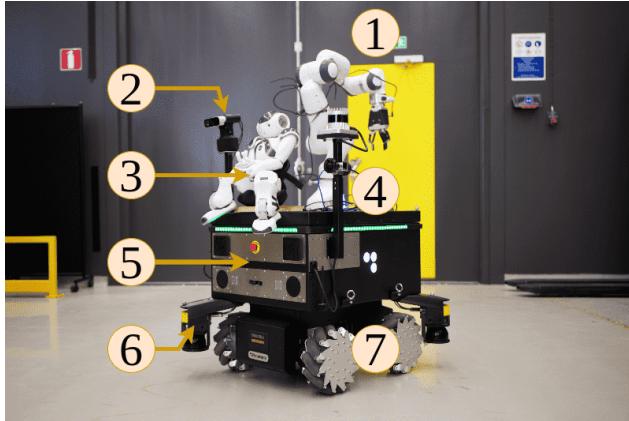


Figure 3.3: The robot used in our data collection (the “DARKO” robot) with an omnidirectional mobile base (RB-Kairos) of the dimensions $760 \times 665 \times 690$ mm (5), equipped with two sensor towers, one hosting two Azure Kinect RGB-D cameras (2), and the other hosting an Ouster OS0-128 lidar and two Basler fish-eye RGB cameras (4). Additional equipment includes two Sick MicroScan 2D safety lidars (6), mecanum wheels (7), and the NAO robot (“AR-MoD”) for interaction with participants (3). The robotic arm was not used in our experiments (1).

as an additional static element in the environment, while it is mobile in the remaining scenarios, promoting human-robot interaction and collaboration.

We conducted multiple runs for each condition across Scenarios 1–5 to capture diverse interactions and behaviors. Specifically:

- For Scenarios 1 and 3, we recorded two runs per condition.
- For Scenario 2, we recorded two runs.
- For Scenario 4, we recorded four runs per condition.
- For Scenario 5, we recorded four runs.

To mitigate learning effects, such as habituation or adaptation, we randomized the recording order of conditions for Scenarios 3–4. This approach ensures that participants do not become overly familiar with a particular condition, which can unintentionally influence their behavior. By employing this systematic methodology, we aim to capture unbiased interactions in each scenario. For a detailed overview of all scenarios and their respective attributes, refer to Fig. 3.4, with the definition of roles provided in the following subsection. Of particular relevance to this thesis are Scenarios 2–3, as they involve heterogeneous agents with distinct roles that naturally influence their trajectory cues, such as velocity, acceleration, and trajectory linearity.

| Information | Scenario 1: Capturing Motion Dynamics in the Environment | Scenario 2: Role-Specific Motion Patterns in Industrial Environments | Scenario 3: Impact of Mobile Robot Motion on Human Behavior | Scenario 4: Spatial HRI and Navigation in a Shared Environment | Scenario 5: Spatial HRI, Proactive Robotic Assistance |
|----------------------------|--|---|---|---|---|
| Roles | Visitors-Alone Visitors-Group 2 Visitors-Group 3 | Visitors-Alone Visitors-Group 2 Visitors-Group 3 Carrier-Box Carrier-Bucket Carrier-Large Object | Visitors-Alone Visitors-Group 2 Visitors-Group 3 Carrier-Box Carrier-Bucket Carrier-Large Object | Visitors-Alone Visitors-Alone HRI Visitors-Group 2 | Visitors-Alone Visitors-Group 2 Carrier-Storage Bin HRI |
| Robot-Motion | Stationary (Obstacle) | Stationary (Obstacle) | Condition based (Teleoperated) | Directional (Semi-Autonomous) | Directional (Semi-Autonomous) |
| Environment-Layout | | | | | |
| Conditions | <u>Condition A</u> Layout without <u>Condition B</u> with <u>semantics</u> | No conditions | <u>Condition A</u> <u>Condition B</u> Omnidirectional-Driving | <u>Condition A</u> Verbal-Only HRI <u>Condition B</u> Multimodal HRI | No conditions |
| Duration and Recording Day | 64 min. on Day 1-4 | 32 min. on Day 1-4 | 64 min. on Day 1-4 | 32 min. on Day 5 | 16 min. on Day 5 |

Figure 3.4: Scenario definitions in the THÖR-MAGNI dataset, including participant roles, robot motion status (e.g., autonomous or teleoperated), environment layout (i.e., obstacle maps), specific scenario conditions, as well as the duration and recording days. Each recording day involved a unique set of participants: nine on day 1 and seven on days 2–4. Three mobile eye-tracking devices were used each day for three participants. On day 5, two devices were used for two separate sets of participants.

Tasks, Activities and Roles Requiring Search and Navigation

We designed tasks to simulate realistic industrial workplace scenarios, focusing on activities requiring search, navigation, and interaction with objects, other participants, and a mobile robot. Participants perform these tasks based on their assigned *role*, representing an observable class in this thesis. Our dataset contains two types of roles: **Visitors** and **Carriers**. Visitors navigate between designated goal points either individually (*Visitors-Alone*) or in groups of two (*Visitors-Group 2*) or three (*Visitors-Group 3*). Visitors use a card-based system to navigate, receiving new destinations each time they reach a designated goal point. At each goal point, a deck of cards features instructions such as “Go to Goal 1”. The instructions specify a new destination or contain information on how to go to the robot. In the case of *Visitors-Alone*, they draw a card and

place it at the bottom of the deck. Afterward, the participant moves to the destination. In the case of groups, the members choose who draws the card.

Carriers are tasked with transporting objects of varying sizes and shapes, including small objects such as plastic buckets of canned vegetables (*Carrier-Bucket*); medium-sized objects like cardboard boxes filled with books, requiring two-handed transportation (*Carrier-Box*); and large objects represented by a poster stand with four wheels, moved collaboratively by two participants (*Carrier-Large Object*). We categorize the objects based on the navigation difficulty they impose: small objects pose the least difficulty, medium-sized objects require moderate effort, and large objects present the most substantial challenge. The overall goal of this setup is to assess how different ongoing roles affect participants' motion trajectory patterns (O-TP defined in Sec. 1.2 and solved in Chapter 4).

Scenario 2: Industrial Role-Specific Motion Patterns

Scenario 2 features the same environment layout as Scenario 1A (Fig. 3.5 right). In addition to the goal-driven navigation (“Visitors” role), this scenario introduces people performing different tasks as “Carriers”. For each run, we assign new roles to the participants. Depending on the total number of participants in a run, the “Visitors” role may include *Visitors-Alone*, *Visitors-Group 2*, and *Visitors-Group 3*. For the “Carriers” role, one participant carries small objects (i.e., buckets), another carries medium objects (i.e., boxes) between two goal points, and two participants move a large object (i.e., a poster stand). We use Discord⁴ to instruct one member of the two-person team responsible for moving the large object, enabling dynamic allocation of new goal points. Fig. 3.5 left depicts examples of trajectories for each role.

In summary, Scenario 2 combines role-specific tasks with goal-driven navigation, providing a versatile platform to study how different activities influence motion profiles in a shared environment.

Scenario 3: Impact of Mobile Robot Motion on Human Behavior

Scenario 3 introduces the opportunity to study the interplay between human activities and the motion style of the mobile robot. Unlike Scenarios 1–2, where the DARKO robot remains stationary, it is now mobile, enabling the exploration of changes in human motion patterns based on the robot’s driving style. This scenario includes two conditions, each characterized by a distinct robot navigation style: **condition A**, where the robot moves with a designated direction, following directional differential-drive kinematics and **Condition B** where it uses its mecanum wheels for omnidirectional movement. A human operator controls the mobile robot remotely to ensure the safety of participants throughout the experiments. Participant roles remain the same in both

⁴Free and easy-to-use communication and collaboration platform <https://discord.com>

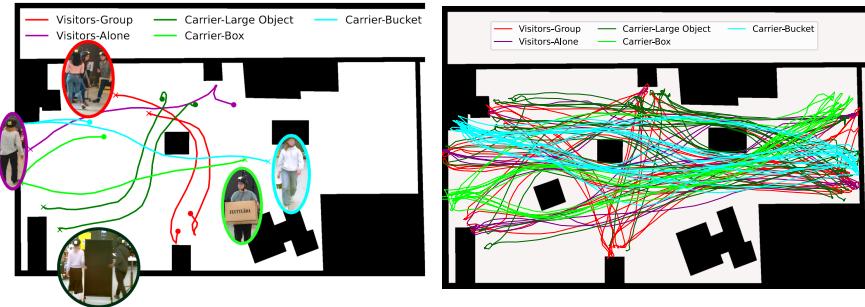


Figure 3.5: Example trajectories in the THÖR-MAGNI dataset. **Left:** participants undertake tasks according to their roles, tailored for industrial settings: visitors navigate individually and in groups between the various goals in the environment, and carriers transport boxes, buckets, and large objects. **Right:** overview of trajectories in Scenario 2 during a four-minute run. In this scenario, the mobile robot remains stationary, performing as an additional static obstacle.



Figure 3.6: Summary of trajectories in Scenario 3A (**left**) and Scenario 3B (**right**) during a single 4-minute run. Both scenarios feature the same obstacle layout but differ in the robot's navigational style: in Scenario 3A, the robot uses differential driving, while in Scenario 3B, the robot uses omnidirectional driving, yielding smoother movement. Each trajectory is color-coded according to the participant's role and the robot's identifier.

conditions as in Scenario 2 (see Fig. 3.6). Although this scenario provides valuable insights into how varying robot motion styles influence human behavior in shared environments, this thesis focuses on the study of human roles in the presence of a mobile robot.

3.2.3 Participants Background and Priming

The participants have an average age of 30.2 years ($SD = 6.7$), indicating a relatively homogenous age distribution. The dataset reflects a balanced gender representation, with 40 participants comprising 21 males and 19 females. Geographically, 23 participants are from Sweden, while 10 are from other European countries, including the Czech Republic, Spain, Germany, and Italy, ensuring diverse European representation. The remaining 7 participants come from countries in Asia, Africa, and South America, providing broader international diversity. Participants were recruited from various areas of the Örebro University campus, and their educational backgrounds varied substantially in terms of academic degrees and fields of study. At the beginning of each recording day, we provided standardized information to all participants to ensure unbiased and natural behaviors. The instructions emphasized the experiment's focus on testing the robot's perception of humans, involving tasks such as navigating the laboratory and executing physical activities, with an estimated duration of 15 minutes. Additionally, participants completed a demographic questionnaire, which we used to create diverse group compositions, aiming for optimal allocation of eye-tracking devices across different roles. For example, in groups of two or three participants, only one participant was equipped with an eye tracker, and at least one carrier was selected to wear the device.

After each run, participants completed the raw version of the NASA Task Load Index (RTLX) [40, 39]. The scale consists of a 21-point set of subscales [1 = low; 21 = high], each of which assesses the mental demand, physical demand, temporal demand, and frustration produced by the task as reported by the participant, as well as their self-perceived performance and frustration. After each session of the last run of Scenarios 3 and 5, participants complete two additional mobile robot questionnaires. First, they filled out the Godspeed Questionnaire Series [6], a semantic differential set of subscales [5-point] that measures participants' perceptions of the robot in terms of anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety. Second, they completed a 5-point Likert scale [1 = strongly disagree; 5 = strongly agree] to assess trust in the robot in industrial human-robot collaborations [18]. All questionnaires were completed on paper. Ethics approval was not required for our data collection in accordance with institutional guidelines and the Swedish Ethical Review Act (SFS number: 2003:460). Written informed consent was obtained from all participants prior to their participation. Additionally, due to the low weight of the robot and objects involved, combined with the implementation of comprehensive safety precautions, there was no risk of harming the participants.

3.3 Development Tools

Most existing datasets in the field lack a dedicated toolbox for streamlined visualization and preprocessing. Addressing this gap, we contribute a set of data visualization tools, including a dashboard, and introduce a specialized Python package named *thor-magni-tools*. This package facilitates the filtering and preprocessing of raw trajectory data, enhancing the accessibility and usability of the THÖR-MAGNI dataset. By making these resources available, we aim to provide researchers with versatile and fast means to analyze and extract valuable insights from the dataset.

3.3.1 Data Visualization

To provide researchers and users with an intuitive interface for the exploration of human movement, gaze patterns, and environmental perception of the THÖR-MAGNI dataset, we made a set of visualization tools publicly available⁵. Our visualization dashboard provides a user-friendly interface with multiple interactive components. The dashboard includes the following key features:

1. **Trajectory visualization:** Users can visualize agents' trajectories in 2D or 3D space. The trajectories are color-coded to represent different agents, allowing the user to identify patterns and variations.
2. **Velocity profiles:** The dashboard also displays velocity profiles corresponding to each trajectory, allowing users to analyze speed variations during different movement phases. This feature helps to understand the dynamics of human movement under different conditions.
3. **Eye-tracking data alignment:** Gaze data is overlaid on the 3D trajectories, providing insights into visual attention during different motion phases. Researchers can explore how gaze patterns align with specific trajectory segments, promoting the study of the cognitive processes underlying human actions.
4. **Lidar data visualization:** Lidar sensor data is presented in a 3D format to show the environmental context of human motion. This information is critical for studying lidar-based human detectors onboard mobile robots, especially in complex environments like THÖR-MAGNI.

In addition to data visualization, our dashboard contains concise scenario descriptions. Each scenario represents a unique context in which human motion data was captured (described in Sec. 3.2.2). These descriptions include information such as the physical environment, task objectives, social interactions, and specific conditions imposed on the participants (e.g., transporting

⁵<https://github.com/tmralmeida/magni-dash/tree/dash-public>

objects between two goal points). Understanding these scenarios is vital for accurately interpreting the data and ensures that researchers can contextualize their analyses effectively.

3.3.2 Data Filtering and Preprocessing with *thor-magni-tools*

To facilitate the use of the agents' trajectories in our dataset, we developed the *thor-magni-tools* Python package⁶, a tool designed specifically for filtering, preprocessing, and visualizing trajectory data. This tool focuses primarily on mitigating tracking issues arising from the motion capture system, enhancing the data quality for downstream tasks, and studying novel trajectory prediction methods. It also provides a visualization tool tailored explicitly for eye-tracking and motion trajectory data visualization (see Fig. 3.7).

To filter 3D trajectory data, we provide two methods: (1) using the most reliable marker, i.e., the marker of each helmet with the highest number of tracking locations, and (2) restoring the helmet tracking based on the average of the tracking locations of each marker. Both approaches offer a trade-off between tracking quantity and quality. The method based on the best marker produces smoother trajectories since it depends on a single marker. Conversely, the method averaging the positions of all visible markers generates longer trajectories but with increased jerkiness, as it incorporates data from multiple markers, which can vary. However, this jerkiness can be alleviated by applying a moving average filter in subsequent processing stages. Fig. 3.8 shows an example of the two methods applied on THÖR-MAGNI trajectory data.

For both 3D and 6D tracks (X, Y, Z, and 3D orientation), we provide an interpolation method based on a predefined maximum number of positions in the absence of tracking. This method is used to fill in the missing data points while maintaining the integrity of the motion patterns and ensuring continuity in the trajectories. An example of the interpolation of a trajectory based on *thor-magni-tools* is depicted in Fig. 3.9. Finally, this tool offers optional preprocessing steps, including downsampling and signal smoothing through a moving average filter, further refining the processed trajectories.

3.4 Results and Analysis of Trajectory Data

This section presents a comparison with popular human trajectory datasets, specifically the ETH/UCY benchmark and THÖR, with our THÖR-MAGNI dataset. Our analysis encompasses a multidimensional evaluation, covering various facets of the data recordings. These include trajectory continuity, social proxemics delineating interpersonal interactions, and motion trajectory cues

⁶<https://github.com/tmralmeida/thor-magni-tools>

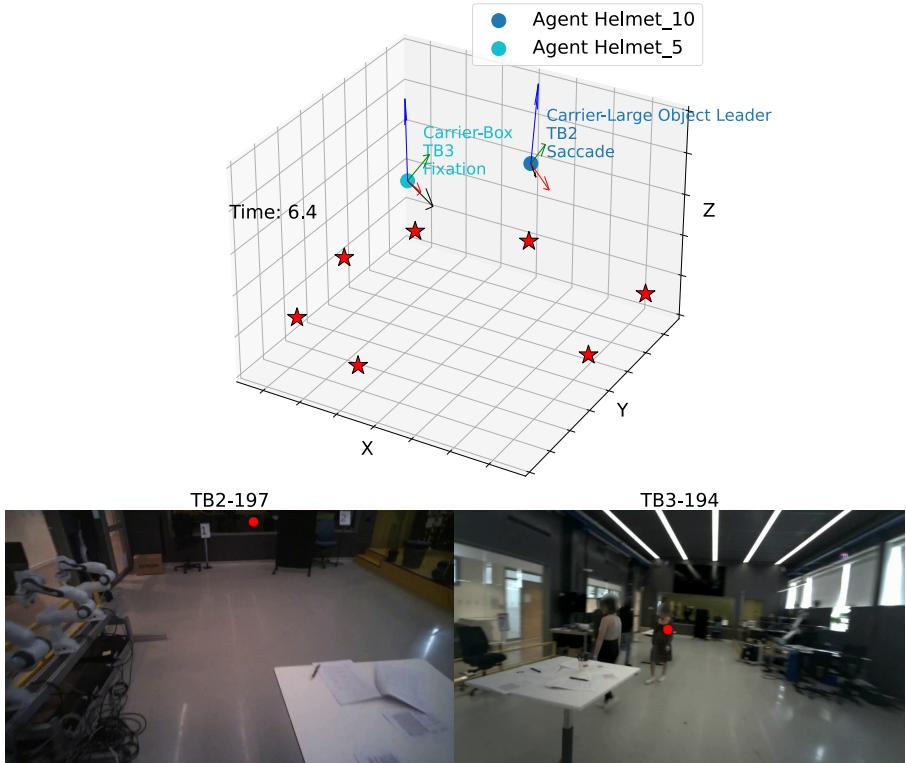


Figure 3.7: Visualization tool provided in *thor-magni-tools*. **Top:** head orientation (RGB reference frame) and 3D gaze vector (black arrow). **Bottom:** 2D gaze points (red dots) overlaid on egocentric videos.

such as velocity profiles and trajectory linearity. This comparison aims to situate THÖR-MAGNI among its predecessors, showing its potential for advancing human motion trajectory analysis and human-robot interaction research.

3.4.1 Metrics

To evaluate the trajectory data of our dataset in comparison to previous data collections, we employ metrics proposed by Rudenko *et al.* [83] and Amirian *et al.* [1]:

- **Tracking duration** (s) represents the average duration of continuous tracking for all human agents. A higher value indicates longer tracking, which is favorable for long-term human motion prediction methods.

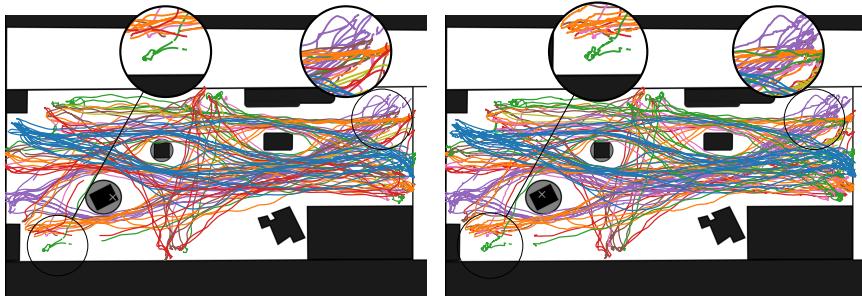


Figure 3.8: Filtering methods in a 4-minute recording from Scenario 1. **Left:** trajectories filtered using the most reliable marker. **Right:** trajectories filtered using the average of the tracking locations of each marker. Although the average tracking markers method provides longer tracks, it induces jerkier trajectories, especially near the boundaries of the motion capture volume (e.g., bottom left and top right).

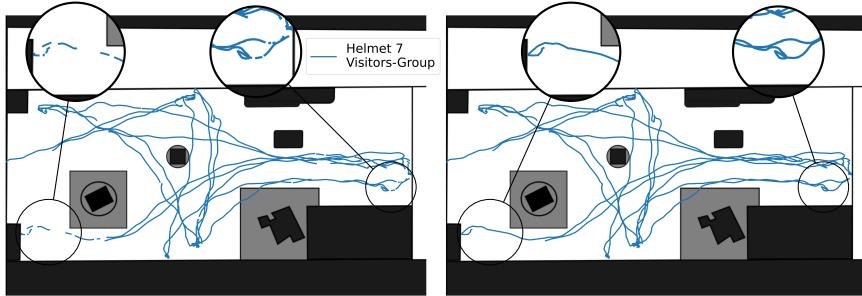


Figure 3.9: Example of a 4-minute Helmet trajectory in Scenario 1. **Left:** raw trajectory data depicting gaps, especially around extreme environmental locations. **Right:** post-processed tracing with 100 maximum positions without tracking (1s) interpolation, showcasing enhanced continuity and completeness in the trajectory.

- **Minimal distance between people (m)** measures the minimum distance observed between individuals in the dataset. It provides insights into the proximity of human agents during their interactions, offering valuable data for studies related to personal space (proxemics) and social dynamics.

- **Number of 8-second tracklets** counts the non-overlapping tracklets of 8-second duration after downsampling to 0.4s and applying a moving average filter. These choices align with current trajectory prediction benchmarks such as those outlined in [52]. These tracklets offer discrete temporal segments for analysis, ensuring compatibility with existing evaluation standards in trajectory prediction.
- **Motion speed** (m/s) represents the velocity of all human agents. A higher standard deviation in motion speed indicates a diverse range of behaviors within the dataset. This diversity is essential for capturing various trajectory patterns and improving robustness in trajectory prediction models. This metric is computed in the 8-second tracklets.
- **Path efficiency** measures the linearity of trajectories in the dataset, ranging between 0 and 1 [1]. It is calculated by dividing the distance between the first and last points by the total navigated distance. A lower coefficient indicates more complex and non-linear trajectories. This metric is also computed in the 8-second tracklets.

3.4.2 Comparison with State-of-the-Art Datasets

We compare our dataset with the THÖR dataset and the ETH/UCY trajectory prediction benchmark. The THÖR dataset encompasses three distinct scenarios, each featuring participants performing different tasks such as individual and group movement, box transportation, different amounts of obstacles, and a mobile robot in the environment. In THÖR Scenario 1 (THÖR-S1), participants navigate the environment with one static obstacle. THÖR Scenario 2 (THÖR-S2) introduces a mobile robot navigating around the static obstacle while participants continue their tasks. Finally, in THÖR Scenario 3 (THÖR-S3), the mobile robot becomes a static obstacle, and an additional obstacle is added to the scene. The ETH/UCY trajectory prediction benchmark consists of five scenes: ETH, HOTEL, UNIV, ZARA1, and ZARA2. These scenes represent five outdoor public spaces that capture natural human motion patterns, resulting in a benchmark widely used by the human trajectory prediction community [85, 28, 103].

Firstly, we show the tracking durations in Fig. 3.10. THÖR presents consistent average tracking durations around 15.5 to 17.6 seconds across the three scenarios. In contrast, THÖR-MAGNI shows wider variations. For instance, Scenario 4 features longer tracking durations (averaging 41.3 seconds), whereas Scenario 2 has the shortest durations (averaging 17.1 seconds). This variability can be attributed to participants' density; Scenarios 4–5, involving fewer human agents in a smaller space, may contribute to higher quality tracking. Nevertheless, THÖR-MAGNI has comparable or higher tracking time than THÖR. Furthermore, compared to the ETH/UCY benchmark (i.e., ETH, HOTEL, UNIV, ZARA1, and ZARA2 scenes), THÖR-MAGNI offers comparable

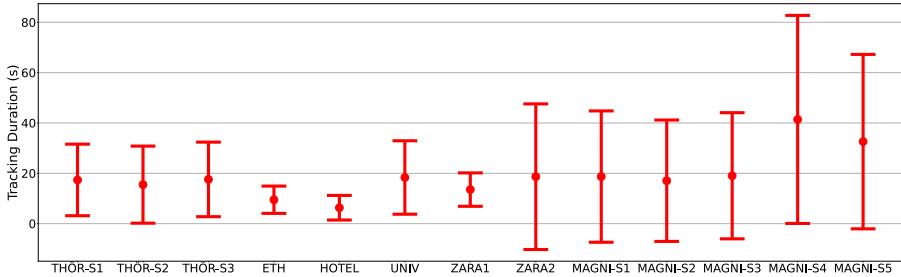


Figure 3.10: Tracking durations (mean \pm one standard deviation) across datasets in seconds. Scenarios 1–3 of THÖR-MAGNI provide comparable tracking durations to previous datasets, while Scenarios 4 and 5 provide longer tracks.

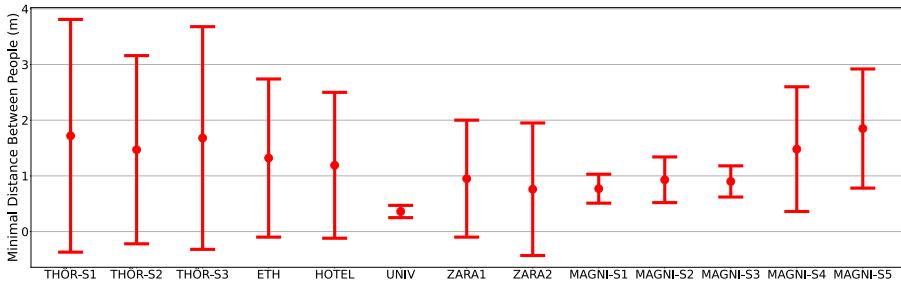


Figure 3.11: Minimal distance between people (mean \pm one standard deviation) across datasets in meters. Lower spatial navigational freedom in Scenarios 1–3 of THÖR-MAGNI potentiates reduced social distances between participants. These results are more consistent with the ZARA1 and ZARA2 scenes, while Scenario 4 and 5 (with more spatial freedom) show similar results to THÖR, ETH, and HOTEL datasets.

or longer tracking durations. This comparison makes our dataset more valuable than its predecessors for long-term human motion prediction and human-robot interactions.

Secondly, we compare the minimal distance between people in Fig. 3.11. Again, human density plays an important role: THÖR-MAGNI Scenarios 1–3 show low values comparable to those in ZARA1/ZARA2, indicating more proximity between humans, while Scenario 4 and 5 reach values similar to THÖR, ETH, and HOTEL. The higher participant density in THÖR-MAGNI Scenarios 1–3 reduces spatial navigational freedom, increasing interactions and decreasing social distances between individuals.

Thirdly, the motion speed statistics are shown in Fig. 3.12. Despite the higher participant density in Scenarios 1–3 of THÖR-MAGNI, these datasets

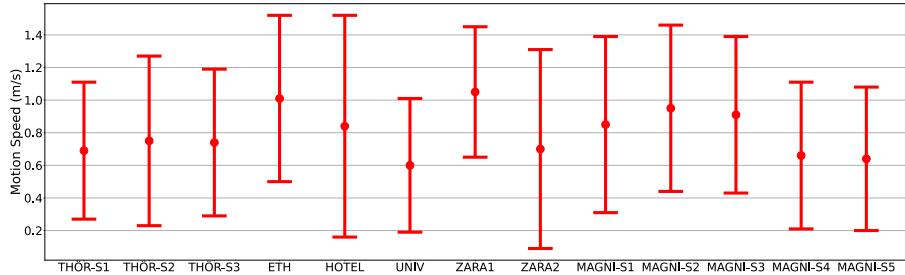


Figure 3.12: Motion speed (mean \pm one standard deviation) for 8-second tracklets across datasets in meters per second.

feature faster human agent navigation than THÖR and akin to those in ETH, HOTEL, and ZARA1 scenes, possibly influenced by the task of object transportation, impacting their velocity profiles. Participants in Scenarios 4–5 of THÖR-MAGNI have an average velocity similar to those in THÖR, UNIV, and ZARA2. Also, generally, THÖR-MAGNI shows comparable standard deviations in motion speeds, indicating diverse and varied movement patterns among human agents. The similarity of the velocity profiles to previous datasets suggests that our dataset is also natural and diverse.

Finally, we compare path efficiency and the number of tracklets in Fig. 3.13. Regarding trajectory linearity, Scenarios 1–3 are aligned with the THÖR and HOTEL datasets, while the other datasets from the ETH/UCY benchmark contain more linear and less complex trajectories. It is also worth noting that THÖR-MAGNI Scenario 4 and 5 display the lowest average metrics (0.78 and 0.75, respectively). The presence of a moving robot might influence these scenarios, prompting human agents to navigate cautiously and align their motion with the robot’s motion profile. Furthermore, THÖR-MAGNI presents a much higher number of non-overlapping tracklets than the other datasets.

These distinctive features make our dataset uniquely challenging, diverse, and valuable as a benchmark for evaluating human trajectory prediction methods. The heightened complexity and diverse range of trajectories in THÖR-MAGNI can provide a robust platform for assessing the effectiveness of trajectory prediction methods, thereby increasing the breadth and depth of research in this area.

3.5 Conclusion

In this chapter, we presented THÖR-MAGNI, a comprehensive human and robot navigation and interaction dataset, extending THÖR with 3.5 times more motion data, novel interactive scenarios, and rich contextual annotations. Both datasets are accessible online at <http://thor.oru.se/>. To further sup-

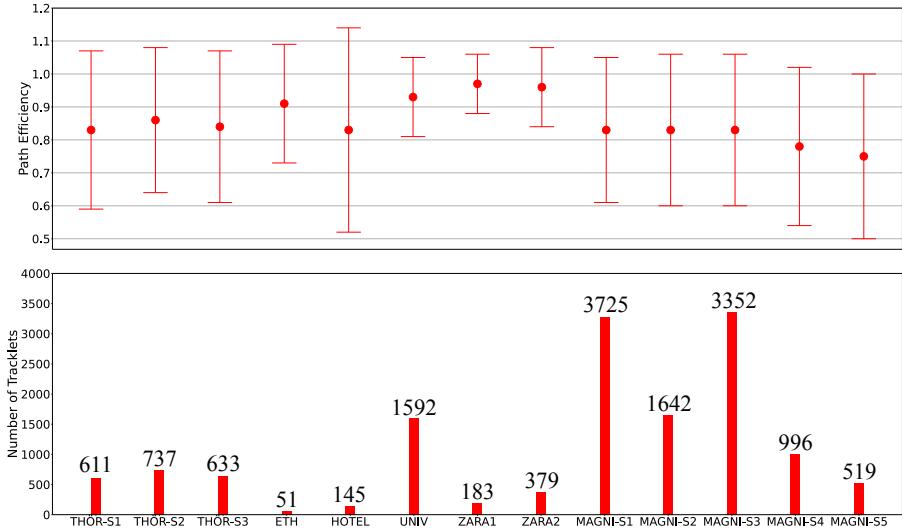


Figure 3.13: **Top:** path efficiency (mean \pm one standard deviation) across datasets, where lower results mean more linear trajectories. **Bottom:** number of non-overlapping 8-second tracklets per dataset. THÖR-MAGNI provides the highest amount of non-linear trajectories.

port researchers, THÖR-MAGNI comes with a dedicated set of user-friendly tools: a dashboard and a specialized Python package called *thor-magni-tools* specifically designed to streamline the visualization, filtering, and preprocessing of raw data. These resources aim to improve the accessibility and usability of the THÖR-MAGNI dataset.

We created THÖR-MAGNI to fill a gap in human motion analysis datasets, limiting HRI research: a lack of comprehensive inclusion of exogenous factors and essential target agent cues hinders holistic studies of human motion dynamics. Unlike existing datasets, THÖR-MAGNI includes a broader set of contextual features and offers multiple variations to facilitate factor isolation. Our dataset integrates different modalities, such as walking trajectories, eye-tracking data, and environmental sensory inputs captured by a mobile robot.

The THÖR-MAGNI dataset has already been used in research papers, demonstrating its usefulness for training class-conditioned trajectory prediction models as described in Chapter 4 and investigating visual attention during human-robot interaction and navigation in shared environments with robots [87]. Furthermore, we set a THÖR-MAGNI trajectory prediction challenge on the Long-term Human Motion Prediction Workshop - ICRA 2024⁷. We set the benchmark to test the generalization capabilities of trajectory prediction mod-

⁷<https://motionpredictionicra2024.github.io/challenge.html>

els in diverse indoor environments. To that end, we provide the five scenarios with unique obstacle layouts, trajectory patterns, and human roles. Out of those, four should be used for training and validation, and the remaining one should be used for testing.

We aim to study trajectory classes to enhance trajectory prediction in various settings. THÖR-MAGNI addresses the gap in the literature, where labeled motion trajectory datasets of heterogeneous agents in industrial environments are scarce, by incorporating human roles that are semantically linked to industrial tasks. The diversity of human roles, especially in Scenarios 2–3, fosters the research of class-conditioned trajectory prediction methods in those industrial environments. We take these steps in Chapter 4, where we leverage THÖR-MAGNI to explore how observable classes (in this case represented by human roles) can improve trajectory prediction. Specifically, we conduct a comprehensive performance analysis of trajectory prediction methods under different data conditions, including class imbalance and low-data regimes.

Chapter 4

Trajectory Prediction with Observable Classes

Forecasts usually tell us more of the forecaster than of the future.

— Warren Buffett

Observable classes represent semantic attributes that categorize moving agents, such as tasks, roles, or other agent-specific characteristics that may influence their dynamics. When these attributes affect the dynamics of those agents, class-conditioned trajectory predictors become appealing for achieving more accurate predictions. By conditioning on observable classes, forecasting uncertainty is reduced, as the predictor is constrained to forecast the trajectory of a specific class, narrowing the input data distribution to a defined subset. However, class-conditioned trajectory predictors still need to be explored in the literature, particularly in mobile robot applications and limited data scenarios, primarily due to the scarcity of relevant datasets. This chapter presents a performance analysis of class-conditioned trajectory prediction methods using two datasets, including the previously introduced THÖR-MAGNI dataset designed for indoor mobile robot environments and the Stanford Drone Dataset (SDD) for outdoor environments. We propose a range of efficient deep learning methods conditioned on observable classes, evaluate their performance on the two datasets, and compare them to a pattern-based predictor. The results demonstrate that conditioning on class labels generally enhances prediction accuracy. More importantly, we observe variations when training with imbalanced datasets or in new environments where data is limited. Specifically, our findings indicate that deep learning methods perform well with balanced datasets. However, in scenarios with limited data, such as a robot's cold start in an unfamiliar environment or when classes are imbalanced, pattern-based methods may be more effective.

4.1 Introduction

4.1.1 Motivation and Contributions

In the context of mobile robots in indoor dynamic environments, prior art has yet to explore prediction approaches that are aware of observable classes. This is particularly evident given that human motion trajectory datasets annotated with observable classes are still rare in indoor robotics settings (see Chapter 3). Moreover, existing heterogeneous trajectory prediction methods tailored for autonomous driving do not transfer well to robotics settings, as they depend on domain-specific contextual features [14]. Furthermore, robotics applications present unique challenges, such as the cold-start scenario, where a robot enters and continuously navigates a previously unseen environment with limited data [23]. Additionally, both robotics and autonomous driving domains may feature non-uniform class distributions, leading to decreased performance of deep learning-based methods [55]. It is crucial to understand whether prediction methods can benefit from class representations in applications with scarce or imbalanced data, and if so, to what extent and under which specific circumstances.

This chapter presents an in-depth study of class-conditioned trajectory prediction methods under different training data conditions. To that end, we analyze class-conditioned methods that are transferrable to other environments, allowing for applicability across different settings. We adapt several deep learning methods to include class labels and compare them to a pattern-based approach, CLIFF-LHMP [107], which uses Maps of Dynamics [56] (MoD). We argue that these methods are well-suited for application in new environments to support safe mobile robot navigation. Specifically, we evaluate four deep learning models such as LSTMs, Transformers, GANs, and VAEs, along with their respective conditional counterparts, on two datasets of human trajectory data in indoor settings (THÖR-MAGNI) [86] and road agent trajectories (Stanford Drone Dataset) [79]. We chose SDD and THÖR-MAGNI due to their distinct characteristics: imbalanced outdoor road agents and balanced human tasks in an indoor robotics environment, respectively.

In contrast to previous methods [67, 33, 72], our proposed deep learning approaches are both memory and energy-efficient as they do not require training or running individual modules per class. We assess their performance across diverse training data conditions, considering both balanced and imbalanced datasets (where class proportions are uniform and non-uniform, respectively), as well as various amounts of training data. The study of imbalanced datasets is important as deep learning methods may struggle to predict underrepresented classes, which is particularly impactful when these classes represent vulnerable road users such as pedestrians. The study of various training data amounts reflects a practical challenge in mobile robotics, where the system is deployed in new environments with limited acquired data, yet requiring anticipation of

other agents' movements for safe navigation. Through this comparative study, we aim to show the preferred methods for specific settings, quantifying their performance in different data regimes and class-imbalanced datasets.

In summary, we make the following contributions:

- We establish a set of deep learning-based trajectory prediction baselines¹ for outdoor mixed traffic scenarios (SDD) and an indoor mobile robots dataset (THÖR-MAGNI) to solve the O-TP task defined in Sec. 1.2.
- We analyze the performance of four deep learning methods against a pattern-based approach (MoD) that considers observable classes in THÖR-MAGNI and SDD.
- We show that class-conditioned methods often outperform their unconditional counterparts. In addition, we show that MoD approaches are preferable over the deep generative methods for class-imbalanced datasets and superior to single-output deep learning methods in low data regimes.

4.1.2 Outline

The chapter is organized as follows: in Sec. 4.2, we formalize the problem of class-conditioned trajectory prediction. Sec. 4.3 details the deep learning methods employed in this study. Sec. 4.4 outlines the datasets, implementation specifics of the trajectory prediction models, and the evaluation metrics used in our performance analysis. Finally, Sec. 4.5 presents a quantitative and qualitative assessment of the proposed methods across the evaluated datasets as well as a detailed analysis of the limitations of observable classes, and the chapter concludes in Sec. 4.6.

4.2 Problem Formalization and Notation

In this section, we introduce notation and formalize our trajectory prediction task: class-conditioned trajectory prediction with observable classes (O-TP). Following the introductory problem formalization in Sec. 1.2, each agent, A_i in a heterogeneous trajectory dataset, is associated with an observable class c_{A_i} . The trajectories of each agent are transformed into tracklets of fixed-length $\mathbf{S} = (\mathbf{s}_t)_{t=1}^O$, where $\mathbf{s}_t = (x, y, \dot{x}, \dot{y})$ for environment-aware formulations and $\mathbf{s}_t = (\dot{x}, \dot{y})$ for environment-agnostic formulations. Velocity is decomposed into 2D speed and orientation for the MoD approach. The future of an observed tracklet consists of 2D velocities, $\mathbf{Y}_{\mathbf{S}} = ((\dot{x}_t, \dot{y}_t))_{t=O+1}^{T_P}$ of length $L = T_P - O$,

¹Code available at <https://github.com/tmralmeida/class-cond-trajpred>

which are subsequently converted into future positions $\mathbf{P}_{\mathbf{S}}$. Therefore, for the O-TP task, the goal is to predict the future of a tracklet through:

$$\psi_p: \mathbf{S}_k, c_{A_k} \mapsto \mathbf{Y}_{\mathbf{S}_k}, \quad (4.1)$$

where $\mathbf{Y}_{\mathbf{S}_k}$ is the future corresponding to the tracklet \mathbf{S}_k of a particular agent A_k with class c_{A_k} . In this work, ψ_p can be a deep learning architecture or a MoD-based trajectory predictor such as CLiFF-LHMP [107].

4.3 Deep Learning Methods

This section presents the deep learning methods to learn the prediction task defined in Sec. 4.2. Our analysis covers single-output trajectory predictors, including the Long Short-Term Memory (LSTM) approach, specifically the RED method [7], and a Transformer-based model (TF) inspired by [35]. We also explore multiple-output approaches, such as Generative Adversarial Networks (GANs) following the framework in [51] and Variational Autoencoders (VAEs) based on [85]. Single-output methods generate one deterministic trajectory for the future, while multiple-output models produce multiple plausible trajectories that reflect the variability of the training dataset’s distribution (see Sec. 1.2.2). We then extend these methods to their class-conditioned versions: cRED, cTF, cGAN, and cVAE, where the class identifiers are mapped through a class embedding layer $CEemb$ to generate dense feature representations. Using an embedding layer offers several advantages over sparse representations, such as one-hot encodings: (1) improved memory efficiency by representing categorical inputs as dense vectors instead of high-dimensional sparse ones, and (2) the capacity to exploit potentially non-orthogonal continuous vector representations that can capture semantic relationships between classes.

The mapping ψ_p presented in Sec. 4.2 has two variants: ψ_{sp} for single-output methods and ψ_{mp} for multiple-output methods. Single-output methods consist of an embedding layer Emb , followed by a temporal encoder Enc , and a decoder Dec , represented as:

$$\psi_{sp}(\mathbf{S}, c) = Dec((Enc \circ Emb(\mathbf{S})) \oplus CEemb(c)). \quad (4.2)$$

In multiple-output methods, ψ_p incorporates a noise vector (\mathbf{z}) to introduce variability in the output, expressed as:

$$\psi_{mp}(\mathbf{S}, c, \mathbf{z}) = Dec((Enc \circ Emb(\mathbf{S})) \oplus CEemb(c) \oplus \mathbf{z}). \quad (4.3)$$

In all models, the embedding and decoder layers are implemented using multilayer perceptrons (MLPs), while the temporal encoder may vary, using either an LSTM or a Transformer-based encoder.

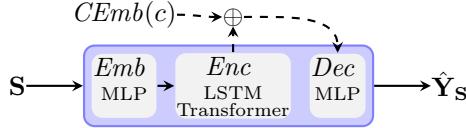


Figure 4.1: Single-output unconditional and conditional methods (**dashed arrows**). The input state \mathbf{S} is first projected into a fixed-length vector via the embedding module Emb , then processed by the temporal feature encoder Enc . Finally, the decoder Dec maps the encoded features to the predicted future velocity vector $\hat{\mathbf{Y}}_{\mathbf{S}}$. In conditional models, a class embedding layer $CEmb$ extracts a dense representation, which is concatenated with the encoded temporal features before decoding.

4.3.1 Single-Output Methods: LSTMs and Transformers

The first step is embedding the input state representation \mathbf{S} using an MLP network. Conditioned methods, cRED and cTF, also embed the integer class label c with an embedding layer. Subsequently, for RED and cRED, the embedded input vector passes through an LSTM layer, while for TF and cTF, the encoded input vector undergoes a Transformer-based encoder. An MLP-based decoder then generates the predicted sequence of velocity vectors from the encoded vectors. For conditional variants (cRED and cTF), class embeddings are concatenated with the temporal features before decoding. The entire process is illustrated in Fig. 4.1. We train single-output networks with the Mean Squared Error (MSE) loss:

$$\mathcal{L}_{\text{MSE}}(\mathbf{P}_{\mathbf{S}}, \hat{\mathbf{P}}_{\mathbf{S}}) = \frac{1}{L} \sum_{j=O+1}^{T_P} \|\mathbf{p}_j - \hat{\mathbf{p}}_j\|_2^2, \quad (4.4)$$

where $\hat{\mathbf{P}}_{\mathbf{S}}$ represents the estimated sequence of positions, $\mathbf{p}_j = (x, y)$ the ground truth 2D position at time step j and $\hat{\mathbf{p}}_j$ the corresponding prediction. The training data for these methods, $\{(\mathbf{S}_k, c_{A_k}, \mathbf{P}_{\mathbf{S}_k})\}_k$, consists of triplets of tracklets, ground truth class labels, and the corresponding ground truth positions.

4.3.2 Multiple-Output Methods: GANs and VAEs

The training data for multiple-output methods, $(\mathbf{S}_k, c_{A_k}, \mathbf{Y}_{\mathbf{S}_k}, \mathbf{P}_{\mathbf{S}_k})_k$, comprises quadruples of observed tracklets, ground truth class labels, future tracklets, and corresponding ground truth positions. The future tracklets $\mathbf{Y}_{\mathbf{S}_k}$ pertain to the input for training auxiliary networks: the discriminator in GANs and the recognition network in VAEs. Further details are provided in the subsequent sections.

GAN-based Trajectory Predictors

A GAN aims to reconstruct the generative process of the underlying input data using two modules: the generator (G) and the discriminator (D). The generator maps the input \mathbf{S} and a latent random vector \mathbf{z}_G to a realistic future sequence of velocities \mathbf{Y}_S . We sample the latent vector from a standard normal Gaussian distribution. Simultaneously, the discriminator differentiates between both real and generated future velocity vectors, \mathbf{Y}_S and $\hat{\mathbf{Y}}_S$, respectively. This adversarial training scenario is essential for producing multiple plausible future trajectories. In cGAN, the generator and discriminator incorporate the trajectory class as an additional input. We optimize the GAN and cGAN discriminators using the binary cross-entropy loss, while the GAN generator is optimized with a weighted sum given by:

$$\mathcal{L}_G(\mathbf{Y}, \hat{\mathbf{Y}}) = \lambda_1 \mathcal{L}_{\text{MSE}} + \lambda_2 \left(\frac{1}{2} \mathbb{E}[(D(\mathbf{Y}) - 1)^2] + \frac{1}{2} \mathbb{E}[D(\hat{\mathbf{Y}})^2] \right), \quad (4.5)$$

where λ_1 and λ_2 are the weights applied to the MSE term (Eq. (4.4)) and to the GAN loss, respectively. For the conditional variant (cGAN), the class is additionally fed as input to both the generator and the discriminator, resulting in the following loss function:

$$\mathcal{L}_{cG}(c, \mathbf{Y}, \hat{\mathbf{Y}}) = \lambda_1 \mathcal{L}_{\text{MSE}} + \lambda_2 \left(\frac{1}{2} \mathbb{E}[(D(c, \mathbf{Y}) - 1)^2] + \frac{1}{2} \mathbb{E}[D(c, \hat{\mathbf{Y}})^2] \right), \quad (4.6)$$

Fig. 4.2 illustrates the network configurations for GAN and cGAN models. The generators generally have the same layer configuration as the TF model. The difference is the latent vector, \mathbf{z}_G , which is concatenated with the temporal features from the Transformer and passed to the decoder. Analogous to [51], the discriminator comprises a Transformer-encoder network and an MLP in the last layer. For cGAN, both the generator and the discriminator concatenate \mathbf{S} to the agent's class embedding. The generator also concatenates the class embeddings to the input of the decoder. For GAN-based predictors, the generator can be seen as the network providing the ψ_{mp} mapping.

VAE-based Trajectory Predictors

VAE-based predictors consist of two main networks: the prior network p_θ and the recognition network q_ϕ . The prior network maps the input state \mathbf{S} and a latent vector \mathbf{z}_V to the predicted future tracklet $\hat{\mathbf{Y}}_S$. In contrast, the recognition network learns to map the ground truth \mathbf{Y}_S to the parameters of a Gaussian distribution, representing a lower-dimensional latent space. We adopt a standard normal Gaussian as the prior for the distribution of future trajectories. The Kullback-Leibler (KL) divergence is used to align the learned distribution to the prior, contributing to the VAE's loss function:

$$\mathcal{L}_V(\mathbf{z}_V, \mathbf{Y}_S, \mathbf{S}) = \beta_1 \mathcal{L}_{\text{MSE}} - \beta_2 (D_{\text{KL}}[q_\phi(\mathbf{z}_V | \mathbf{Y}_S) \| p_\theta(\mathbf{z}_V | \mathbf{S})]), \quad (4.7)$$

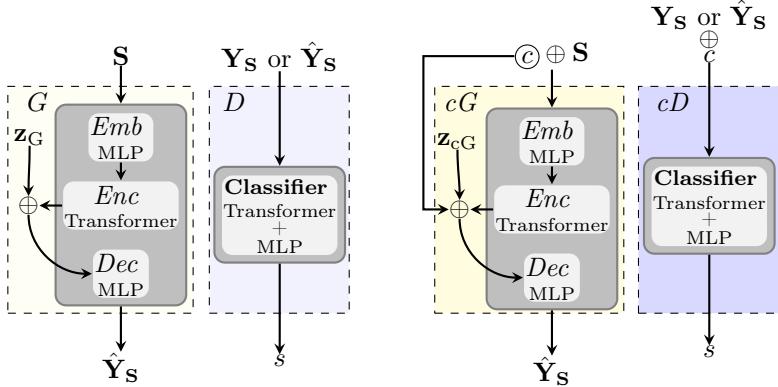


Figure 4.2: GAN-based models: unconditional GAN (**left**) and conditional cGAN (**right**). In the generator (yellow boxes), the input state (\mathbf{S}) is embedded via *Emb*, resulting in a fixed-length vector, which is then processed by the temporal feature encoder *Enc*. The decoder *Dec* combines the encoded temporal features with a latent vector (\mathbf{z}_G) to generate the predicted future velocity vector ($\hat{\mathbf{Y}}_S$). For conditional models, the class embeddings are concatenated with the temporal features and the noise vector before being passed through the decoder. In the discriminator (blue boxes), the future ground truth velocity vector (\mathbf{Y}_S) or the predicted velocity vector ($\hat{\mathbf{Y}}_S$), concatenated with class embeddings for conditional models, is processed through a Transformer-based encoder and an MLP to classify the input as real or generated.

where β_1 and β_2 are the weights applied to the MSE and KL terms, respectively. For the conditional variant (cVAE), the agent’s class is added as input to both p_θ and q_ϕ , resulting in the following loss function:

$$\mathcal{L}_{cV}(\mathbf{z}_V, \mathbf{Y}_S, \mathbf{S}, c) = \beta_1 \mathcal{L}_{\text{MSE}} - \beta_2 (D_{\text{KL}}[q_\phi(\mathbf{z}_V | \mathbf{Y}_S, c) \| p_\theta(\mathbf{z}_V | \mathbf{S}, c)]), \quad (4.8)$$

Fig. 4.3 shows the network configurations for the VAE and cVAE models. The predictor’s network (prior network and decoder) configuration is identical to the generator in the GAN and cGAN models. The difference lies in the training process, where the latent vector \mathbf{z}_V is sampled based on parameters generated by the recognition network (q_ϕ). The recognition network processes the ground truth prediction akin to p_θ but concludes with two linear layers producing the Gaussian parameters. For VAE-based predictors, p_θ followed by the decoder can be seen as the architecture providing the ψ_{mp} mapping.

4.4 Experiments

This section presents experiments conducted to analyze the impact of observable classes (i.e., roles and agent classes) on trajectory prediction accuracy. We

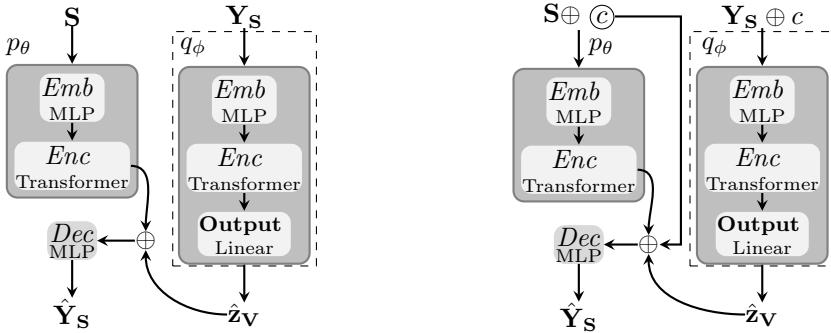


Figure 4.3: VAE-based models: unconditional VAE (left) and cVAE (right). The recognition network (q_ϕ , shown with a dashed border) is used exclusively during training. In the prior network, the input state (S) is embedded into a fixed-length vector via *Emb*, which is then processed by the temporal feature encoder *Enc*. The decoder *Dec* combines these encoded temporal features with a latent vector (z_v) to generate the predicted future velocity vector (\hat{Y}_s). During training, the latent vector is derived from the recognition network, q_ϕ , while during inference, it is sampled from a standard Gaussian distribution. The recognition network processes the ground truth future velocity vector (Y_s) to produce the parameters of a Gaussian distribution.

begin by outlining the experimental setup, including data preprocessing and baseline models. We then describe the implementation details of the quantitative and qualitative experiments and the evaluation metrics used to assess the predictors' performance.

4.4.1 Datasets and Baselines

In this study, we evaluate and compare the performance of the trajectory prediction methods described in Sec. 4.3 on two datasets, THÖR-MAGNI [86] and SDD [79]. These datasets represent distinct environments, with SDD situated in an outdoor university campus and THÖR-MAGNI in an indoor robotics setting (see Chapter 3 for a detailed description). Both datasets contain various classes of agents: THÖR-MAGNI comprises human-centered agents in an industrial environment, whereas SDD includes road agents. Importantly for our analysis, the two datasets show a substantial difference in class proportions, as illustrated in Fig. 4.4. Specifically, SDD shows class imbalance compared to THÖR-MAGNI. This inter-dataset class imbalance poses challenges to accurate trajectory prediction. We analyze how these challenges are handled by the two categories of predictors: deep learning models and MoD approaches.

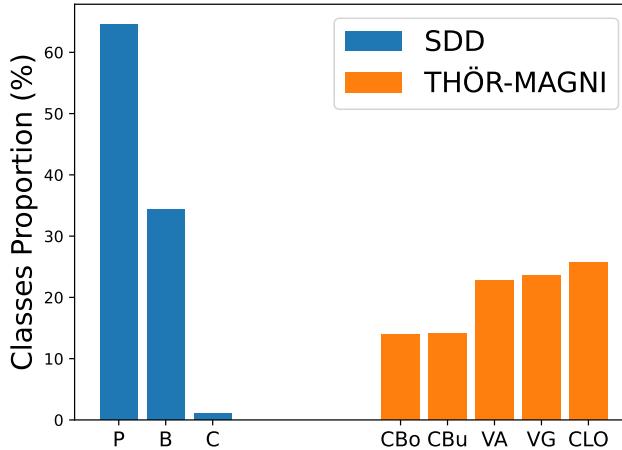


Figure 4.4: Agent class distribution in SDD (**left blue**) and THÖR-MAGNI (**right orange with oblique lines**). In SDD, the *Pedestrian* class (“P”) is the most representative, followed by the *Bicyclist* class (“B”), with the *Car* class (“C”) being the least representative. In the THÖR-MAGNI dataset, class proportions are more uniform, with the *Carrier-Large Object* (“CLO”) being the most prevalent, followed by *Visitors-Group* (“VG”), *Visitors-Alone* (“VA”), *Carrier-Bucket* (“CBu”) and *Carrier-Box* (CBo).

THÖR-MAGNI

The THÖR-MAGNI dataset, described in detail in Chapter 3, is of particular interest in this work as it captures human motion trajectories in a robotics environment where participants perform various tasks, such as moving objects like boxes, buckets, and poster stands. This study focuses on Scenarios 2, 3A, and 3B, including data from 30 participants over 1.5 hours of recorded motion. Five distinct agent roles are recorded in these scenarios: *Carrier-Large Object*, *Visitors-Group*, *Visitors-Alone*, *Carrier-Bucket*, and *Carrier-Box*, with corresponding sample proportions of 25.7%, 23.6%, 22.7%, 14.1%, and 13.9%. These roles are associated with different physical tasks, influencing the corresponding motion patterns, especially the velocity profile. Specifically, Tab. 4.1 summarizes the data, showing the number and percentage of 20-time-step trajectories and velocity statistics per role across scenarios. A broad look at this table shows that participants within the same role tend to move at similar velocities across different scenarios. However, if we compare the roles within each scenario, we can observe that the *Carrier-Bucket* (small object) moves the fastest, followed by the *Carrier-Box*. On the opposite side of the spectrum, *Carrier-Large Object* is the slowest role. This is expected

| Role | Magni-S2 | Magni-S3A | Magni-S3B |
|-----------------------------|---|---|---|
| <i>Carrier-Box</i> | 223 (14.13%) $1.12 \frac{m}{s} \pm 0.21$ | 224 (13.74%) $1.15 \frac{m}{s} \pm 0.27$ | 220 (13.70%) $1.08 \frac{m}{s} \pm 0.26$ |
| <i>Carrier-Bucket</i> | 224 (14.20%) $1.21 \frac{m}{s} \pm 0.24$ | 226 (13.87%) $1.21 \frac{m}{s} \pm 0.20$ | 227 (14.13%) $1.13 \frac{m}{s} \pm 0.18$ |
| <i>Carrier-Large Object</i> | 394 (24.97%) $0.72 \frac{m}{s} \pm 0.27$ | 440 (26.99%) $0.68 \frac{m}{s} \pm 0.32$ | 405 (25.22%) $0.76 \frac{m}{s} \pm 0.36$ |
| <i>Visitors-Alone</i> | 452 (28.64%) $0.95 \frac{m}{s} \pm 0.20$ | 318 (19.51%) $0.92 \frac{m}{s} \pm 0.29$ | 322 (20.05%) $0.87 \frac{m}{s} \pm 0.32$ |
| <i>Visitors-Group</i> | 285 (18.06%) $0.92 \frac{m}{s} \pm 0.31$ | 422 (25.89%) $0.87 \frac{m}{s} \pm 0.26$ | 432 (26.90%) $0.84 \frac{m}{s} \pm 0.31$ |
| Global | 1578 (100%) $0.95 \frac{m}{s} \pm 0.51$ | 1630 (100%) $0.91 \frac{m}{s} \pm 0.48$ | 1606 (100%) $0.90 \frac{m}{s} \pm 0.48$ |

Table 4.1: Data summary per role in our experiments: number and ratio of 20-time steps tracklets, velocities average and standard deviation.

as transporting a small object implies less effort than moving a bigger object like a box. Moreover, a group of two people moves a poster stand (a large object), which entails a team effort and, therefore, a slower pace. Finally, it is worth highlighting that, on average, *Visitors-Group* move slower than *Visitors-Alone*, reflecting the influence of group dynamics on movement speed.

Stanford Drone Dataset

SDD encompasses 5 hours of heterogeneous trajectory data from 60 videos on the Stanford University campus. It includes trajectory data on cyclists, pedestrians, skateboarders, carts, cars, and buses. Notably, certain classes such as *Bicyclist* and *Pedestrian* coexist in shared spaces (see Fig. 4.5) but exhibit distinct movement patterns (e.g., bicyclists typically move faster). The dataset provides agent coordinates in pixel values. For our evaluation, we choose videos that contain at least two classes of agents and have more than 10 trajectories per class, resulting in 7 scenes (gates, little, nexus, coupa, bookstore, death-Circle, hyang) and three agent classes: *Pedestrian*, *Bicyclist* and *Car* with corresponding sample proportions of 64.6%, 34.3%, and 1.1%.

Baseline Models

We compare the performance of our deep learning class-conditioned predictors with the following baselines: unconditional deep learning models (i.e., without class conditioning) and MoD-based predictors. We also include a class-conditioned version of the MoD-based predictor, which we refer to as cMoD.

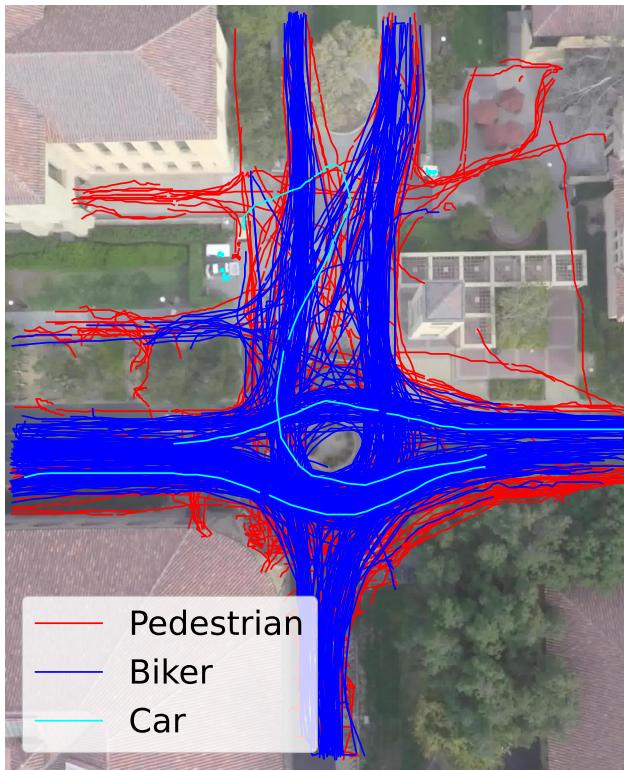


Figure 4.5: Example of trajectory data in SDD (deathCircle scene) for the three classes evaluated in this work.

Maps of Dynamics encode spatial or spatio-temporal motion patterns as a feature of the environment [56]. By generalizing velocity observations, human dynamics can be represented through flow models. Prior work proposes CLiFF-LHMP [107], which exploits MoD for long-term human motion prediction. It uses a multi-modal probabilistic representation of a velocity field (CLiFF-map), which is built from observations of human motion, and employs Semi-Wrapped Gaussian mixture models (SWGMM) to capture local velocity distributions. This method implicitly accounts for obstacle layouts and predicts trajectories that follow the environment's complex topology. CLiFF-LHMP excels in predicting up to 50 s ahead, [107], even with sparse, incomplete, and very limited training data [106]. In a class-conditioned CLiFF-map, individual CLiFF-maps are built for each agent class using their specific trajectories. We refer the reader to [107] for more details on the CLiFF-LHMP approach.

4.4.2 Implementation and Evaluation Details

To evaluate the predictors, we employed a repeated random sub-sampling validation method. For each iteration, we randomly selected $ptr\%$ of the dataset for training and used the remaining $(100 - ptr)\%$ for testing. This process was repeated ten times, with the selection of test and training data being independently randomized in each iteration. In the accuracy analysis (Sec. 4.5.1), we set $ptr = 90$. In the data efficiency analysis (Sec. 4.5.2), we decreased the percentage of data used for training from $ptr = 90$ to $ptr = 10$ in steps of 10. Following current trajectory prediction benchmarks [52], we define the observation length as $O = 8$ and the prediction horizon as $L = 12$. To compare the trajectory predictors, we use the Top- K ADE and FDE, in pixels for SDD and meters for THÖR-MAGNI, as defined in Eqs. (1.1) and (1.2). We report both Top-1 and Top-3 ADE/FDE scores. For Top-1, the single most likely predicted trajectory is selected, while Top-3 evaluates the best match among the three predictions. We calculate and report the mean and standard deviation of these metrics across validation iterations.

For deep learning-based predictors: We maintained a uniform hyper-parameter setting to ensure a fair comparison. All networks were trained for up to 100 epochs with early stopping if no improvement was observed for 20 consecutive epochs. We optimize the networks with the Adam optimizer [47], a learning rate of $1e-3$, and a batch size of 32. We also reduce the learning rate on the plateau of the validation loss during training (patience set to 5 epochs).

For training generative models, including GAN, cGAN, VAE, and cVAE, we have standardized the weights in their respective loss functions. Consequently, $\lambda_1 = \beta_1 = 2$ and $\lambda_2 = \beta_2 = 1$, indicating a preference for the reconstruction of predictions based on the MSE term in the loss functions.

Furthermore, each model receives as input state the position concatenated with the velocity vector for THÖR-MAGNI scenarios. In contrast, for SDD, the velocity vector alone is used as input due to the aggregation of diverse scenes, making the position an irrelevant input feature.

For MoD-based predictors: We use identical parameters for the class-conditioned and the general CLiFF-LHMP. The CLiFF-map grid resolutions for the SDD and the THÖR-MAGNI dataset are 20 pixels and 0.2 m, respectively. The sampling radius is adjusted for each dataset to match the CLiFF map grid resolution. The kernel parameter is set to 5 for all experiments. In the figures and tables presenting the results, CLiFF-LHMP is denoted as MoD, and class-conditioned CLiFF-LHMP is denoted as cMoD.

4.5 Results

In our analysis, we aim to (1) quantify the improvement in trajectory prediction performance when using observable classes and (2) evaluate trajectory prediction performance based on the specific characteristics of the dataset. The latter provides insights into selecting the most suitable trajectory prediction method for a given application context.

4.5.1 Accuracy Analysis Conditioned on Class Balance

Tabs. 4.2 and 4.3 show the Top-1 prediction accuracy results separately for each class on Scenario 2 of the THÖR-MAGNI and SDD datasets, respectively. It also shows the global results for all trajectories (last column for each dataset). A broad view of the THÖR-MAGNI results shows that conditional methods outperform their unconditional counterparts regardless of the type of method (deep learning and MoD). This difference is least pronounced when predicting trajectories from the *Visitors-Group*. We speculate that this may be because the motion patterns of these agents are less structured compared to the other classes, as shown in Fig. 4.6. This also highlights the importance of suitable class labels, such that each class encompasses specific motion patterns, and observable classes may not always do so. For the imbalanced dataset (SDD), deep learning methods face the challenge of identifying a representative number of different motion patterns across classes. This difficulty is most pronounced in single-output deep learning methods (RED and TF). In contrast, cMoD is less sensitive to class proportions and can use class information for more accurate predictions. In summary, we highlight two key points: (1) the superiority of deep learning methods over MoD-based approaches in balanced datasets like THÖR-MAGNI, and (2) the appropriateness of conditional MoD over deep generative methods (cGAN, cVAE) for imbalanced datasets like SDD.

In the MoD-aware predictor, cMoD outperforms general MoD in both datasets. Prediction accuracy improvements were more pronounced in classes with distinct motion patterns, such as *Carrier-Box* and *Carrier-Bucket* (see Fig. 4.6 left and Fig. 4.7), which deviate more from the general motion pattern. In SDD, variances in speed are observed among different classes, as depicted in Fig. 4.8. A single CLiFF-map struggles to accurately model variations across multiple classes, leading to inaccurate predictions compared to the class-conditioned MoD-aware method.

4.5.2 Data Efficiency Analysis

To assess how training data volume affects model performance, we conducted a data efficiency analysis aimed at identifying the most appropriate models for various data settings. Fig. 4.9 shows the performance of single-output methods (RED, TF, and MoD, along with their conditioned variants) in the THÖR-

Table 4.2: Top-1 ADE/FDE scores in THÖR-MAGNI Scenario 2 with a 90% train ratio. Bold values highlight superior performance of conditional models over their unconditional counterparts.

| Model | Carrier-Box | Carrier-Bucket | Visitors-Alone | Visitors-Group | Carrier-Large Object | Global |
|-------|------------------|------------------|------------------|------------------|----------------------|------------------|
| RED | 0.64±0.07 | 0.71±0.06 | 0.81±0.05 | 0.72±0.05 | 0.73±0.05 | 0.74±0.01 |
| | 1.23±0.14 | 1.35±0.18 | 1.53±0.12 | 1.34±0.17 | 1.44±0.12 | 1.41±0.04 |
| cRED | 0.60±0.07 | 0.67±0.06 | 0.78±0.06 | 0.72±0.07 | 0.69±0.03 | 0.71±0.03 |
| | 1.10±0.14 | 1.21±0.15 | 1.48±0.13 | 1.35±0.18 | 1.38±0.10 | 1.34±0.06 |
| TF | 0.66±0.07 | 0.65±0.05 | 0.79±0.04 | 0.74±0.06 | 0.69±0.05 | 0.72±0.02 |
| | 1.24±0.15 | 1.24±0.13 | 1.52±0.12 | 1.40±0.15 | 1.41±0.12 | 1.39±0.04 |
| cTF | 0.60±0.07 | 0.60±0.06 | 0.75±0.04 | 0.68±0.05 | 0.64±0.04 | 0.67±0.02 |
| | 1.10±0.13 | 1.12±0.16 | 1.45±0.14 | 1.29±0.13 | 1.31±0.08 | 1.29±0.05 |
| GAN | 0.76±0.07 | 0.78±0.04 | 0.88±0.07 | 0.80±0.08 | 0.78±0.08 | 0.81±0.05 |
| | 1.50±0.19 | 1.48±0.18 | 1.72±0.14 | 1.52±0.15 | 1.59±0.16 | 1.59±0.09 |
| cGAN | 0.70±0.10 | 0.73±0.06 | 0.85±0.08 | 0.80±0.06 | 0.75±0.06 | 0.78±0.05 |
| | 1.33±0.19 | 1.37±0.14 | 1.67±0.19 | 1.57±0.16 | 1.56±0.13 | 1.53±0.09 |
| VAE | 0.68±0.06 | 0.73±0.09 | 0.84±0.06 | 0.78±0.08 | 0.77±0.07 | 0.77±0.03 |
| | 1.31±0.13 | 1.44±0.20 | 1.62±0.16 | 1.59±0.16 | 1.50±0.15 | 1.49±0.05 |
| cVAE | 0.66±0.05 | 0.74±0.08 | 0.83±0.05 | 0.75±0.06 | 0.72±0.06 | 0.76±0.02 |
| | 1.26±0.11 | 1.43±0.19 | 1.61±0.14 | 1.56±0.13 | 1.44±0.12 | 1.47±0.06 |
| MoD | 0.81±0.11 | 0.92±0.18 | 0.94±0.06 | 0.82±0.10 | 0.83±0.10 | 0.87±0.05 |
| | 1.59±0.25 | 1.78±0.37 | 1.97±0.20 | 1.78±0.24 | 1.73±0.22 | 1.79±0.10 |
| cMoD | 0.73±0.07 | 0.72±0.10 | 0.92±0.09 | 0.83±0.10 | 0.75±0.08 | 0.80±0.05 |
| | 1.40±0.17 | 1.30±0.17 | 1.95±0.22 | 1.80±0.24 | 1.61±0.19 | 1.67±0.09 |

MAGNI dataset. cMoD outperforms deep learning methods in Top-1 ADE in low data regimes for all three scenarios, where 10% of data is available during training. Moreover, performance for deep learning methods declines with less training data. In contrast, MoD approaches (MoD and cMoD) are more stable across different data regimes (i.e., they attain consistent performance, not necessarily superior on all metrics, across all training set ratios). Therefore, the CLiFF-map efficiently captures major human motion patterns with limited training data. Beyond a 30% training data increase, CLiFF-map improvements are less notable, especially compared to the training set expansion from 10% to 20%. Once major motion patterns are captured, the representations stabilize, and unlike deep learning methods, MoD approaches do not show performance improvements with more data. This stability highlights the advantage of the MoD approach when extensive data collection is impractical.

Fig. 4.10 presents the performance of multiple-output methods (VAE, GAN, and MoD, along with their respective conditioned variants) on THÖR-MAGNI datasets. Deep generative methods are more effective in generating one out of $K = 3$ trajectories than MoD-based methods. Fig. 4.11 shows the same performance results on SDD. Contrary to the results in THÖR-MAGNI datasets,

Table 4.3: Top-1 ADE/FDE scores in SDD with a 90% train ratio. Bold values highlight superior performance of conditional models over their unconditional counterparts.

| Model | Pedestrian | Car | Bicyclist | Global |
|-------|-------------------|--------------------|--------------------|-------------------|
| RED | 18.63±0.54 | 8.44±7.48 | 64.08±2.49 | 33.95±0.91 |
| | 37.55±1.22 | 16.63±15.03 | 137.42±5.31 | 71.23±2.07 |
| cRED | 18.76±0.54 | 8.66±7.05 | 64.38±2.53 | 34.14±1.00 |
| | 37.69±1.08 | 16.55±14.30 | 137.56±4.97 | 71.38±1.99 |
| TF | 18.99±0.89 | 9.36±6.91 | 65.33±2.37 | 34.62±0.90 |
| | 37.60±1.76 | 17.79±14.36 | 142.08±4.60 | 72.87±1.70 |
| cTF | 19.00±0.79 | 9.64±7.62 | 64.01±2.67 | 34.19±1.00 |
| | 37.72±1.30 | 18.11±15.32 | 139.97±4.75 | 72.23±1.63 |
| GAN | 20.26±0.69 | 10.99±7.39 | 67.32±2.75 | 36.14±1.07 |
| | 40.27±1.25 | 20.78±14.83 | 145.50±5.04 | 75.79±1.96 |
| cGAN | 20.21±0.49 | 10.51±7.12 | 67.04±2.41 | 36.01±0.75 |
| | 40.31±0.92 | 19.55±14.23 | 144.50±4.61 | 75.46±1.55 |
| VAE | 20.92±1.25 | 10.83±7.00 | 68.22±3.58 | 36.87±1.67 |
| | 41.49±2.32 | 20.81±13.94 | 147.00±6.32 | 77.09±3.08 |
| cVAE | 20.09±0.77 | 10.41±7.28 | 67.80±3.36 | 36.19±1.45 |
| | 39.90±1.44 | 18.72±13.86 | 145.13±6.64 | 75.40±2.81 |
| MoD | 19.88±0.46 | 9.95±11.05 | 64.35±2.02 | 34.60±0.62 |
| | 40.02±1.07 | 20.61±23.07 | 142.51±4.40 | 74.03±1.50 |
| cMoD | 19.69±0.46 | 8.73±9.56 | 63.60±2.02 | 34.21±0.73 |
| | 39.64±1.14 | 18.48±20.16 | 141.01±4.24 | 73.25±1.60 |

in the imbalanced dataset SDD, MoD-based methods consistently outperform deep generative methods across all train set ratios. These results underscore the preference for MoD-based methods for multiple outputs in imbalanced datasets.

4.5.3 Qualitative Results

We provide qualitative trajectory prediction results for each multiple-output approach in Fig. 4.12 and for each single-output method in Fig. 4.13 for the SDD and THÖR-MAGNI datasets, respectively. For both datasets, conditioned methods are more accurate than their unconditional counterparts. On the SDD dataset, which is characterized by imbalanced classes, cMoD is the most effective compared to deep learning methods. On the THÖR-MAGNI dataset, we observed that conditioned deep learning methods outperform both unconditional deep learning methods and the MoD approaches, which is consistent with the quantitative results.

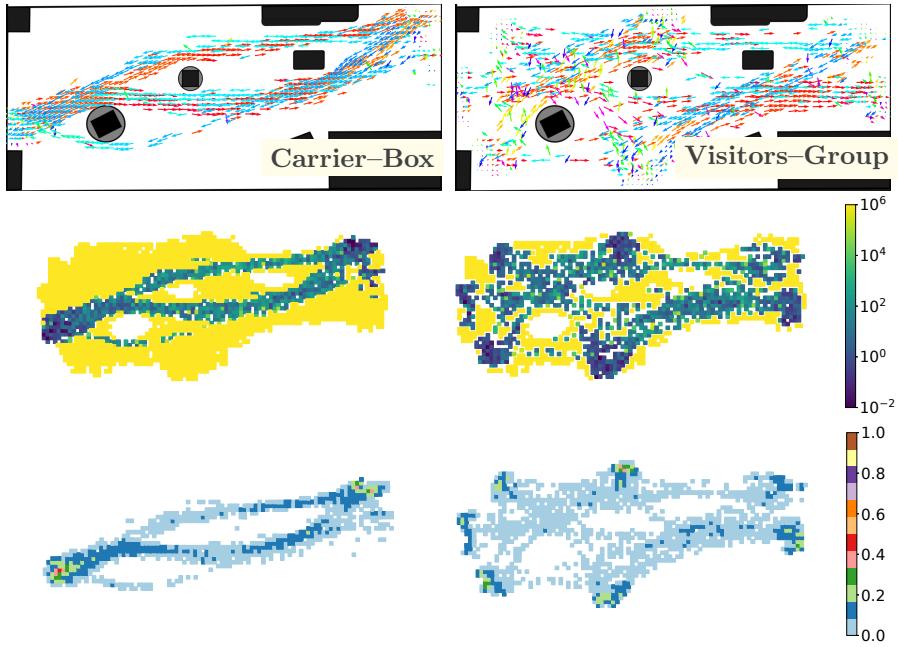


Figure 4.6: Comparison of motion patterns of *Carrier-Box* and *Visitors-Group* in THÖR-MAGNI, Scenario 2. Class-conditioned CLIFF-maps (**top row**) show that *Carrier-Box* has a more distinct and structured motion pattern compared to *Visitors-Group*. The KL divergence heatmap (**middle row**) quantifies the difference between the class-conditioned CLIFF-map and the general one. *Visitors-Group* shows less divergence from the general motion patterns and lower motion intensity (**bottom row**), resulting in a less pronounced improvement in prediction accuracy from using class labels.

4.5.4 Pitfalls of Observable Classes

Observable classes can provide valuable contextual information for trajectory prediction. However, they also introduce several challenges related to ambiguity, semantic inconsistency, and intra-class variability. For example, in the SDD, the observable classes do not always distinguish the underlying dynamics of the agents. A parked car labeled as a *Car* may exhibit a stationary trajectory pattern that closely resembles that of a *Pedestrian* or other static agents. Although the THÖR-MAGNI dataset provides more precise class labels tailored to agent roles and activities, it is still susceptible to semantic ambiguity. For instance, the *Carrier-Box* class may encompass diverse trajectory patterns, including walking with a box, picking up the box (stationary behavior), or navigating without carrying an object. Such intra-class variabil-

ity can devalue the intended semantics of the class, ultimately diminishing the predictive capacity of class-conditioned prediction models. As a result, using these class labels directly in prediction models can lead to misrepresentations of agent dynamics.

To quantitatively analyze this problem, we conduct a Latent Semantic Analysis (LSA) of trajectory data to explore the relationship between tracklets and observable classes in our THÖR-MAGNI dataset. Analogous to LSA in Natural Language Processing [46], our analysis constructs an occurrence matrix \mathbf{O} capturing the frequency of trajectory cluster membership across observable classes. Each column in \mathbf{O} corresponds to an observable class, and each row represents a cluster obtained from the trajectory data. Specifically, we extract trajectory features using Singular Value Decomposition (SVD), followed by clustering in the reduced feature space using the K-means algorithm [66]. SVD was also used in [4] to extract general motion patterns from trajectory data and enhance trajectory representation for the prediction task. The number of clusters is selected based on the Davies-Bouldin Index (DBI), as defined in Eq. (1.5).

In this analysis, “*C*” abbreviates *Carrier*, e.g., *C-LOF* abbreviates *Carrier-Large Object Follower* and *C-LOL* abbreviates *Carrier-Large Object Leader*; “*V*” abbreviates *Visitors*, e.g., *V-A* abbreviates *Visitors-Alone* and *V-G2* abbreviates *Visitors-Group 2*. The resulting matrix highlights clusters that dominate within each class. We consider a cluster as representative of a class if it accounts for at least 20% of the class’s samples. These are shaded in gray in Tab. 4.4. For example, cluster #5 is a representative for both *C-LOF*, *C-LOL*, and *V-A*, showing overlap in the underlying trajectory dynamics of these roles. Similarly, cluster #1 is shared between *C-Box* and *V-G2*, while cluster #3 is representative of *C-Bucket*, *V-A*, and *V-G3*. Tab. 4.4 and Figs. 4.14 and 4.15 visualize this phenomenon, illustrating that observable classes often map onto overlapping regions of the trajectory feature space.

In Fig. 4.14, we present the centroids of the most representative clusters for each of the carrier classes. The centroids for *C-Box* and *C-Bucket* differ substantially, reflecting the distinct motion associated with object transportation. However, *C-LOF* and *C-LOL* share a common centroid (a shorter trajectory) likely due to the stop-and-go dynamics of collaboratively transporting a large object. Similarly, Fig. 4.15 shows the centroids of the most representative clusters for the visitor roles. These centroids can be seen across different observable classes, further highlighting the limitations of using high-level semantic roles as cues to represent distinct motion behaviors.

Shared trajectory clusters across multiple observable classes highlight an important limitation: semantic class labels do not always align with trajectory dynamics. Conditioning trajectory prediction models on such ambiguous classes may degrade performance (e.g., *Visitors-Group* in the THÖR-MAGNI dataset), particularly in scenarios where accurate behavioral modeling is critical.

Table 4.4: Cluster-observed class occurrence matrix in MAGNI-S2. Gray background color means a sample proportion greater than or equal to 20% out of all samples from each class.

| Cluster id | <i>C-Box</i> | <i>C-Bucket</i> | <i>C-LOF</i> | <i>C-LOL</i> | <i>V-A</i> | <i>V-G2</i> | <i>V-G3</i> | Total |
|--------------|--------------|-----------------|--------------|--------------|------------|-------------|-------------|-------|
| 1 | 62 | 3 | 25 | 29 | 73 | 43 | 8 | 243 |
| 2 | 6 | 1 | 14 | 13 | 53 | 33 | 3 | 123 |
| 3 | 35 | 91 | 22 | 31 | 87 | 30 | 21 | 317 |
| 4 | 77 | 37 | 34 | 30 | 60 | 36 | 17 | 291 |
| 5 | 20 | 21 | 73 | 59 | 106 | 37 | 11 | 327 |
| 6 | 2 | 54 | 13 | 16 | 47 | 19 | 7 | 158 |
| Total | 202 | 207 | 181 | 178 | 426 | 198 | 67 | 1459 |

One promising strategy to mitigate this issue is to augment the input state of the predictor using fine-grained representations, such as frame-based action labels. These provide temporally localized behavioral cues that may resolve ambiguity and enhance the predictive capacity of class-conditioned models. We explore this approach in detail in Chapter 5.

4.6 Conclusions and Outlook

In this chapter, we empirically demonstrate that integrating observable classes enhances the performance of trajectory prediction methods across diverse data settings. Specifically, we analyze how prior art in deep learning-based and pattern-based prediction can be adapted to consider class labels. For the deep learning methods, we modify models such as LSTM, Transformer-based single-output models, GANs, and VAEs to create efficient class-conditioned variants. Additionally, we evaluate the deep learning baseline against a pattern-based prediction approach, which uses Maps of Dynamics (CLIFF-LHMP) to encode motion trajectory patterns.

Our findings show that class-conditioned methods outperform their unconditional counterparts across most scenarios. Fig. 4.16 highlights the key quantitative outcomes of this work: (1) MoD-based predictors outperform single-output deep learning models in low-data regimes, regardless of class balance, (2) MoD-based predictors also surpass multiple-output deep learning models in imbalanced datasets, and (3) observable classes provide valuable cues for improving prediction accuracy in most cases. The improvement provided by observable classes stems from the fact that class conditioning reduces forecast uncertainty by focusing on specific agent behaviors rather than generalizing

across heterogeneous classes with complex and diverse motion trajectory patterns.

Consequently, choosing the most appropriate method depends heavily on the available data and the application requirements. For instance, in new environments with limited data or scenarios containing imbalanced class proportions, like vulnerable road users in automated driving, pattern-based methods offer advantages over deep learning models. Building on this work, Fig. 4.17 presents a *model selection tree* to guide model choice based on data characteristics (e.g., class balance and data volume) and application requirements (e.g., single- versus multiple-output predictions).

Despite the advantages, observable classes sometimes face two limitations: they rely on human annotation, which is labor-intensive, and may lack precision if a single class encompasses diverse motion patterns that overlap with other classes. For example, in the Stanford Drone Dataset (SDD), many cars are parked, leading to static trajectories that resemble those of pedestrians or stationary agents, which can impair model performance when using observable classes. Additionally, in the THÖR-MAGNI dataset, the *Carrier-Box* class may include various trajectory patterns, such as walking with a box, picking up the box (stationary behavior), or navigating without carrying an object, which overlaps with the *Visitors-Group 2* motion patterns. To address this lack of precision, we explore finer labels that we refer to as frame-based actions, which can better describe the agent’s trajectory in Chapter 5. In addition, we also study unsupervised trajectory and dynamics clustering to create more natural and informative class definitions. This approach seeks to improve the utility of class representations by identifying more meaningful and unambiguous trajectory groups, which can enhance model predictive performance. This next step requires learning unsupervised, data-driven trajectory classes and integrating them into the prediction task, a challenge we address in Chapter 6 and Chapter 7, respectively.

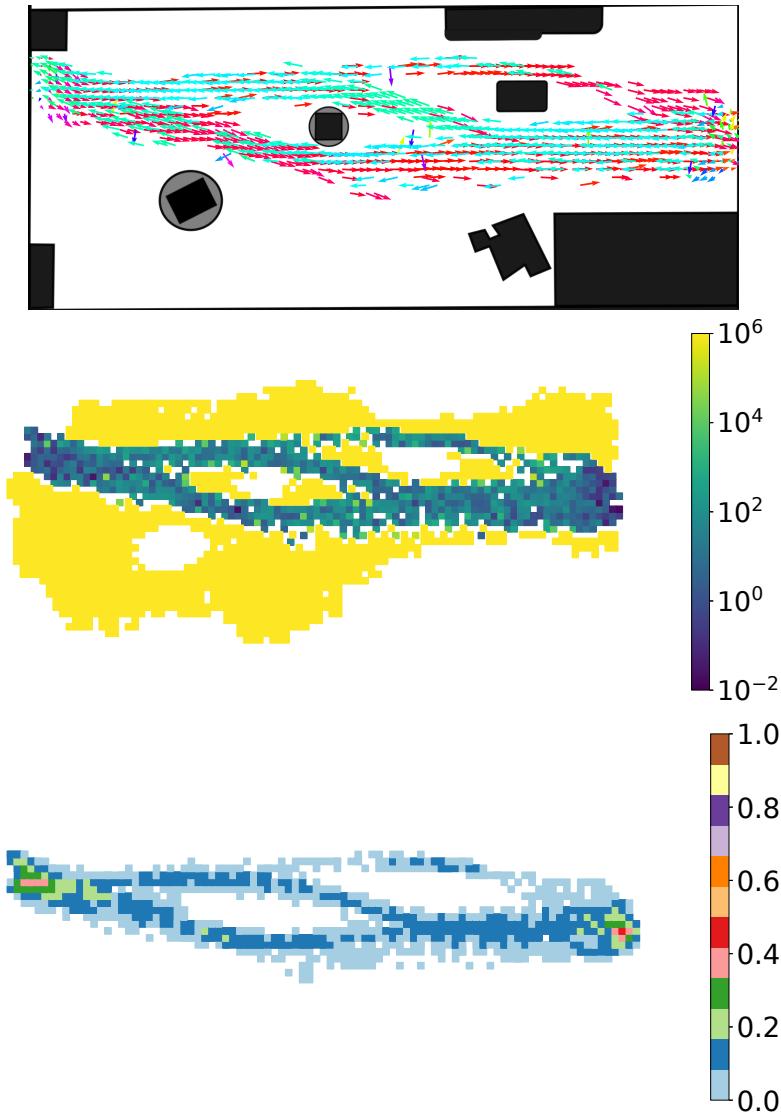


Figure 4.7: Motion patterns of *Carrier-Bucket* in THÖR-MAGNI, Scenario 2. Class-conditioned CLIFF-maps (**top row**) show that *Carrier-Bucket* has distinct and structured motion patterns. The KL divergence heatmap (**middle row**) quantifies the difference between the class-conditioned CLIFF-map and the general one. It also shows a clear divergence from the general motion patterns and higher motion intensity across specific paths (**bottom row**), resulting in a pronounced improvement in prediction accuracy from using class labels.

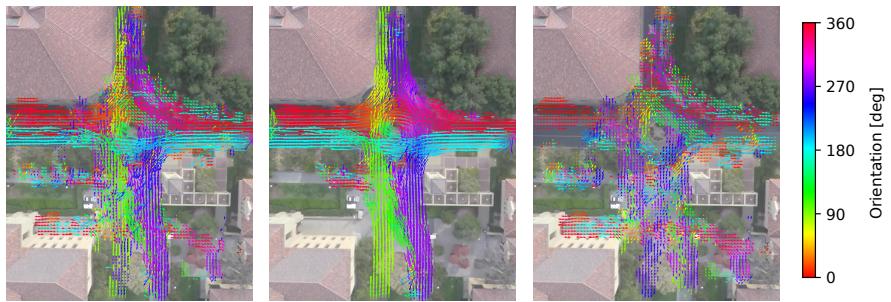


Figure 4.8: General and class-conditioned CLiFF-maps in the *DeathCircle* scene of the SDD. **Left:** all classes combined, **middle:** *Bicyclist* class, **right:** *Pedestrian* class. Colored arrows depict the mean speed (length) and direction (orientation) within the SWGMM of CLiFF-map, highlighting distinct motion patterns for different classes.

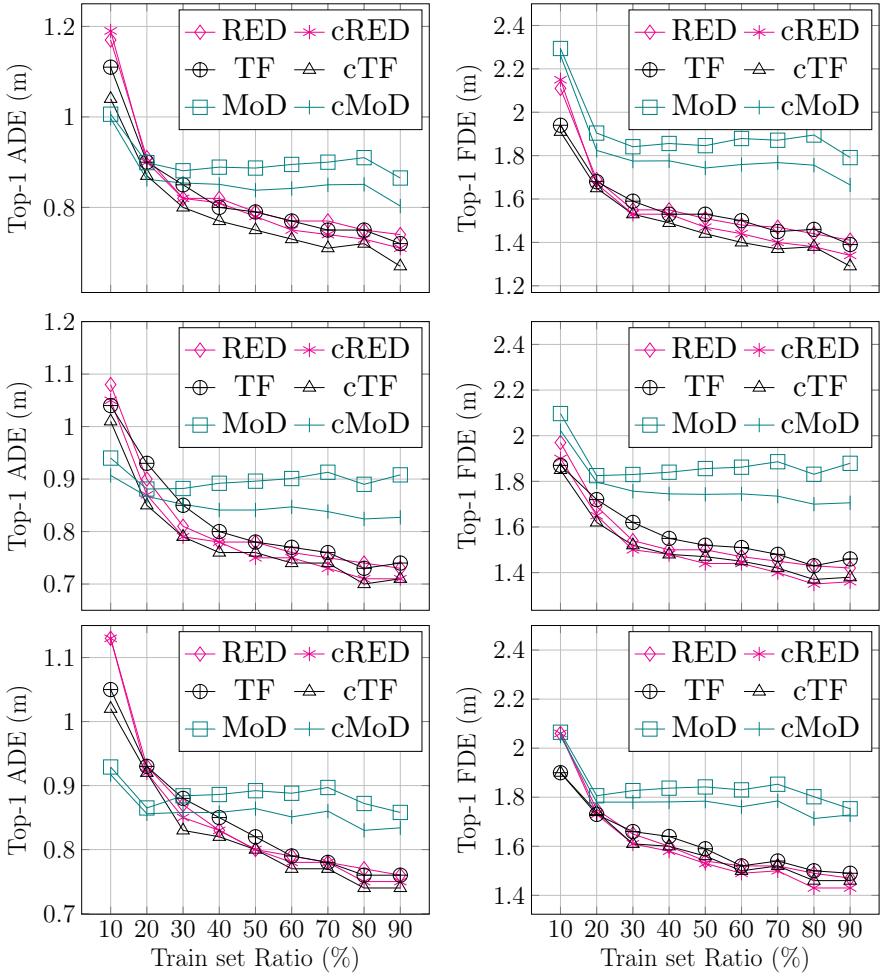


Figure 4.9: Top-1 ADE/FDE scores in THÖR-MAGNI Scenario 2 (**top row**), 3A (**middle row**) and 3B (**bottom row**). In this class-balanced setting, deep learning methods surpass MoD approaches for higher data regimes. However, MoD methods (MoD and cMoD) outperform deep learning methods for low data regimes and maintain stability even with reduced training data.

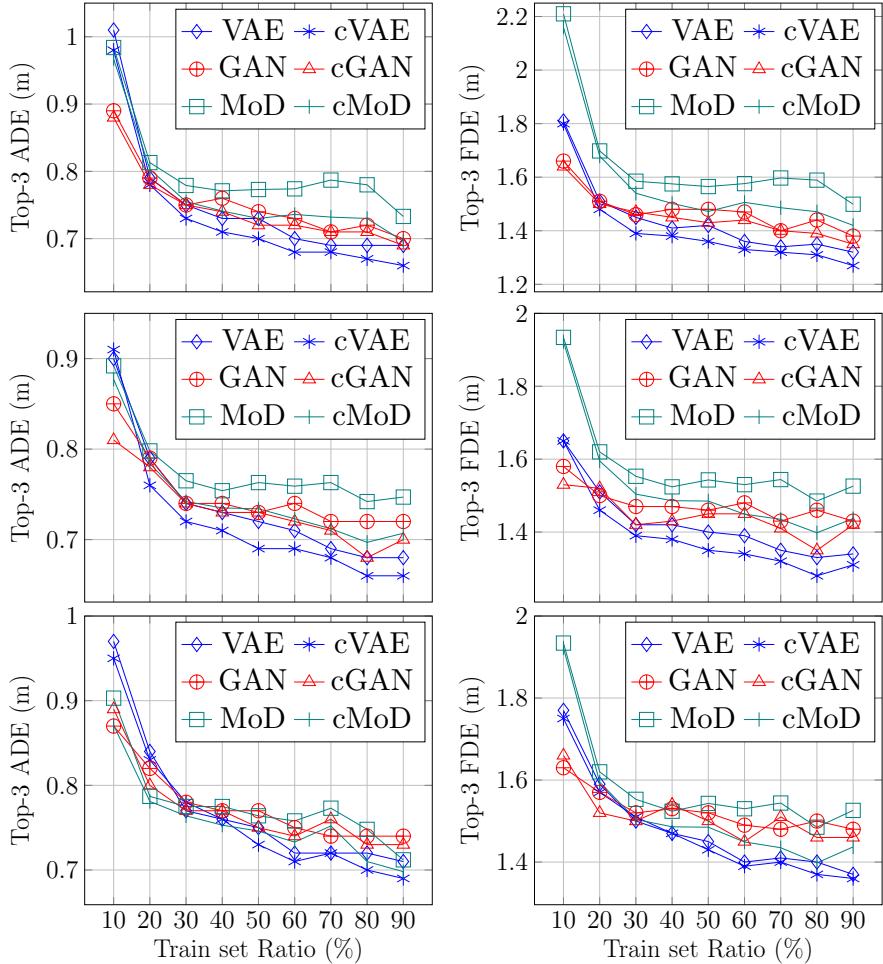


Figure 4.10: Top-3 ADE/FDE scores across THÖR-MAGNI Scenarios 2 (**top row**), 3A (**middle row**), and 3B (**bottom row**). In the class-balanced THÖR-MAGNI, deep generative methods excel over MoD.

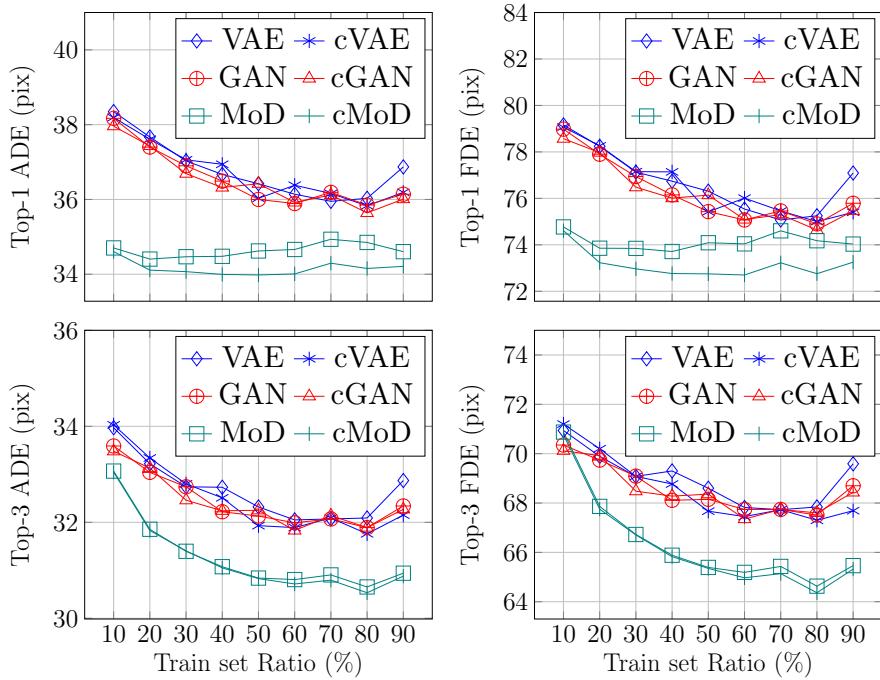


Figure 4.11: Top-1 (**top row**) and Top-3 (**bottom row**) ADE/FDE scores for SDD. In the imbalanced SDD, MoD methods outperform deep generative methods across all data regimes.

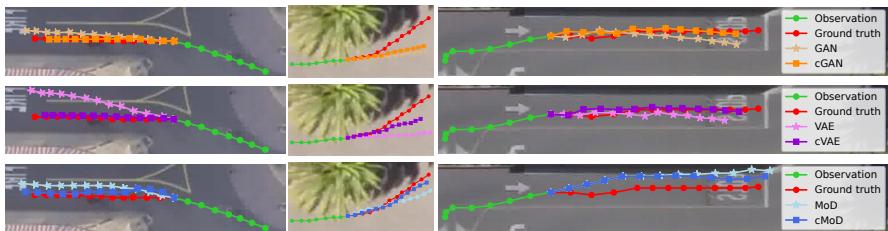


Figure 4.12: Prediction examples of *Byciclist* (**left**), *Pedestrian* (**middle**) and *Car* (**right**) in SDD with 4.8 s prediction horizon.

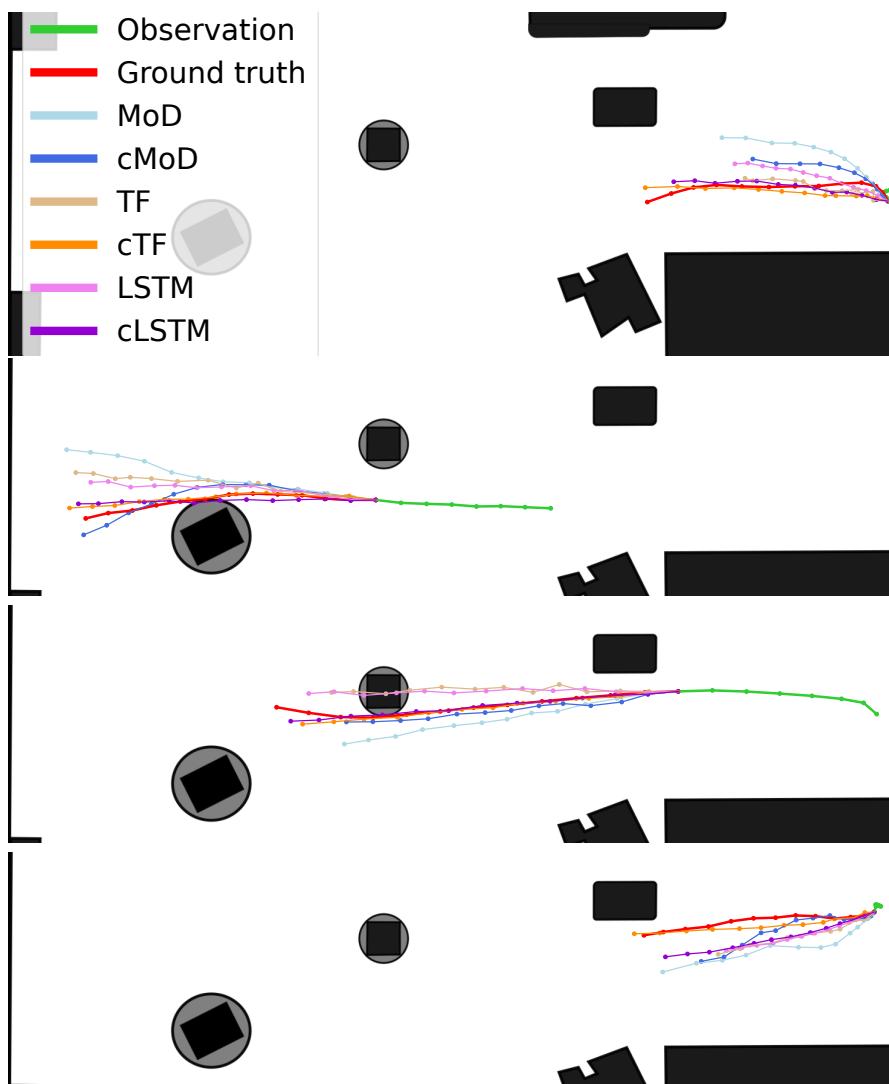


Figure 4.13: Prediction examples of *Carrier-Box* (**top left**), *Carrier-Bucket* (**top right**), *Visitors-Alone* (**bottom left**) and *Carrier-Large Object* (**bottom right**) in THÖR-MAGNI with 4.8 s prediction horizon.

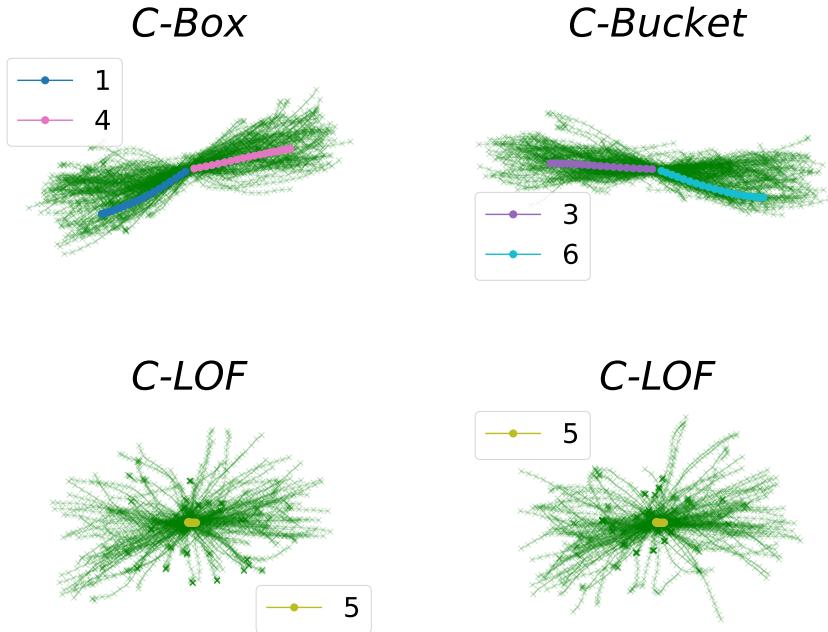


Figure 4.14: Original translated trajectories overlaid with the corresponding centroids of clusters representing more than 20% of the samples in MAGNI-S2 for each of the *Carriers* classes.

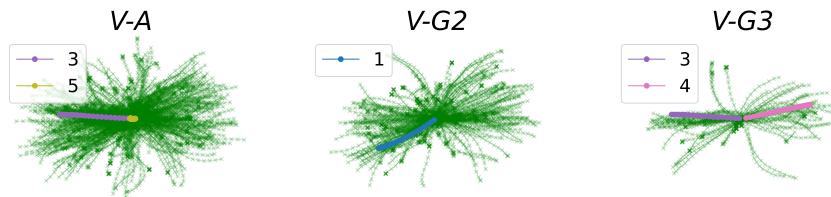


Figure 4.15: Original translated trajectories overlaid with the corresponding centroids of clusters representing more than 20% of the samples in MAGNI-S2 for the *Visitors* classes.

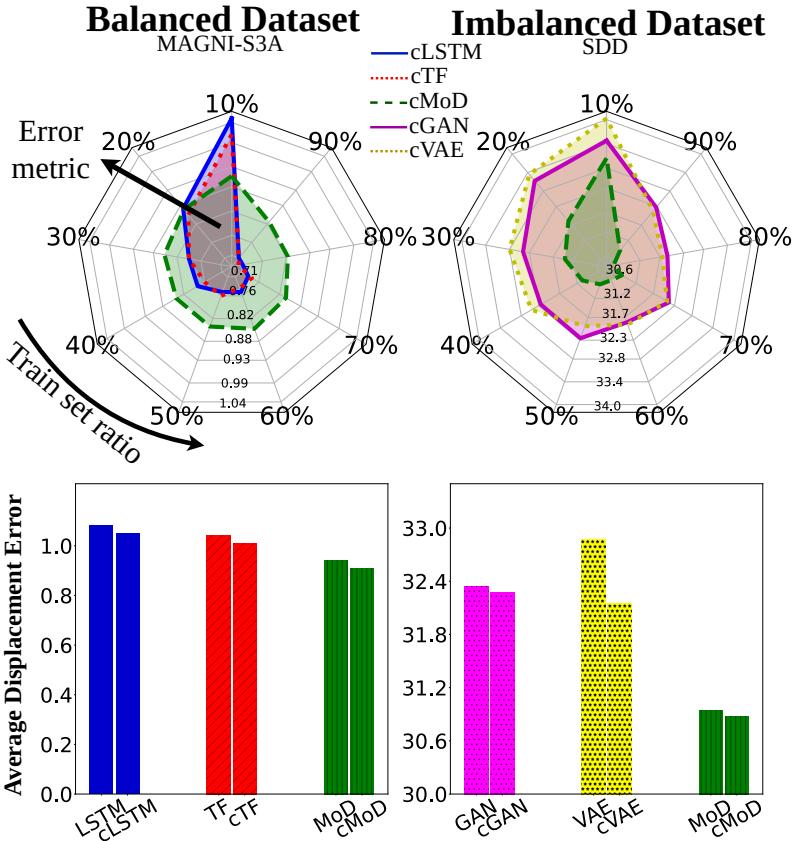


Figure 4.16: ADE of class-conditioned trajectory prediction methods across balanced and imbalanced datasets. **Top:** In balanced datasets, at low data regimes (10% train set ratio), the pattern-based method (cMoD) is more accurate than deep learning methods (**left**). In imbalanced datasets, cMoD is more accurate than deep generative models (**right**). **Bottom:** Class-conditioned methods (c^{***}) consistently outperform their unconditional counterparts across both datasets.

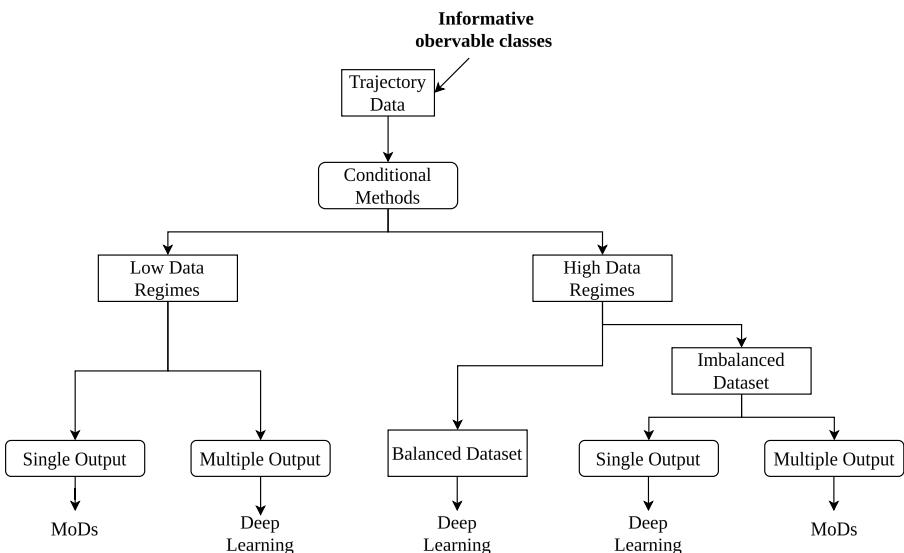


Figure 4.17: Class-conditioned trajectory prediction model selection tree: A structured decision tree for selecting trajectory prediction models based on data and model requirements. **Square boxes** represent data features, such as class balance or data amount. **Rounded boxes** correspond to model-specific attributes, including output determinism and class-conditioning suitability.

Chapter 5

THÖR-MAGNI *Act*: Fine-grained Human Actions in THÖR-MAGNI

Action is the foundational key to all success.

— Pablo Picasso

As demonstrated in the previous chapter, static observable classes provide powerful cues to enhance trajectory prediction. However, they sometimes lack precision in accurately representing the state of the agent acting in a context-rich, unconstrained environment, as a single class can encompass diverse behaviors. Also, those diverse behaviors can be shared between different classes (see Sec. 4.5.4). In order to resolve this ambiguity, we extend the state representation to consider fine-grained representations of agent states. This chapter introduces THÖR-MAGNI *Act*, an extension of the THÖR-MAGNI dataset, providing 8.3 hours of manually labeled participant actions derived from egocentric videos recorded via eye-tracking glasses. These actions, aligned with the provided THÖR-MAGNI motion cues, enrich the input state representation of a predictive system capable of anticipating human actions and trajectories in complex environments. We demonstrate the utility of THÖR-MAGNI *Act* for two tasks: action-conditioned trajectory prediction (A-TP) and joint action and trajectory prediction (TAP). To that end, we extend the Transformer-based model outlined in Chapter 4 to address these tasks. Our findings highlight the potential of THÖR-MAGNI *Act* as a valuable resource for developing advanced predictive models, facilitating improved human-robot interaction in complex, industrial-like environments. In addition, our action-conditioned trajectory prediction model outperforms the baselines across various input settings, demonstrating the effectiveness and generalizability of our predictive and meaningfulness of the corresponding action labels. Finally, our

multi-task learning model achieves strong performance in both trajectory and action prediction tasks, outperforming the baselines in trajectory prediction, matching single-task models in action prediction, and being more memory and energy-efficient.

5.1 Introduction

5.1.1 Motivation and Contributions

Until now, we have been focusing on observable classes that may affect the trajectory patterns of the dynamic agents in the scene. While these high-level attributes, such as roles or activities, effectively describe the overall behavior of an agent, they assign a uniform characterization to all trajectories at every time step associated with that agent. This uniform labeling introduces ambiguity, particularly in structured yet complex industrial environments, where robots interact with humans performing intricate tasks [9]. For example, in the THÖR-MAGNI dataset, described in Chapter 3, participants are assigned static roles representing their ongoing industrial activities (e.g., transporting objects; see Sec. 3.2.2). However, these high-level roles do not capture fine-grained sub-tasks such as walking, picking up an object, transporting it, or delivering it. As a result, conditioning trajectory predictors solely on static roles underperforms in scenarios requiring a more informed understanding of human motion. Moreover, datasets providing accurate, fine-grained labels for human motion and activities in industrial environments are scarce. Most existing datasets focus on social navigation in public spaces, where dominant behaviors include walking and standing [69, 30, 34, 78]. Therefore, human motion trajectory datasets with finer contextual information on the activities performed by the agents are needed.

To overcome these limitations, we present THÖR-MAGNI *Act*, an extension of the original dataset, which provides 8.3 hours of fine-grained actions derived from the first-person view videos of participants wearing eye-tracking glasses. Our THÖR-MAGNI *Act* is unique in aligning action labels with high-quality multi-modal first-person gaze and third-person motion capture data, as shown in Fig. 5.1. These labels enable the robot to anticipate not only long-term human motion trajectories but also actions, which is important to ensure more effective and informed human-robot interaction and cooperation.

To demonstrate the utility of THÖR-MAGNI *Act*, we present two prediction frameworks: (1) observable class- and action-conditioned trajectory prediction, extending the Transformer-based models described in Sec. 4.3, and (2) joint prediction of future trajectory and corresponding actions via multi-task learning [104]. Our results show that incorporating action labels improves the performance of these predictive models. THÖR-MAGNI *Act* and the corre-

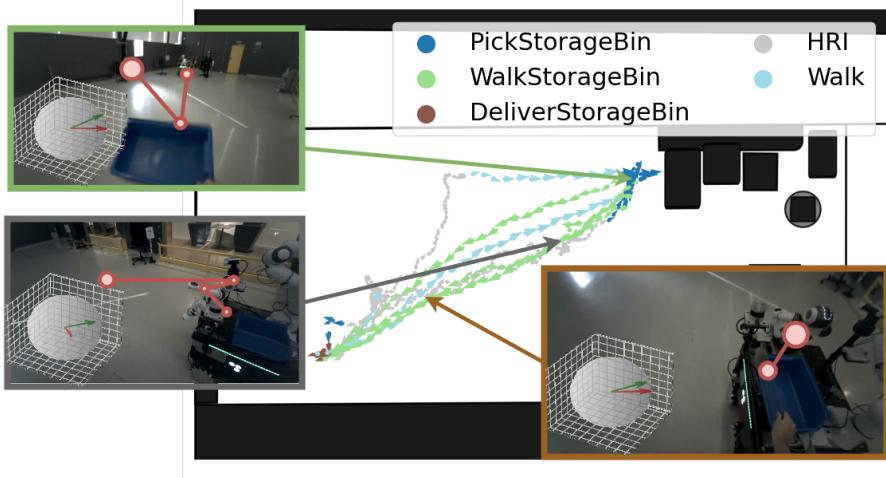


Figure 5.1: Action annotations for a 4-minute recording of a person carrying storage bins while interacting with a mobile robot, synchronized with the motion capture data. Inset images display snapshots from gaze overlaid videos, featuring visualizations of head orientation vector (red) and gaze vector (green). The length of the arrows on the map denotes the velocity magnitude.

sponding scripts are stored in a publicly accessible repository¹. Documentation on how to use and visualize the dataset can be found in the same repository.

5.1.2 Outline

In Sec. 5.2, we describe the actions in the THÖR-MAGNI *Act* dataset and present statistical insights on the provided data. In Sec. 5.3, we formalize two tasks enabled by our dataset: action-conditioned trajectory prediction (ATP) and multi-task learning for simultaneous trajectory and action prediction (TAP). In the same section, we also introduce the corresponding prediction methods. In Sec. 5.4, we describe the experimental setups for evaluating our methods, highlighting the target scenarios and the evaluation setup. Sec. 5.5 presents quantitative and qualitative results for trajectory and action prediction. Finally, in Sec. 5.6, we summarize the dataset’s key findings and contributions.

¹<https://github.com/tmralteida/thor-magni-actions>

Table 5.1: Amount of eye-tracking and trajectory data recorded per human role or observable class.

| Observable Class | Eye-tracking (min.) | Trajectory data (min.) |
|--------------------------------|---------------------|------------------------|
| <i>Visitors-Alone</i> | 108 | 392 |
| <i>Visitors-Group 2</i> | 124 | 344 |
| <i>Visitors-Group 3</i> | 52 | 168 |
| <i>Visitors-Alone HRI</i> | 64 | 112 |
| <i>Carrier-Bucket</i> | 32 | 96 |
| <i>Carrier-Box</i> | 60 | 96 |
| <i>Carrier-Large Object</i> | 92 | 192 |
| <i>Carrier-Storage Bin HRI</i> | 16 | 16 |
| Total | 548 | 1416 |

5.2 THÖR-MAGNI *Act* Dataset

The observable classes in THÖR-MAGNI represent high-level roles assigned to participants for the duration of a 4-minute recording session, such as: *Carrier-Box*, *Carrier-Bucket*, *Carrier-Large Object*, *Visitors-Alone*, *Visitors-Group*, and *Visitors-Alone HRI* (see Sec. 3.2.2). During each session, one to three participants wore eye-tracking glasses to capture egocentric video data, providing a first-person perspective of their actions and movements. Tab. 5.1 presents a detailed summary of the amount of eye-tracking data recorded for each observable class in the original THÖR-MAGNI dataset. THÖR-MAGNI *Act* builds upon this foundation by introducing manually annotated action labels derived from the recorded egocentric videos. These fine-grained action annotations offer a more detailed description of human behavior within the high-level roles already defined in THÖR-MAGNI. In this section, we describe the action annotation process in detail and provide statistical insights into the dataset, highlighting the alignment between trajectory data and the newly introduced action labels.

5.2.1 Action Annotations

We annotated actions in the entire THÖR-MAGNI dataset using the 8.3 hours of egocentric videos (see Tab. 5.1) and the “Event Marker” feature in the eye-tracking software [92]. We placed Event markers at the initial fixations indicating action transitions, such as reaching for objects or bending to deliver items. In ambiguous transitions, where hand visibility was insufficient, we selected subsequent fixations within the ongoing activity as reference points. Additional data from the eye tracker, such as IMU readings and audio cues,

were used to improve the precision of annotations in these ambiguous cases. This annotation process was manually curated to ensure high-quality labels.

From the annotation process, we define a set \mathcal{A} of 14 unique action labels that provide a detailed representation of participant activities within the dataset. These actions are aligned with existing observable classes in THÖR-MAGNI and capture various stages of goal-oriented tasks and interactions:

- *Walk*: Walking between designated goal points.
- *DrawCard*: Drawing a card at a goal point.
- *ObserveCardDraw*: Observing another participant drawing a card at a goal point.
- *WalkLO*: Transporting a large object.
- *PickBucket*: Picking up a bucket.
- *WalkBucket*: Transporting a bucket between goal points.
- *DeliverBucket*: Dropping a bucket at a goal point.
- *PickBox*: Picking up a box at a goal point.
- *WalkBox*: Transporting a box between goal points.
- *DeliverBox*: Delivering a box at a goal point.
- *PickStorageBin*: Picking up a storage bin.
- *WalkStorageBin*: Transporting a storage bin between goal points.
- *DeliverStorageBin*: Dropping a storage bin at a goal point.
- *HRI*: Interacting with the mobile robot.

Each observable class is associated with specific actions, while some actions are shared across different classes (see Fig. 5.2 for an overview). An observable class is constant in all trajectories of a particular agent, whereas an action may change at every time step. Consequently, this data extension provides finer labeling of internal factors (e.g., goal-driven actions) that can influence human motion. In particular, the actions *Walk* and *DrawCard* are shared across multiple observable classes, indicating that trajectories involving these actions are likely to have similar characteristics, even when performed by agents of different observable classes.

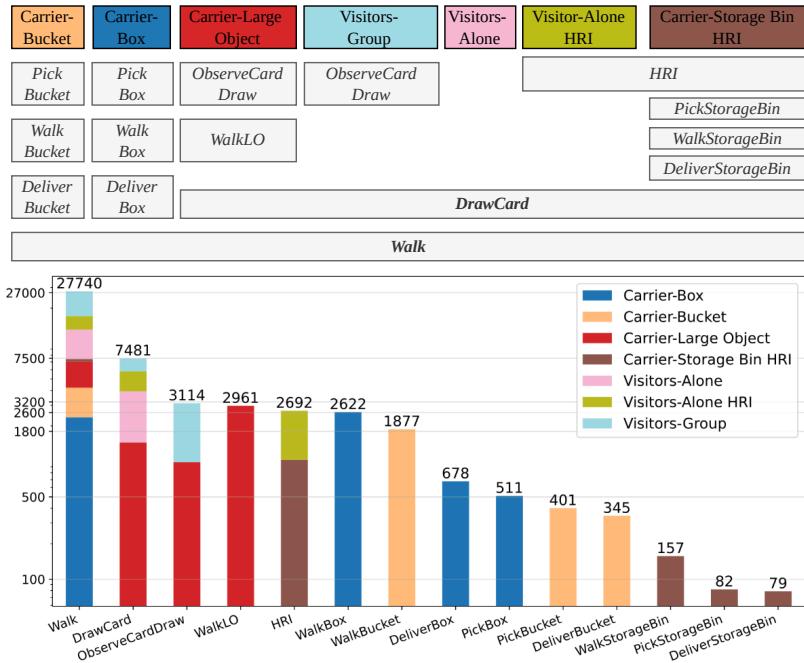


Figure 5.2: **Top:** Observable class-actions mapping. Grey boxes denote actions, and colored boxes represent the observable classes. **Bottom:** Distribution of actions in *log-scale* sorted by descending order, with colors indicating observable classes.

5.2.2 Dataset Statistics

We computed the THÖR-MAGNI *Act* statistics for non-overlapping 8-second trajectory segments, in line with common trajectory prediction benchmarks [52] and the analysis conducted in Sec. 3.4 for the THÖR-MAGNI dataset. Fig. 5.2 bottom presents the distribution of actions in *log-scale*, along with their representation across different observable classes. Although the dataset’s actions follow a long-tailed distribution, it includes novel action labels specific to human tasks and mobile robot interactions, setting it apart from existing social navigation datasets. Along with motion cues and gaze vectors, these labels support research on egocentric action prediction models from visual input and gaze pattern analysis.

Fig. 5.3 presents the average and standard deviation of acceleration, velocity, and navigation distance of motion in each action class, along with the corresponding global metrics (aggregated across all 8-second segments). For acceleration, static actions such as picking up or delivering an object result

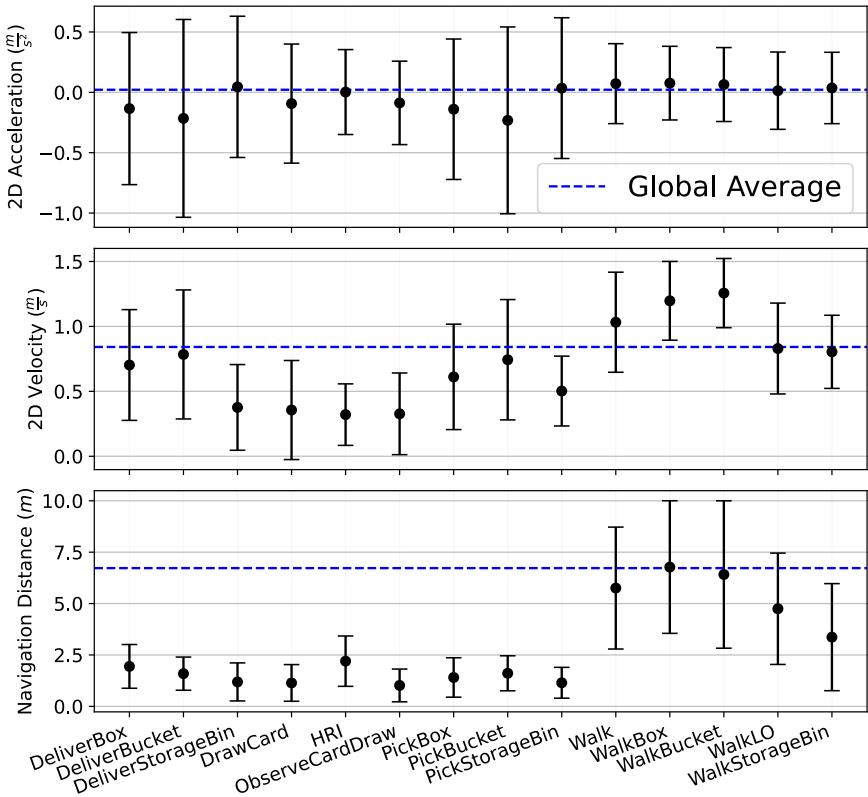


Figure 5.3: **Top:** 2D acceleration (mean \pm one standard deviation), where values near zero indicate constant velocity. **Middle:** 2D velocity (mean \pm one standard deviation), where values near zero correspond to static actions. **Bottom:** navigation distance (mean \pm one standard deviation), where values near zero indicate static actions and higher values reflect walking actions.

in small negative accelerations, while walking actions generally show constant velocities or small positive accelerations. Consequently, in terms of velocity, static actions fall below the global average, whereas actions like *WalkBox* and *WalkBucket* involve higher velocities compared to *WalkLO* or *WalkStorageBin*, where participants transport a large object in groups of two and move alongside the robot, respectively. Finally, distance correlates with the acceleration and velocity trends, highlighting distinct actions and further demonstrating the diversity and complexity of our dataset.

5.3 Trajectory and Action Prediction

In this section, we introduce two examples of tasks this dataset can be used for: *action-conditioned trajectory prediction* (A-TP) and multi-task learning for *joint trajectory and action prediction* (TAP), along with the respective proposed models. The observed horizon (O) spans 3.2 s (8 time steps), while the prediction horizon (L) extends to 4.8 s (12 time steps). The observed tracklets are denoted as $\mathbf{S} = (\mathbf{s}_t)_{t=1}^O$, where the states \mathbf{s}_t comprise 2D positions, velocities, and the corresponding action a , represented as $\mathbf{s}_t = (x, y, \dot{x}, \dot{y}, a)$. The *future* of an observed tracklet $\mathbf{Y}_{\mathbf{S}}$ consists of 2D velocities, $\mathbf{Y}_{\mathbf{S}} = ((\dot{x}_t, \dot{y}_t))_{t=O+1}^{T_P}$ of length $L = 12$, which are subsequently converted into future positions $\mathbf{P}_{\mathbf{S}}$. The future sequence of actions temporally aligned with $\mathbf{Y}_{\mathbf{S}}$ is denoted by $\mathbf{a}_{\mathbf{S}} = (a_t)_{t=O+1}^{T_P}$, $a_t \in \mathcal{A}$.

5.3.1 Action-conditioned Trajectory Prediction

The goal is to predict the future of a tracklet conditioned on the observed actions (part of the input state) and observable class, $\psi_{\text{A-TP}} : (\mathbf{S}_k, C_k) \mapsto \mathbf{Y}_{\mathbf{S}_k}$, where $\mathbf{Y}_{\mathbf{S}_k}$ is the future corresponding to the observed tracklet \mathbf{S}_k . The training data for this task, $\{(\mathbf{S}_k, C_k, \mathbf{P}_{\mathbf{S}_k})\}_k$, consists of triplets of observed tracklets including actions, ground truth observable class labels, and ground truth future positions.

The $\psi_{\text{A-TP}}$ model has an encoder-decoder structure as in [52, 27] and Chapter 4. The encoder Enc , a Transformer-based encoder [93], processes the result of an embedding mapping (a single-hidden layer multilayer perceptron or MLP). The encoded features are then concatenated with the observable class embeddings and processed through the decoder network Dec_T (a two-hidden layer MLP) to generate the future sequence of velocities, $\mathbf{Y}_{\mathbf{S}}$. Fig. 5.4, excluding the yellow branch, depicts the graphical representation of $\psi_{\text{A-TP}}$. The blue dotted arrow in the figure indicates the baseline model, which operates without observable class conditioning or actions in \mathbf{S} . We train $\psi_{\text{A-TP}}$ with the MSE loss given by Eq. (4.4).

5.3.2 Trajectory and Action Prediction

The goal is to predict the future of a tracklet and the corresponding actions, $\psi_{\text{TAP}} : (\mathbf{S}_k, C_k) \mapsto (\mathbf{Y}_{\mathbf{S}_k}, \mathbf{a}_{\mathbf{S}_k})$, where $\mathbf{Y}_{\mathbf{S}_k}$ is the future tracklet and $\mathbf{a}_{\mathbf{S}_k}$ the future sequence of actions corresponding to the observed tracklet \mathbf{S}_k . The training data for this task, $\{(\mathbf{S}_k, C_k, \mathbf{P}_{\mathbf{S}_k}, \mathbf{a}_{\mathbf{S}_k})\}_k$, consists of quadruples of observed tracklets including actions, ground truth observable class labels, ground truth future positions, and the sequence of actions.

The ψ_{TAP} model is similar to the previously described $\psi_{\text{A-TP}}$ model (see Sec. 5.3.1). The key difference is the additional decoder, Dec_A , which shares the same network configuration as Dec_T . This decoder generates probabilities for

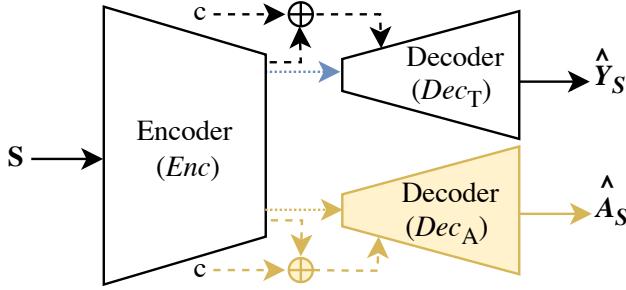


Figure 5.4: Action-conditioned models and multi-task learning methods (additional **yellow** branch). **Dashed** arrows indicate methods using observable class, while **dotted** arrows represent baseline models where S excludes actions in the trajectory prediction task.

the sequence of actions at each future time step, denoted as $\mathbf{A}_S^{L \times N_A}$, where $N_A = |\mathcal{A}|$. The final sequence of actions \mathbf{a}_S is determined by applying the *argmax* operator to these probabilities. Fig. 5.4, including the yellow branch, depicts the graphical representation of ψ_{TAP} , whose baseline consists of two models tailored to each task. We train ψ_{TAP} using a weighted loss function that combines trajectory prediction (as defined in Eq. (4.4)) and sequence of actions prediction, as follows:

$$\mathcal{L}_{\text{TAP}}(\mathbf{P}_S, \hat{\mathbf{P}}_S, \mathbf{a}_S, \hat{\mathbf{a}}_S) = \mathcal{L}_{\text{MSE}}(\mathbf{P}_S, \hat{\mathbf{P}}_S) + \lambda \mathcal{L}_A(\mathbf{a}_S, \hat{\mathbf{a}}_S), \quad (5.1)$$

where λ is a weighting factor that balances the contribution of the action prediction term, \mathcal{L}_A , relative to the trajectory prediction term, \mathcal{L}_{MSE} . The action prediction loss, \mathcal{L}_A , is defined as the cross-entropy loss:

$$\mathcal{L}_A(\mathbf{a}_S, \hat{\mathbf{a}}_S) = -\frac{1}{L} \sum_{j=O+1}^{T_P} \sum_{m=1}^{N_A} a_m^j \log(\hat{a}_m^j), \quad (5.2)$$

where \hat{a}_m^j is the predicted probability for class m at time step j , and a_m^j is a binary indicator of the ground truth for class m at time step j . In our experiments, we tested $\lambda = 1$.

5.4 Experiments

5.4.1 Target Scenarios

For our analysis, we merged Scenarios 1 (humans navigating freely), 2 (human task-oriented roles and static robot), and 3 (human task-oriented roles

and moving robot) data from THÖR-MAGNI, as these scenarios encompass a more diverse set of observable classes and, consequently, a broader vocabulary of action classes. These scenarios comprise a total of 5 observable classes: *Carrier-Box*, *Carrier-Bucket*, *Carrier-Large Object*, *Visitors-Alone*, and *Visitors-Group*. In addition, they include 10 action classes: *DrawCard*, *Walk*, *WalkLO*, *PickBucket*, *WalkBucket*, *DeliverBucket*, *ObserveCardDraw*, *PickBox*, *WalkBox*, and *DeliverBox*. In total, the dataset contains 1909 trajectories of 8 seconds.

5.4.2 Evaluation Setup

To evaluate the proposed models, we employ 5-fold cross-validation. In the prediction results, we use Top-1 ADE and FDE in meters, as defined in Eqs. (1.1) and (1.2). In the action prediction results, we also show the accuracy (ACC) and F1 score (F1), as defined in Eqs. (1.3) and (1.4). We compute all metrics' mean and standard deviation across the validation folds to ensure robust evaluation.

5.5 Results

5.5.1 Quantitative Results

Tab. 5.2 shows the results for the A-TP task with various cue settings. The results show that incorporating action information (fourth row) improves the prediction accuracy compared to the baseline model (first row), with further gains when combined with observable class information (last row). To evaluate the generalizability of these cues, we also report results using translated and rotated input trajectories aligned with the x-axis, shown in the fifth column. This transformation removes environment-specific dependencies, yielding a more generalizable input representation (see Sec. 1.2.1). Such layout-agnostic settings are particularly relevant in real-world applications, where robots operate in unseen environments or cold-start scenarios as discussed in Chapter 4. Importantly, even under this more generalizable configuration, action conditioning outperforms other methods, and the performance gap becomes more pronounced compared to models using raw positions (fourth column). In fact, action cues prove more informative in layout-independent conditions than high-level observable classes, achieving superior prediction accuracy. These findings suggest that fine-grained action annotations offer robust contextual signals that enhance the representation of context-agnostic input states. These results demonstrate that actions serve as valuable contextual cues, comprising fine-grained events and enhancing the trajectory prediction process with additional, informative descriptions of current activities.

Tab. 5.3 shows the results on the TAP task. We show the best baseline for action prediction where $\mathbf{s}_t = (x, y, \dot{x}, \dot{y}, a)$. The results show that observed ac-

Table 5.2: Action-conditioned trajectory prediction results for raw positions in the input state, with bold values indicating superior performance of our agent and action class-aware models compared to the baseline.

| Model | Agent Class | Actions Class | ADE FDE | ADE* FDE* | Number Parameters (K) |
|----------|-------------|---------------|--------------------------------------|--------------------------------------|-----------------------|
| BASELINE | | | 0.71±0.03 1.37±0.05 | 0.89±0.04 1.88±0.09 | 36.7 |
| | ✓ | | 0.68±0.03 1.30±0.07 | 0.86±0.03 1.84±0.07 | 38.1 |
| OURS | | ✓ | 0.69±0.03 1.31±0.07 | 0.82±0.03 1.73±0.08 | 37.3 |
| | ✓ | ✓ | 0.67±0.03 1.28±0.07 | 0.81±0.03 1.71±0.08 | 38.7 |

* Results with translated and rotated positions.

Table 5.3: Comparative multi-task learning results, with bold values showing superior performance.

| Model | Agent Class | Actions Class | ADE FDE | ACC F1 | Number Parameters (K) |
|-----------|-------------|---------------|--------------------------------------|--------------------------------------|-----------------------|
| BASELINES | | | 0.71±0.03 1.37±0.05 | 0.85±0.01 0.85±0.01 | 36.7+42.6 |
| | ✓ | | 0.68±0.04 1.29±0.08 | 0.62±0.02 0.61±0.02 | 46.3 |
| OURS | | ✓ | 0.70±0.03 1.33±0.07 | 0.83±0.01 0.83±0.01 | 43.3 |
| | ✓ | ✓ | 0.70±0.04 1.32±0.08 | 0.85±0.01 0.85±0.01 | 46.8 |

tion sequences are crucial for strong performance in action prediction (second row versus third and fourth rows). Our best approach can perform strongly in trajectory and action prediction simultaneously, outperforming baselines in trajectory prediction and matching single-task models in action prediction (last row). Also, our method (46.8K) is more efficient than the baselines (36.7K+42.6K). In summary, these results highlight that THÖR-MAGNI *Act* enables the development of novel predictive systems capable of performing multi-task predictions for both trajectories and actions. Furthermore, the proposed method is an effective and efficient foundational prototype for such systems.



Figure 5.5: Prediction examples for *Carrier-Box* in Scenario 3, for our multi-task learning framework (“MTL-OURS”, **left**) for joint trajectory and action prediction, and our action-conditioned trajectory prediction model (“ACT-OURS”, **right**), with a 4.8 s prediction horizon.

5.5.2 Qualitative Results

Fig. 5.5 illustrates scenarios where incorporating actions improves prediction accuracy. For instance, in the case of the “picking up a box” behavior, the multi-task learning approach (MTL) reduces the ADE/FDE errors from 0.69/1.00 to 0.23/0.11, with only a single future action misprediction. Similarly, action-conditioned predictions leverage observed action sequences to reduce trajectory errors for the “dropping a box” behavior. These examples highlight cases where fine-grained actions change throughout the trajectory. In such scenarios, either incorporating observed actions or accurately predicting future actions provides informative and meaningful contextual information, leading to substantial improvements in trajectory prediction accuracy.

5.6 Conclusions and Outlook

The interplay between complex activities, actions, and locomotion dynamics of people and other agents still needs to be explored, particularly in industrial scenarios. This research gap can be addressed with comprehensive datasets that capture the relationship between actions and motion trajectories. Our work introduces the THÖR-MAGNI *Act* dataset to align action labels with diverse human trajectory cues. These cues, including position data, head orientation, gaze, and semantic attributes, provide a rich description of human motion in industrial settings. In this work, we also developed efficient and accurate models for two key applications enabled by THÖR-MAGNI *Act*: (1) action-conditioned trajectory prediction (A-TP) and (2) joint action and trajectory prediction (TAP). We have shown that the proposed models can leverage the rich contextual information the dataset provides to improve prediction accuracy in both A-TP and TAP tasks. The results demonstrate that augmenting the input state representation with action labels can substantially enhance the performance of trajectory prediction models, even in layout-agnostic scenarios.

In addition, the proposed multi-task learning framework can effectively learn to predict both future trajectories and actions simultaneously, achieving better performance in the trajectory prediction task and competitive performance in the action prediction task than single-task baselines. THÖR-MAGNI *Act* with the diverse annotation classes pave the way for future research in human trajectory modeling based on rich contextual cues. However, from a practical perspective, these contextual cues often depend on external perception algorithms, which can introduce errors and negatively impact prediction accuracy. To address this limitation, data-driven trajectory classes offer a robust alternative by providing representations derived directly from motion patterns, independent of class or action detection errors. This approach is introduced in Chapter 6, where we propose methods for learning such classes. Subsequently, in Chapter 7, we integrate the learned representations into prediction frameworks to improve forecasting performance.

Chapter 6

Learning Data-driven Classes

(...) he thought about the impermanence of life, the transience of things, the ephemerality of being; before him, existence flowed like a breath, always changing, everything changing at every moment and nothing ever being the same again.

— José Rodrigues dos Santos, *O Sétimo Selo*

Although observable classes are valuable cues for trajectory prediction, they face two limitations. First, human-defined semantic labels (e.g., *Carrier-Box* and *Carrier-Bucket* in Chapter 4) may not always align consistently with actual motion patterns. This misalignment arises from the static nature of observable classes, which assign a single label to all trajectories of a given agent, potentially encompassing diverse and ambiguous motion behaviors. Second, even when actions enhance semantic granularity (Chapter 5), both observable classes and actions ultimately depend on external perception stacks. Errors in sensor data or downstream detection pipelines can propagate through the system and impair trajectory prediction performance.

This chapter explores an alternative approach based on learning data-driven trajectory classes to address these challenges. These classes are formed by clustering trajectory data directly, bypassing predefined semantic labels, and grouping trajectories based on shared patterns instead. The intuition is that data-driven classes enhance class representation and, consequently, downstream trajectory predictions.

Data-driven classes for the prediction task should capture the underlying structure of motion patterns, including future intent and latent semantics. A natural alternative to observable classes is to cluster trajectories based on their observed states, intuitively assuming that similar observed sequences will continue with similar future movements. However, observed states alone do not provide sufficient information to capture future intent, as they only reflect past motion. An alternative is to cluster the future or entire sequence of trajectory states, which can hold important patterns for the prediction task. Therefore, a key question we address in this chapter is whether clustering based on future

or full trajectory states yields more informative classes than those based solely on observed states.

To study the impact of data-driven classes on trajectory prediction, we conduct a performance analysis. We define predictive models conditioned on clusters derived from trajectory states based on traditional clustering techniques, such as K-means. We show that only clusters including future information enhance trajectory prediction compared to clusters based on observed trajectory states. While impractical at inference time as they depend on future information, these models are theoretical baselines indicating the informative content of trajectory patterns. We also explore alternative input representations (i.e., displacement vectors, normalized trajectories, statistical descriptors) that impact cluster formation. Specifically, we find that statistical descriptors outperform the other representations in final displacement error. From the clustering algorithm perspective, K-means applies the Euclidean distance in flattened inputs, disregarding sequential patterns that are similar in dynamics but misaligned in time. Thus, novel clustering methods are required to study, as traditional classical approaches present limitations.

An alternative to K-means is Time-Series K-Means, which uses time-series distance metrics to capture misaligned temporal patterns. Although theoretically better suited for time-series, Time-Series K-Means is inefficient due to the use of the Dynamic Time Warping distance. To overcome these limitations, we present a method for finding data-driven trajectory classes based on a deep generative framework. We propose *Self-Conditioned GAN*, a multi-task learning framework that simultaneously learns to cluster trajectories in a deep feature space and generate realistic trajectory samples. We benchmark our approach against K-means and Time-Series K-Means and assess the quality and utility of the resulting clusters. As a result, we show that our Self-Conditioned GAN captures more generalizable clusters by outperforming traditional clustering techniques in data distribution drift settings.

6.1 Introduction

6.1.1 Motivation and Contributions

Observable classes, while effective in many settings, are static: a fixed class is assigned to each agent’s trajectories regardless of the possible diversity of motion trajectory patterns. As a result, these classes may fail to distinguish between distinct motion patterns that occur across different contexts. Moreover, similar trajectories from different agents may be grouped under separate observable classes, leading to incoherent or ambiguous representations (Sec. 4.5.4). Although extending the input with actions can alleviate some of this ambiguity, actions still rely on perceptual pipelines that may produce inaccurate estimations.

To overcome these challenges, this chapter proposes a shift from predefined observable classes to data-driven classes, i.e., groupings derived directly from trajectory data. Contrary to observable classes, data-driven classes (1) are not defined by human labels but rather emerge from the data itself, capturing the underlying structure and dynamics of the trajectories; (2) can also be more robust to noise and errors in the input data, as they are based on the actual motion patterns rather than relying on potentially flawed observable class or action detections; (3) operate at a finer level of granularity by clustering trajectories rather than grouping agents' trajectories according to the agent type, role, or activity.

In this thesis, we define data-driven classes as clusters of trajectories with similar spatio-temporal characteristics obtained through unsupervised learning techniques that focus uniquely on the intrinsic structure of motion. However, clustering sequential trajectory data is not trivial due to its high dimensionality, temporal dependencies, and non-linear structure. Traditional clustering methods, such as K-means [66], require reshaping datasets of multidimensional time-series into 2D matrices, where, in our case, each row represents a trajectory, and the columns correspond to the flattened time-series representation. K-means uses the Euclidean distance in this flattened space, which fails to account for temporal misalignments of trajectory patterns. Time-Series K-Means (TS K-means) [91] partially addresses this issue by using time-series distance metrics such as Dynamic Time Warping (DTW), but its high computational cost limits its applicability in large-scale or real-time scenarios. These limitations highlight the need for advanced, unsupervised methods that automatically detect trajectory classes while integrating seamlessly into trajectory prediction frameworks. Such unsupervised methods should leverage the complexity and richness of the input data through an effective input representation to enhance prediction accuracy without compromising computational efficiency.

The choice of input features influences the clustering process, as different representations capture different characteristics of the data and thus yield distinct clustering outcomes. In this chapter, we evaluate three types of input representations, each motivated by its ability to encode specific motion properties: (1) displacement vectors, which capture short-term velocity and directionality and are expected to group trajectories based on local kinematic behaviors, (2) normalized trajectories, which abstract away from spatial context and enable grouping similar trajectories independent of orientation, useful in environment-invariant prediction settings, and (3) statistical features, which provide a compact matrix-based summary of a trajectory using descriptors such as mean velocity, path efficiency, and turning behavior, effectively aggregating several motion aspects into a single feature space. Beyond feature representations, the trajectory segment used for clustering also affects the informativeness and applicability of the resulting clusters for downstream tasks like prediction. We investigate three strategies for cluster derivation: (1)

observation-driven clustering, which uses only the observed segment of the trajectory and is naturally deployable within prediction tasks, as the observation can be directly mapped to the corresponding cluster class at inference time (similar to observable classes), (2) future-driven clustering, which explores the future portion of the trajectory to uncover modes of future behavior, and (3) full-driven clustering, which uses the complete trajectory to capture both past and future motion patterns, potentially improving the correspondence between observed segments and cluster assignments by accounting for their full temporal context. In this chapter, we comprehensively study the combination of different input features and the three segments of the trajectory (i.e., observation, future, full), allowing us to explore how the nature and timing of input data influence the formation of trajectory classes and their effectiveness when integrated into prediction frameworks. Our experiments show that only future- and full-driven clustering yield informative classes for the prediction task, as these segments contain information about the unseen future. On the other hand, observation-driven clusters overlap with the input that is already accessible to the predictor. Clusters derived from future information enable the construction of models that represent theoretically achievable performances under privileged future-driven cluster knowledge. Though impractical for deployment, these predictors are analytically valuable for quantifying model limitations and informing the design of other effective prediction systems. We leverage the insights from the analysis of predictors conditioned on future trajectory segments in Chapter 7 to develop practical predictors that benefit from the structure uncovered by future- and full-driven trajectory classes.

To address current traditional clustering methods' shortcomings, this chapter proposes the Self-Conditioned GAN (SC GAN), a deep generative framework that combines trajectory clustering and generation in a unified architecture. SC GAN learns a 2D deep feature space in which trajectory embeddings are clustered while simultaneously generating trajectory samples conditioned on the discovered clusters. This multi-task setup ensures that the learned clusters are predictive of future behavior and that the temporal structure of the data is preserved throughout the learning process. We validate the effectiveness of SC GAN by comparing it to a predictor conditioned on observable classes. Our experiments demonstrate that SC GAN attains lower prediction errors, highlighting the strength of SC GAN's clusters derived from future motion. We also benchmark SC GAN against traditional clustering techniques, such as K-means and TS K-means, on both road users datasets (Argoverse [17]) and human motion trajectory datasets (THÖR [83], ETH/UCY [60, 76]). While all methods yield comparable results under standard conditions, SC GAN substantially outperforms traditional approaches in data drift scenarios, where training and testing trajectories follow different motion patterns. These results suggest that SC GAN is more robust to underrepresented or diverse trajectory behaviors.

In summary, the contributions of this chapter are as follows:

- We introduce the concept of data-driven classes, which represent groups of trajectories learned directly from trajectory data without relying on predefined labels or perceptual pipelines.
- We analyze the impact of different input feature representations and clustering strategies on the quality of the learned clusters, stressing the importance of future- and full-driven clustering for the trajectory prediction task and highlighting the differences between various input feature representations.
- We propose the Self-Conditioned GAN (SC GAN), a deep generative framework that simultaneously learns to cluster trajectories and generate realistic samples, preserving temporal dependencies. We evaluate SC GAN against traditional clustering methods (K-means and TS K-means) on various datasets, demonstrating its robustness and predictive capabilities, especially in data distribution drift scenarios.

6.1.2 Outline

This chapter is organized as follows. Sec. 6.2 formalizes and describes the process of clustering trajectory data. This section also details the different settings for clustering trajectory data through various input feature representations and compares traditional methods with our proposed Self-Conditioned GAN (SC GAN). Sec. 6.3 describes predictors conditioned on future trajectory states and compares them to baseline models on synthetic and real-world datasets. Sec. 6.4 introduces our SC GAN framework to cluster trajectory data followed by experiments comparing SC GAN with other trajectory clustering methods. Finally, Sec. 6.5 concludes this chapter with a description of the key contributions and findings, outlining the introduction of data-driven classes in trajectory prediction systems in Chapter 7.

6.2 Clustering Trajectory Data

The objective of trajectory clustering is to identify N_c distinct classes from a collection of trajectories, as illustrated in Fig. 6.1. Formally, the clustering task is defined as:

$$\phi_{\text{Clustering}}: \mathbf{P}_k \mapsto c_k, \quad c_k \in \{1, \dots, N_c\}, \quad (6.1)$$

where \mathbf{P}_k denotes the k -th trajectory and c_k its assigned cluster label. These clusters aim to capture consistent and distinctive patterns in trajectory behavior, which can support downstream tasks such as trajectory prediction.

We first transform raw trajectories into feature representations:

$$f_{\text{ext}}: \mathbf{P}_k \mapsto \mathbf{f}_k, \quad (6.2)$$

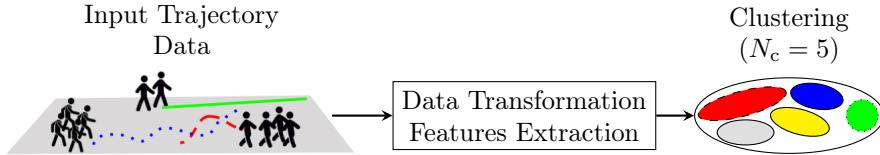


Figure 6.1: Trajectory clustering framework. From left to right: raw trajectory data is transformed into a suitable feature representation, which is then clustered into N_c groups (in this example, $N_c = 5$).

where f_{ext} denotes the feature extraction function, and \mathbf{f}_k the resulting feature vector for trajectory \mathbf{P}_k . The feature extraction process is discussed in detail in Sec. 6.2.1.

Subsequently, we apply clustering algorithms to the extracted features. We consider standard clustering algorithms, such as K-means, which require a 2D feature matrix. To accommodate sequential trajectory data, K-means requires flattening time-series inputs, which often results in the loss of misaligned temporal patterns. Time-Series K-Means mitigates this by preserving temporal alignment using Dynamic Time Warping. However, the high computational cost of DTW makes this approach unsuitable for large-scale or real-time applications.

To overcome the limitations of traditional clustering methods, we introduce the Self-Conditioned GAN, a deep generative framework detailed in Sec. 6.4.

6.2.1 Input Feature Representations

The inputs of clustering algorithms, particularly the choice of features and the segment of the trajectory from which they are extracted, are crucial in determining the resulting clusters and the quality of the identified classes. Clustering algorithms take as input a 2D matrix or a 3D tensor of trajectory-derived features \mathbf{f}_k as defined in Eq. (6.2). In what follows, we describe the feature representations explored in this work for a trajectory of arbitrary length H . For clarity, we omit the trajectory index k in the notation.

- **Displacement vectors:** As defined in Sec. 1.2.1, calculated as finite differences of 2D positions, capturing local motion between successive time steps. For a trajectory $\mathbf{P} = \{(x_t, y_t)\}_{t=1}^H$, the displacement vector at time t is given by:

$$\mathbf{d}_t = (x_t - x_{t-1}, y_t - y_{t-1}), \quad \text{for } t = 2, \dots, H. \quad (6.3)$$

The resulting trajectory is then flattened into a vector:

$$\mathbf{f} = (\mathbf{d}_1, \dots, \mathbf{d}_H) \in \mathbb{R}^{H \times 2}, \quad \mathbf{d}_1 = (0, 0). \quad (6.4)$$

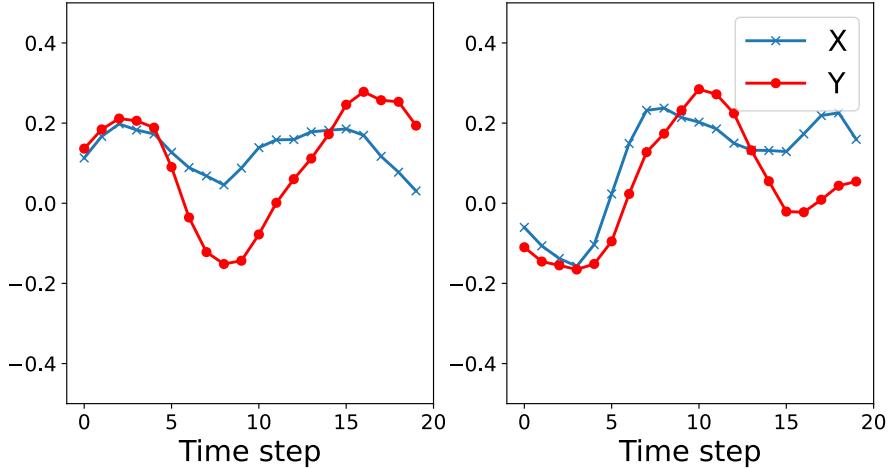


Figure 6.2: Top rank-2 principal components in THÖR-MAGNI Scenario 2 trajectory dataset: in the X (blue) and Y (red) axes.

- **Trajectory normalization:** As defined in Sec. 1.2.1, each trajectory is translated to the origin and rotated to align with the X-axis, thus removing environment-specific spatial biases (illustrated in Fig. 1.6). The normalized position $\mathbf{p}_t^{\text{norm}}$ at time step t is computed as:

$$\mathbf{p}_t^{\text{norm}} = \mathbf{R}_\theta^\top \cdot (\mathbf{p}_t - \mathbf{p}_1), \quad \text{for } t = 2, \dots, H, \quad (6.5)$$

where \mathbf{R}_θ is the 2D rotation matrix aligning the first displacement vector with the X-axis. The resulting trajectory is then flattened into a vector:

$$\mathbf{f} = (\mathbf{p}_1^{\text{norm}}, \dots, \mathbf{p}_H^{\text{norm}}) \in \mathbb{R}^{H \times 2}, \quad \mathbf{p}_1^{\text{norm}} = (0, 0). \quad (6.6)$$

- **Statistical features:** We compute summary statistics (minimum, maximum, mean, and standard deviation) over motion cues such as velocity, acceleration, heading, and path efficiency. To capture the dominant modes of variation, we additionally apply Principal Component Analysis (PCA) [75] to the aggregated features. Fig. 6.2 illustrates the first two principal components, which capture the most significant modes of variation in the statistical features, providing insights into the underlying motion patterns over time in the THÖR-MAGNI dataset.

Besides the type of features, the segment of the trajectory from which features are extracted plays a key role in trajectory prediction. Let $H \in \{O, L, T_P\}$ denote the segment length, where O is the number of observed steps, L the number of future steps, and T_P the total trajectory length. We consider three configurations:

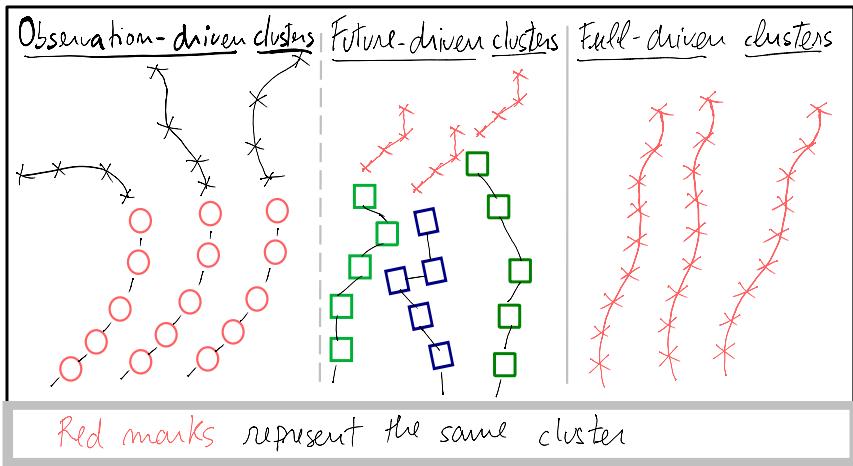


Figure 6.3: Cluster types derived from different segments of a trajectory. Each panel shows three example trajectories with 5 observation and 4 prediction time steps from the same cluster. **Left:** Observation-driven clusters (red circles). **Center:** Future-driven clusters (red crosses). **Right:** Full-driven clusters combining both segments.

- **Observation-driven clustering:** Features are computed from the observed segment only ($\mathbf{P}_k \in \mathbb{R}^{O \times 2}$ in Eq. (6.1)).
- **Future-driven clustering:** Features are computed from the future segment, unavailable at inference time ($\mathbf{P}_k \in \mathbb{R}^{L \times 2}$ in Eq. (6.1)).
- **Full-driven clustering:** Features are computed from the entire trajectory, including both observation and future segments ($\mathbf{P}_k \in \mathbb{R}^{T_P \times 2}$ in Eq. (6.1)).

Clusters derived from observed segments offer limited utility since they do not provide additional information beyond what is already accessible to the predictor. In contrast, future- and full-driven clusters contain privileged information that can be leveraged during training to improve predictive performance. These configurations are illustrated in Fig. 6.3 and their incorporation in predictors in Fig. 6.4.

6.2.2 Traditional Clustering Methods versus SC GAN

Traditional clustering methods like K-means operate on fixed-length vectors in a flat 2D feature space. To apply K-means to trajectory data, the trajectory

dataset must be reshaped into a matrix where each row corresponds to a trajectory, and each column represents a time-flattened feature dimension, such as sequential 2D displacements. The Euclidean distance applied to the flattened representation disregards temporal patterns misaligned with the time axis. As a result, trajectories that are similar in shape but temporally misaligned may be assigned to different clusters, leading to a loss of meaningful sequential information. An alternative approach to clustering sequential data involves extracting statistical descriptors from the trajectory, such as average velocity, acceleration, or path efficiency. While these features offer compact, low-dimensional representations, they may underrepresent non-linearities and transient motion patterns, reducing their effectiveness in capturing fine-grained trajectory dynamics.

Time-Series K-means (TS K-means) [91] mitigates the misalignment issue by using time-series distance metrics such as Dynamic Time Warping (DTW) to measure similarity between trajectories. DTW can capture misaligned similarities in temporal sequences, making TS K-means more suitable for time-series data, such as human motion trajectories. Nonetheless, the computational cost of DTW is substantially higher than that of standard Euclidean distance, limiting the scalability of TS K-means in large datasets or real-time systems.

To overcome these limitations, we propose SC GAN, a multi-task generative framework that jointly learns temporal embeddings and trajectory clusters. SC GAN uses LSTM-based architectures to encode the sequential structure of input trajectories and performs clustering within a learned feature space optimized for generation and prediction tasks. In addition, it employs K-means with Euclidean distance on the learned deep feature space, benefiting from the efficiency of K-means while preserving the temporal structure of the data. Hence, our approach resolves two critical limitations of K-means and TS K-means:

1. **Preservation of temporal dependencies:** Temporal patterns are encoded via deep sequence modeling underlying a trajectory classification task.
2. **Alignment with predictive objectives:** The learned clusters are optimized for the trajectory prediction task.

6.3 Prediction Conditioned on Future Insights

Data-driven classes aim to capture the latent structure of trajectory data and, ideally, encode information about future motion. When data-driven classes are derived from segments of the trajectory that extend beyond the observation window (future- or full-driven clustering), they provide privileged information that is not accessible at inference time. Although these classes cannot be directly incorporated into prediction systems as future information is not

available at inference time, they form theoretical trajectory forecasters conditioned on ideal information. The performance of predictors conditioned on ideal, future-informed class labels quantifies the upper limits of the potential forecasting performance of trajectory predictors only with respect to the predictor’s network configuration (i.e., this upper limit does not apply to other prediction methods). Moreover, predictors conditioned on future insights allow us to quantify and analyze the performance gap between practical and potentially ideal systems, providing insights into the benefits of ideal data-driven classes (e.g., what information they are capturing?) and the limitations of current trajectory prediction methods. In this section, we first introduce and formalize predictors conditioned on future insights in Sec. 6.3.1 and empirically evaluate their behavior under various feature and clustering configurations in Sec. 6.3.2. This analysis provides theoretical insights into the utility of data-driven classes and informs the design of practical predictive systems explored in Chapter 7.

6.3.1 Formalization

For a given dataset, data-driven class-conditioned predictors, whose class assignments are based on future trajectory states (future-driven clusters) or entire trajectory states (full-driven clusters) (gray images in Fig. 6.4), can be seen as members of a set of predictors, denoted by $\mathcal{S}_{\text{Ideal}}$, assuming there is a finite number of f_{ext} functions to extract features from the input states. Predictors in $\mathcal{S}_{\text{Ideal}}$ are conditioned on information that depends on the future and, thus, are impractical for real-world deployment where future states are unavailable at inference time. Formally, we define this set as:

$$\mathcal{S}_{\text{Ideal}} = \{\psi_{f_{\text{ext}}}(\mathbf{S}_k, c_k) \mid f_{\text{ext}} \in \mathcal{F}, c_k = \text{Clustering}(f_{\text{ext}}(\mathbf{S}_k, \mathbf{Y}_{\mathbf{S}_k}))\} \quad (6.7)$$

where $\psi_{f_{\text{ext}}}$ denotes a class-conditioned predictor, and c_k is the cluster label assigned to the k^{th} trajectory based on the features \mathbf{f}_k extracted from the future sequence of states $\mathbf{Y}_{\mathbf{S}_k}$, possibly in combination with the observed segment \mathbf{S}_k in the full-driven setting. The network configuration of these predictors is the same as the class-conditioned trajectory predictors presented in Fig. 4.1. The only difference lies in using the cluster id c_k as the conditioning variable, which is obtained from the clustering process. These predictors serve as valuable theoretical baselines, providing insights into the maximum achievable performance when privileged future information is available during training and inference.

6.3.2 Performance Analysis

This section analyzes predictors conditioned on data-driven class labels derived from the K-means algorithm, focusing on comparing different input feature representations for the clustering based on the prediction accuracy re-

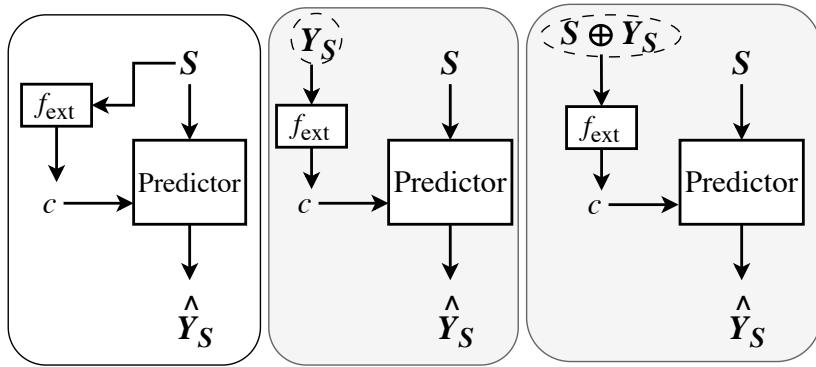


Figure 6.4: Data-driven class conditioned trajectory predictors at inference time: observation-driven clusters (**left**) conditioning trajectory predictors, future-driven clusters (**middle**) conditioning trajectory predictors, and full-driven clusters (**right**) conditioning trajectory predictors. Gray boxes indicate the predictors conditioned on future insights, and dashed objects highlight oracle information (i.e., information unavailable at inference time in practical applications).

sults. Therefore, it establishes theoretical insights by comparing predictors with access to privileged information, such as future trajectory states, with baseline models, including unconditional predictors and those conditioned on observation-driven clusters.

Datasets

We conduct our analysis on two datasets: a synthetic dataset and the THÖRMAGNI dataset previously used in Chapter 4.

The synthetic dataset is a controlled environment where we define ground-truth observable classes according to agent-specific motion dynamics. The synthetic dataset allows us to reduce class ambiguity (see Sec. 4.5.4), which enables the study of the influence of trajectory classes on trajectory prediction performance. We generate trajectories via a custom simulator with class-specific kinematic constraints. Each trajectory belongs to one of the following agent types:

- **slow-walker:** Low velocity, minimal orientation change; simulates cautious or conservative agents.
- **fast-walker:** High velocity, low directional variability; efficient, goal-directed motion.

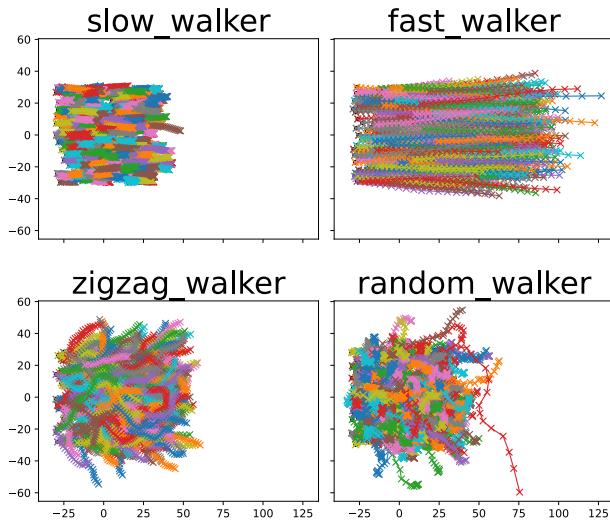


Figure 6.5: Trajectories in the synthetic dataset with four hard-coded observable classes: *slow-walker*, *fast-walker*, *zigzag-walker*, and *random-walker*.

- **zigzag-walker:** Frequent directional changes; mimics erratic or reactive movement.
- **random-walker:** High randomness in both speed and orientation; future motion is highly stochastic.

We generate 500 trajectories per class (balanced distribution) with 8 observation and 12 prediction time steps. Fig. 6.5 depicts the synthetic trajectory dataset.

THÖR-MAGNI dataset complements our analysis by offering real-world human motion data from a mock industrial environment. It allows us to evaluate how predictors perform under real-world noise and complexity and to quantify the generalization gap between synthetic and real-world settings. We use Scenarios 2 and 3 from THÖR-MAGNI, as previously described in Chapter 4.

Baselines and Metrics

We compare predictors conditioned on cluster labels with two baselines:

1. **Vanilla (class-unaware) predictors**, using only the observed trajectory segment as input to the predictors.

2. **Observable class-conditioned predictors**, using the observed trajectory segment and the human-annotated labels as inputs to the predictors.

We employ K-means on flattened inputs for trajectory clustering and use the Davies-Bouldin Index (DBI) [24], as defined in Eq. (1.5), to select the number of clusters. All prediction models use the single-output LSTM and Transformer-based architectures introduced in Chapter 4. We assess model performance using ADE and FDE, as defined in Eqs. (1.1) and (1.2), averaged across 5-fold cross-validation.

Results

Tab. 6.1 presents prediction performance for all model variants using 2D flattened displacements as clustering features. Both LSTM and Transformer-based predictors benefit substantially from conditioning on future- or full-driven clusters, which consistently outperform baselines and data-driven conditioning based on observed states. The improvement achieved by these predictors confirms the presence of latent trajectory patterns that are not captured by observed segments alone. In addition, observation-driven clusters yield negligible improvement, indicating that such clusters offer limited additional information beyond what the model already extracts from input sequences. We also show qualitative results in Fig. 6.6 where the difference between the representations derived from each cluster type is clear: observation-driven induces marginal information while future- and full-driven clusters substantially enhance trajectory predictions.

We perform a detailed analysis using the synthetic dataset to understand which trajectory types benefit most from data-driven clustering. Tab. 6.2 compares performance across cluster ids, highlighting that full-driven cluster conditioning yields the most substantial improvements for complex behaviors in the synthetic dataset. Refer to Fig. 6.7 for cluster compositions and the occurrence matrix illustrating the correspondence between the data-driven classes from Tab. 6.2 and the observable classes in the synthetic dataset. Those improvements primarily stem from the *zigzag-walker* and *random-walker* observable classes, which are decomposed into multiple data-driven clusters. This finer decomposition improves the representational discrimination of these observable classes by capturing subtle variations in motion patterns that are otherwise abstracted when conditioning on observable classes. Furthermore, Fig. 6.7 shows that complex trajectories (coming from the *zigzag-walker* and the *random-walker* observable classes) are distributed across clusters 4, 5, 6, and 7, reflecting the fine-grained distinctions captured by the clustering process. Clusters 1 and 2 align more closely with simpler, deterministic trajectory types (*slow* and *fast-walker*). These results suggest that data-driven classes are particularly valuable when modeling highly non-linear and complex trajectory patterns.

Table 6.1: Top-1 ADE/FDE for Transformer- and LSTM-based models on the synthetic and MAGNI datasets with 2D flattened displacements as clustering inputs. Bold and *italic* values highlight the best models and second-best models, respectively.

| Model Network | Model Type | Synthetic | MAGNI |
|---------------|----------------------------|--------------------------------------|--------------------------------------|
| Transformer | Baseline | 2.03±0.08 4.14±0.12 | 0.89±0.01 1.92±0.03 |
| | Observable class | 1.98±0.08 3.93±0.09 | 0.87±0.02 1.87±0.04 |
| | Observation cluster | 2.04±0.05 4.09±0.10 | 0.89±0.02 1.93±0.04 |
| | Future cluster | 1.80±0.05 3.61±0.06 | 0.65±0.01 1.47±0.02 |
| | Full cluster | <i>1.89±0.09</i> <i>3.80±0.15</i> | <i>0.78±0.01</i> <i>1.71±0.02</i> |
| | Baseline | 2.04±0.09 4.06±0.12 | 0.89±0.02 1.92±0.04 |
| LSTM | Observable class | 1.96±0.09 3.95±0.12 | 0.87±0.02 1.86±0.04 |
| | Observation cluster | 2.03±0.09 4.12±0.13 | 0.89±0.01 1.92±0.03 |
| | Future cluster | 1.77±0.07 3.62±0.11 | 0.64±0.01 1.45±0.02 |
| | Full cluster | <i>1.84±0.06</i> <i>3.76±0.13</i> | <i>0.77±0.01</i> <i>1.69±0.02</i> |

Furthermore, we visualize and compare the observed- and future-driven clusters derived from the synthetic dataset in Figs. 6.8 and 6.9. The observation-driven clusters show limited diversity and contain several ambiguous groupings. This outcome suggests that the observation horizon (8 time steps) cannot fully capture the underlying motion patterns, resulting in suboptimal clustering due to the lack of temporal context. In contrast, the future-driven clusters demonstrate greater granularity and heterogeneity. These clusters more effectively capture finer trajectory characteristics, such as speed variations and directional changes (e.g., clusters 4, 7, 13, and 15), producing a more meaningful partitioning of the trajectory space. Altogether, these visualizations reinforce that observed tracklets offer limited context for finding effective trajectory

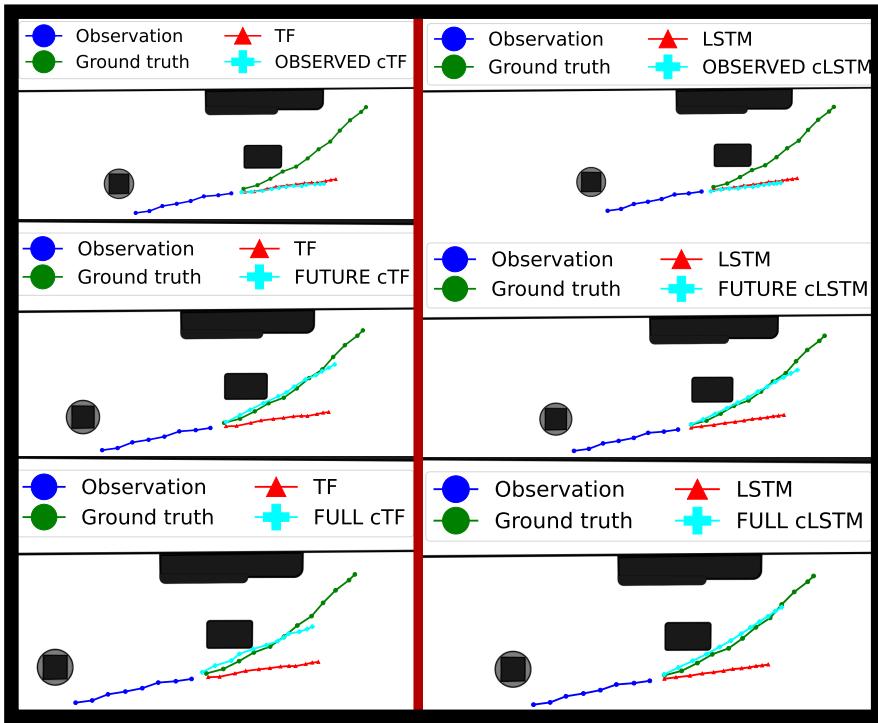


Figure 6.6: Trajectory predictions for THÖR-MAGNI dataset with 2D flattened displacements as clustering inputs. The observation-driven clusters induce marginal information, and future- and full-driven clusters substantially enhance trajectory predictions.

classes while future- and full-driven segments encode critical information for discovering meaningful classes for trajectory prediction.

We further evaluate the impact of various feature representations on clustering effectiveness. Using statistical summaries (e.g., velocity, acceleration, path efficiency, PCA components) to represent trajectories yields competitive results, as shown in Tab. 6.3. While ADE values are comparable to displacement-based clustering, FDE scores are substantially improved, particularly in the real-world THÖR-MAGNI dataset. This indicates that statistical features capture final displacement trends more effectively.

Finally, we consider normalized trajectories aligned to a common frame via translation and rotation. As shown in Tab. 6.4, performance remains on par with displacement-based inputs (see Tab. 6.1). However, normalized features offer superior generalization potential for unseen or dynamic environments where invariance to scene geometry is critical.

Table 6.2: Top-1 ADE/FDE average prediction scores per cluster id in the synthetic dataset for 1 fold.

| Cluster id | Baseline | Observable Class | Data-driven Class |
|------------|-------------|------------------|-------------------|
| 1 | 2.11 | 2.04 | 2.08 |
| | 4.12 | 3.98 | 3.84 |
| 2 | 0.64 | 0.72 | 0.67 |
| | 1.10 | 1.27 | 1.20 |
| 3 | 1.80 | 1.75 | 1.76 |
| | 4.48 | 4.33 | 4.38 |
| 4 | 2.89 | 2.93 | 2.60 |
| | 5.83 | 5.79 | 4.90 |
| 5 | 7.28 | 7.02 | 6.44 |
| | 13.06 | 12.34 | 11.36 |
| 7 | 2.38 | 2.37 | 2.11 |
| | 4.66 | 4.52 | 3.91 |

Table 6.3: Top-1 ADE/FDE for Transformer- and LSTM-based models on the synthetic and MAGNI datasets with statistical-based features clustering. Bold and *italic* values highlight the best models and second-best models, respectively.

| Model Network | Model Type | Synthetic | MAGNI |
|---------------|----------------------------|--------------------------------------|--------------------------------------|
| Transformer | Observation cluster | 2.04±0.07 4.06±0.09 | 0.90±0.02 1.94±0.04 |
| | Future cluster | 1.83±0.10 3.57±0.14 | 0.65±0.01 1.28±0.04 |
| | Full cluster | 1.95±0.08 3.81±0.15 | 0.71±0.01 1.39±0.03 |
| LSTM | Observation cluster | 2.04±0.08 4.13±0.12 | 0.89±0.02 1.92±0.04 |
| | Future cluster | 1.82±0.07 3.64±0.10 | 0.63±0.02 1.26±0.04 |
| | Full cluster | 1.89±0.09 3.74±0.15 | 0.69±0.01 1.36±0.04 |

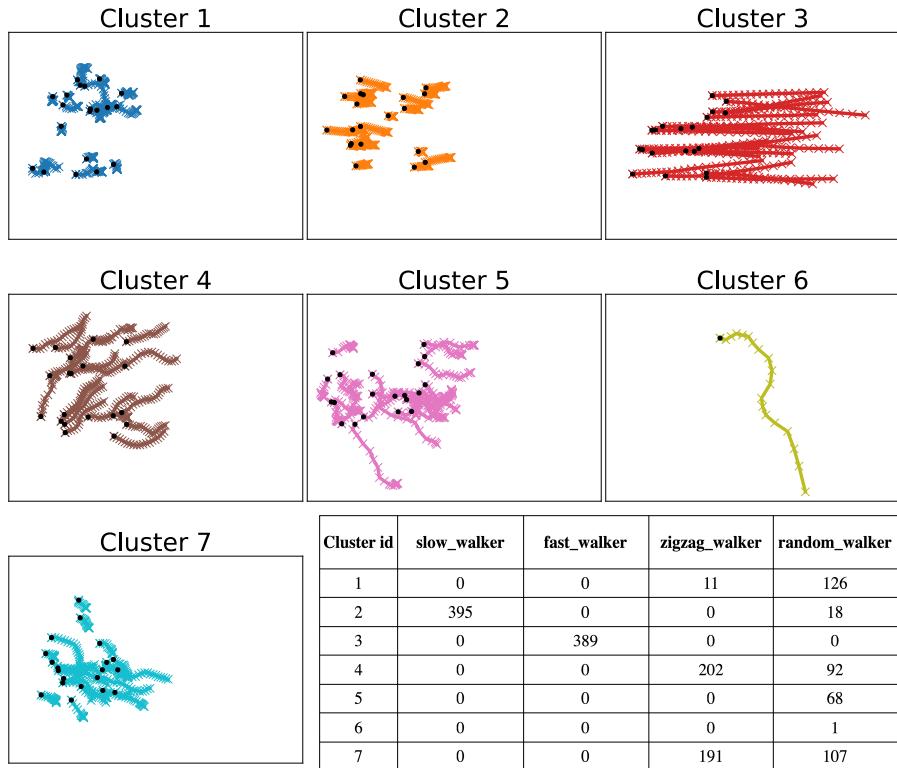


Figure 6.7: Full-driven clusters in the synthetic dataset: different motion patterns are grouped in each cluster, where the *slow-walker* and *fast-walker* observable classes have particularly clear representation in two of the seven clusters. The remaining clusters distinguish fine-grained dynamics present in the *zigzag-walker* and *random-walker* observable classes. The bottom right shows the “data-driven–observable class” occurrence matrix.

In summary, these analyses demonstrate that data-driven trajectory classes, particularly those derived from future or full trajectories, are strong predictors of trajectory dynamics under ideal conditions. Their superiority over vanilla and observable class-conditioned baselines underscores the latent structure in trajectory data that is inaccessible through observed segments alone. These findings validate the theoretical value of predictors conditioned on future insights and motivate the development of practical mechanisms to approximate or infer such latent classes during inference.

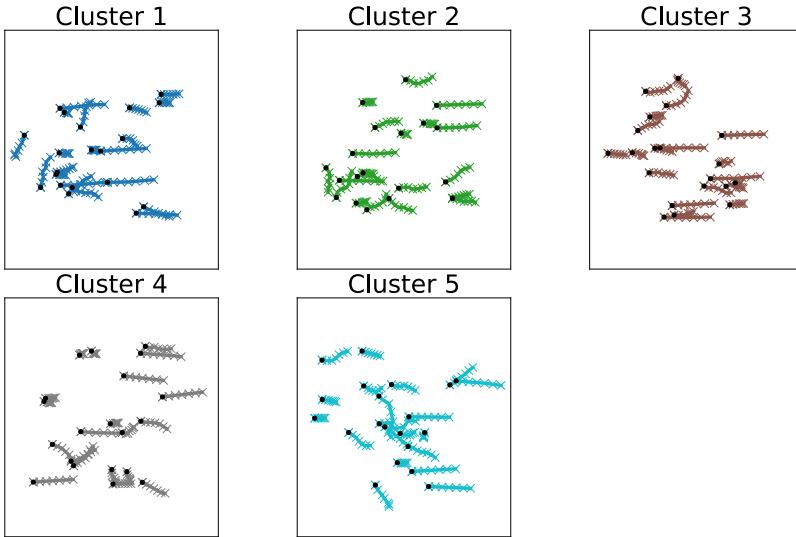


Figure 6.8: The limited observation horizon (8 time steps) leads to coarse and sometimes inconsistent groupings. Several motion patterns are mixed due to insufficient temporal context, making it difficult to distinguish between distinct behaviors.

6.4 Self-Conditioned GAN for Learning Trajectory Classes

As an alternative to the traditional clustering methods discussed in Sec. 6.2.2, we propose a novel approach to learning trajectory classes based on deep learning methods, specifically the Self-Conditioned GAN (SC GAN) [64]. We adapt SC GAN from image generation [64] to the trajectory clustering and generation tasks. In its original formulation, the Self-Conditioned GAN was designed to generate realistic images and address the well-known issue of mode collapse in GANs. While GANs aim to capture diverse modes from the original data distribution, they often focus disproportionately on dominant parts when the dataset is biased, neglecting less-represented ones. Conditional GANs [71] partially address this limitation by incorporating explicit conditions, such as class labels, which provide greater control over the modes generated by the model. The Self-Conditioned GAN takes this further by using unsupervised classes derived from the discriminator’s feature space to condition the generator, mitigating mode collapse in a self-conditioned manner. The generation and clustering dual functionality allows the SC GAN to predict trajectories (a useful ability we will explore in the next chapter) and automatically cluster the data within the discriminator’s feature space during training. The Self-

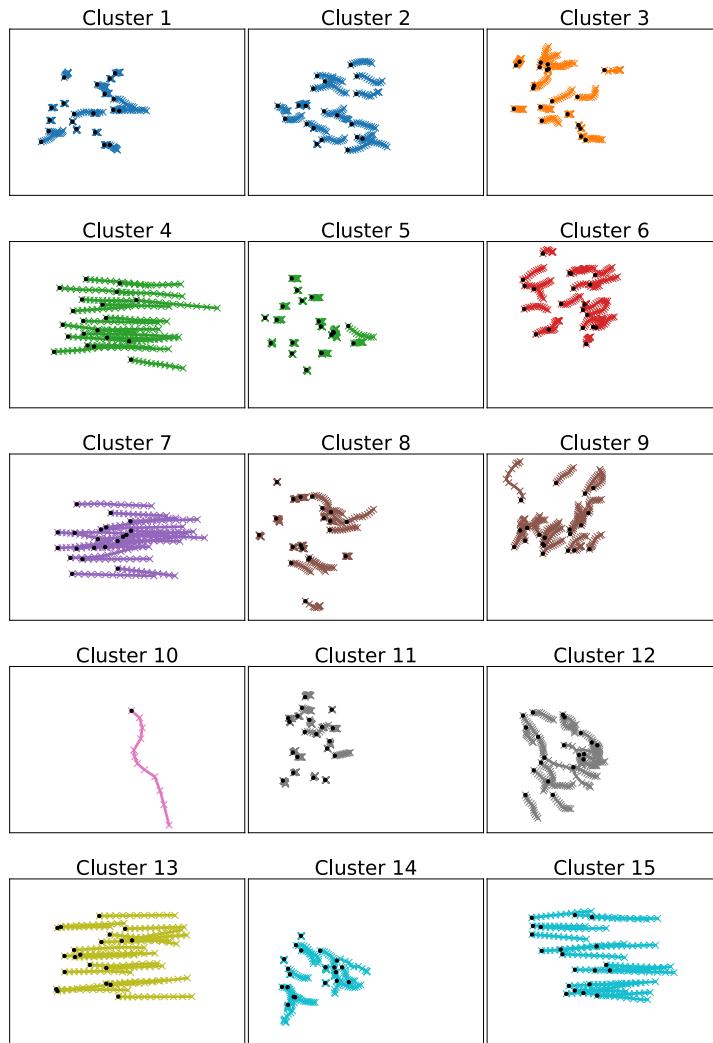


Figure 6.9: These clusters show greater diversity and behavioral specificity than observed- and full-driven clusters, capturing fine-grained distinctions based on motion speed (e.g., clusters 4, 7, 13, and 15) and trajectory geometry. The future trajectory segment provides a richer context for identifying meaningful agent classes.

Conditioned GAN leverages the features produced by the discriminator's encoder to build clusters. The underlying intuition is that the discriminator's fea-

Table 6.4: Top-1 ADE/FDE for Transformer- and LSTM-based models on the synthetic and MAGNI datasets with normalized trajectory clustering. Bold and *italic* values highlight the best models and second-best models, respectively.

| Model Network | Model Type | Synthetic | MAGNI |
|---------------|----------------------------|--------------------------|--------------------------|
| Transformer | Observation cluster | 2.05 ± 0.05 | 0.90 ± 0.02 |
| | | 4.08 ± 0.08 | 1.94 ± 0.03 |
| | Future cluster | $\mathbf{1.79 \pm 0.12}$ | $\mathbf{0.67 \pm 0.02}$ |
| LSTM | Future cluster | $\mathbf{3.58 \pm 0.18}$ | $\mathbf{1.50 \pm 0.03}$ |
| | Full cluster | 1.80 ± 0.10 | 0.63 ± 0.01 |
| | | 3.61 ± 0.16 | 1.42 ± 0.02 |
| Transformer | Observation cluster | 2.07 ± 0.07 | 0.88 ± 0.02 |
| | | 4.19 ± 0.11 | 1.91 ± 0.03 |
| | Future cluster | $\mathbf{1.77 \pm 0.06}$ | $\mathbf{0.67 \pm 0.01}$ |
| LSTM | Future cluster | $\mathbf{3.60 \pm 0.10}$ | $\mathbf{1.51 \pm 0.03}$ |
| | Full cluster | 1.83 ± 0.10 | 0.65 ± 0.02 |
| | | 3.71 ± 0.15 | 1.44 ± 0.04 |

ture space, designed to differentiate real from generated samples, inherently extracts meaningful representations that can be used for other downstream tasks [77]. For instance, the discriminator has been used in other trajectory prediction methods to filter socially acceptable future trajectories [51]. Finally, the Self-Conditioned GAN creates a synergistic relationship between trajectory clustering and generation, where the quality of the clusters directly influences trajectory generation, and the generation process, in turn, iteratively refines the clusters. This integration ensures that the clusters are deeply embedded in the trajectory generation and prediction tasks, stressing their relevance and effectiveness.

6.4.1 Overall Model Architecture

The goal of SC GAN is to estimate the mapping defined in Eq. (6.1) by leveraging the samples processed within a GAN-based framework while generating trajectories conditioned on such clusters. Those clusters are derived from features extracted from the discriminator’s encoder:

$$Enc_D : \mathbf{Y}_k \mapsto c_k, \quad c_k \in \{1, \dots, N_c\}, \quad (6.8)$$

where Enc_D is the discriminator’s encoder, and \mathbf{Y}_k corresponds to the trajectory displacements given by the finite differences of 2D positions. For future-

driven clusters, \mathbf{Y}_k comprises only the future trajectory displacements following \mathbf{S}_k , the observed tracklet comprising the observed displacements. Alternatively, full-driven clusters include the entire vector of displacements, similar to generating an entire trajectory starting from the origin of a 2D reference frame $(0, 0)$. Hence, the SC GAN framework provides two distinct clustering modes as illustrated in Fig. 6.10: future-driven clusters (FD SC GAN in blue) and full-driven clusters (FP SC GAN in red). In both cases, the resulting predictors are members of the set $\mathcal{S}_{\text{Ideal}}$, as they rely on future trajectory information not available at inference time for the prediction task.

As outlined in Sec. 4.3.2, the GAN’s discriminator (D) consists of two main components: a temporal encoder (Enc_D) and an MLP. The temporal encoder, implemented as an LSTM, processes trajectory states to extract time-dependent features, which are passed through a K-means clustering step. Those features are subsequently fed into the MLP to produce the final discriminative score. During trajectory generation, the cluster indices condition the generator (G), producing samples corresponding to the conditioning clusters.

To maintain the relevance of trajectory clusters as the discriminator’s representational power evolves, the feature space is periodically re-clustered during training, following the approach outlined in [64]. This dynamic updating enables the framework to capture increasingly refined trajectory patterns over time. To avoid the computational overhead of retraining the GAN entirely after each re-clustering step, we apply the classical Hungarian minimum-cost matching algorithm [57]. This algorithm efficiently aligns the newly derived clusters $\{c^{\text{new},(i)}\}_{i=1}^{N_c}$ with the existing clusters $\{c^{\text{old},(i)}\}_{i=1}^{N_c}$ by minimizing:

$$\mathcal{L}_{\text{match}}(\rho) = \sum_{i=1}^{N_c} |c^{\text{old},(i)} \setminus c^{\text{new},\rho(i)}|, \quad (6.9)$$

where ρ is the permutation function that we aim to find, and $|c^{\text{old},(i)} \setminus c^{\text{new},\rho(i)}|$ represents the number of samples in the old cluster i that are missing from the new cluster $\rho(i)$. Alg. 1 summarizes the entire training process of SC GAN.

6.4.2 Future-driven SC GAN

The training data for this task, $\{(\mathbf{S}_k, \mathbf{Y}_{\mathbf{S}_k}, \mathbf{P}_{\mathbf{S}_k})\}_k$, consists of triplets of tracklets, the futures, and the corresponding ground truth positions. FD SC GAN processes the input trajectory states (\mathbf{S}_k), incorporates the cluster id (c_k), and generates the future states predictions ($\hat{\mathbf{Y}}_{\mathbf{S}_k}$). Here, the cluster id (c_k) represents the cluster to which the embeddings of the k^{th} sample belong. As described in Sec. 6.4.1, \mathbf{Y}_k corresponds to the future of a tracklet ($\mathbf{Y}_{\mathbf{S}_k}$), indicating that the generator functions as a trajectory predictor. Thus, akin to the GAN-based forecasting model (see Sec. 4.3.2), the generator’s loss function combines the prediction loss (MSE) with the GAN loss (see Eq. (4.5)).

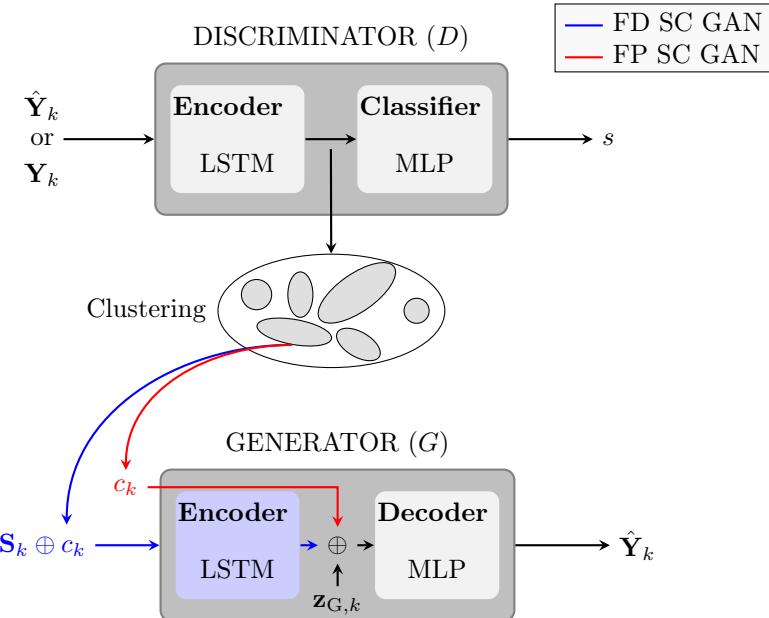


Figure 6.10: SC GAN architecture for generating future-driven clusters (FD SC GAN in **blue**) and full-driven clusters (FP SC GAN in **red**) embedded in the trajectory generation task. **Top:** During the discriminator’s training, inputs consist of either generated samples (\hat{Y}_k) or real samples (Y_k). Features extracted from real samples are clustered to drive the generator’s outcomes. **Bottom:** During the generator’s training, the cluster indices obtained from the discriminator’s feature space condition the generator, guiding its output.

The distinction from the vanilla GAN-based forecaster lies in including the data-driven cluster id, c_k , which explicitly influences the prediction.

This approach’s resulting clusters are based solely on future trajectory data, without connection to the observed tracklets. This lack of connection introduces challenges when integrating future-driven clustering into trajectory prediction frameworks, particularly during inference. Since the clustering process relies solely on future trajectory information (unavailable at inference time), it becomes non-trivial to establish a link between the observed tracklets and the appropriate clusters. Without this link, the predictor cannot effectively leverage future-driven clusters during inference, making their integration into a practical prediction pipeline difficult. Therefore, future-driven clusters represent possible futures and can only implicitly provide this information to downstream predictors (see Sec. 7.2). In contrast, full-driven clusters (detailed in Sec. 6.4.3) offer a more practical alternative for end-to-end trajectory prediction. By incorporating information from both the observed tracklet and the

Algorithm 1 Training Process of Self-Conditioned GAN

Require: Training data $\{(\mathbf{Y}_k, \mathbf{P}_{\mathbf{Y}_k})\}_k$ (trajectory states and ground truth positions), number of clusters N_c

- ```

1: Initialize GAN components: generator G , discriminator D (with temporal
 encoder Enc_D and MLP).
2: Run K-means on Enc_D outputs $\{c^{\text{old},(i)}\}_{i=1}^{N_c}$.
3: while training not converged do
4: Step 1: Discriminator Training
5: $s_k = (\text{MLP} \circ Enc_D)(Y_k)$ \triangleright Discriminator's scores.
6: Update D using the BCE loss.
7: Step 2: Generator Training
8: $f_k = Enc_D(Y_k)$ \triangleright Temporal features from discriminator.
9: $c_k = \text{K-means inference}(f_k)$ \triangleright Get class from ground-truth sample.
10: $\hat{Y}_k = G(\mathbf{z}_{G,k}, c_k)$ \triangleright Include S_k in the input if future-driven clusters.
11: Update G using Eq. (4.5) if future-driven or Eq. (6.10) if full-driven
 clusters.
12: Step 3: Re-clustering (Periodic)
13: if re-clustering condition met then
14: Re-cluster the discriminator's feature space with K-means.
15: Find the matching ρ that minimizes Eq. (6.9).
16: Update $\{c^{(i)}\}_{i=1}^{N_c} \leftarrow \{c^{\text{new},\rho(i)}\}_{i=1}^{N_c}$.
17: end if
18: end while
19: return Trained SC GAN model

```

future, they provide a cohesive representation that simplifies their application in prediction frameworks.

### 6.4.3 Full-driven SC GAN

The training data for this task,  $\{\mathbf{Y}_k, \mathbf{P}_k\}_k$ , consists of tuples of ground truth displacements and the corresponding 2D positions. As illustrated in Fig. 6.10, the primary distinction from the previously described FD SC GAN lies in the generator's inputs. Here, the generator relies solely on Gaussian noise ( $\mathbf{z}_{G,k}$ ) and the clustering id,  $c_k$ , sampled from the corresponding clustering distribution. Unlike the future-driven case, the cluster index in this context represents the cluster encompassing the entire trajectory's states for the  $k^{\text{th}}$  sample. Additionally,  $\mathbf{Y}_k$  corresponds to the entire displacements vector, indicating that the generator produces complete synthetic trajectories translated to the origin of the 2D reference frame. In this setup, the generator's loss function combines

a reconstruction loss (MSE over the entire trajectory) with the GAN loss, as follows:

$$\mathcal{L}_{\text{G-F}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \lambda_1 \mathcal{L}_{\text{MSE-F}} + \lambda_2 \left( \frac{1}{2} \mathbb{E}[(D(\mathbf{Y}) - 1)^2] + \frac{1}{2} \mathbb{E}[D(\hat{\mathbf{Y}})^2] \right), \quad (6.10)$$

where the trajectory reconstruction term  $\mathcal{L}_{\text{MSE-F}}$  is given by:

$$\mathcal{L}_{\text{MSE-F}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{T_P} \sum_{j=1}^{T_P} \|\mathbf{p}_j - \hat{\mathbf{p}}_j\|_2^2. \quad (6.11)$$

#### 6.4.4 Experiments

This subsection presents an evaluation of the SC GAN framework and other class-conditioned trajectory prediction methods to show the representational power of SC GAN in practical prediction settings. Specifically, we first compare FD SC GAN with the observable conditioned counterpart predictor and then FP SC GAN with other clustering methods, both quantitatively and qualitatively.

#### Datasets

We evaluate SC GAN using two experimental protocols: (1) a train/test split for in-distribution evaluation of FD SC GAN (future-driven) and (2) a leave-one-dataset-out setting for cross-domain generalization of FP SC GAN (full-driven). FD SC GAN evaluation is conducted on THÖR [83] and Argoverse [17] (see Fig. 6.11). These datasets provide diverse scenarios representing indoor robotics and road-based environments, respectively. The THÖR dataset consists of trajectories of humans operating in an industrial-like environment. The dataset includes three experimental scenarios where individuals move between goal points while engaged in industrial-tailored activities (see Sec. 2.1.2), assuming one of three roles: 5-6 participants as *Visitors*, two as *workers*, and one as *inspector*. Inspired by established benchmarks [52], we segmented each trajectory into tracks of 8-time steps (3.2 s) for observation and 12-time steps (4.8 s) for prediction. To prepare the data for experiments, we applied preprocessing steps<sup>1</sup> based on [82], which include:

1. Downsampling the signal to 400 ms intervals.
2. Linearly interpolating missing detections to ensure continuity.
3. Smoothing trajectories using a moving average filter with a window size of 800 ms.

---

<sup>1</sup><https://github.com/tmralmeida/python-tools>

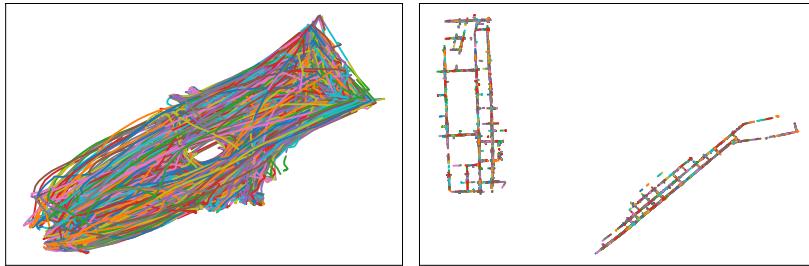


Figure 6.11: Trajectories in the THÖR (**left**) and Argoverse (**right**) datasets.

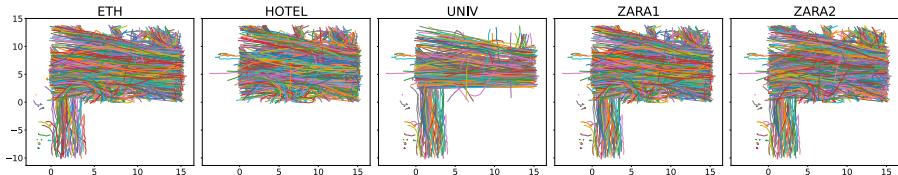


Figure 6.12: Trajectories in the ETH/UCY benchmark.

The Argoverse dataset comprises trajectories collected in road environments involving three distinct agent types: autonomous vehicles (*av*), regular vehicles (*agents*), and other road users (*others*). Each trajectory consists of observed tracklets spanning 20 time steps (2 s) and corresponding predictions over 30 time steps (3 s). Following [16], we sampled 5726 trajectories for training, 2100 for validation, and 1678 for testing. We train and evaluate our methods 5 times and average the final scores in both datasets.

FP SC GAN leave-one-dataset-out evaluation is conducted on the widely adopted ETH/UCY benchmark [60, 76] comprising five datasets: ETH, HOTEL, UNIV, ZARA1, and ZARA2 (Fig. 6.12). This benchmark has an observation length of 8-time steps (3.2 s) and a prediction length of 12-time steps (4.8 s), consistent with THÖR. We do not overlap segments of entire trajectories. In this setup, four datasets are used for training, while the remaining one is used for testing. We repeat the leave-one-dataset-out experiment five times and report averaged metrics.

### Baselines and Metrics

The baseline models share the same configuration as the one used for our SC GAN variants, but with other conditioning classes. For instance, the conditioning classes are the observable classes for the class-conditioned GAN (cGAN). In contrast, for our FD SC GAN, the conditioning classes are the data-driven classes from an entire sequence of trajectory states (see Tab. 6.6).

Table 6.5: Number of clusters found in the training set of each dataset.

| Dataset   | K-means | TS K-means | FP SC GAN |
|-----------|---------|------------|-----------|
| THÖR      | 3       | 12         | 7         |
| Argoverse | 5       | 5          | 5         |
| ETH       | 5       | 5          | 6         |
| HOTEL     | 5       | 5          | 5         |
| UNIV      | 5       | 5          | 7         |
| ZARA1     | 5       | 5          | 4         |
| ZARA2     | 5       | 5          | 4         |

For trajectory prediction, we report Top- $K$  ADE and FDE for  $K = 1$  and  $K = 3$ , as defined in Eqs. (1.1) and (1.2). To determine the number of clusters for trajectory clustering, we use the Davies-Bouldin Index (DBI), as defined in Eq. (1.5). The resulting optimal number of clusters for the three datasets is reported in Tab. 6.5. As we can see, the number of clusters varies across datasets, with K-means and TS K-means yielding similar results for all datasets except for THÖR, while FP SC GAN presents more diversified results than the other traditional clustering methods.

## Results

We start by evaluating the impact of future-driven trajectory classes on prediction accuracy. Specifically, we compare the performance of FD SC GAN, which conditions on clusters derived from future trajectory segments, with an observable class-conditioned GAN (cGAN). Tab. 6.6 presents the Top-1 ADE and FDE metrics on the test sets of the THÖR and Argoverse datasets. Across both domains, FD SC GAN consistently outperforms the observable class-conditioned baseline. These results indicate that, when assumed to be known, future-driven trajectory classes provide more informative and predictive conditioning signals than human-annotated semantic classes. This supports the hypothesis that clustering based on motion dynamics offers a more discriminative basis for forecasting than externally defined categories. In addition, SC GAN is validated as a framework providing powerful data-driven signals for trajectory prediction.

To evaluate the effectiveness of FP SC GAN (full-driven clustering) in trajectory prediction, we integrate it into a multi-stage predictive framework in which clustering explicitly conditions the prediction process. This approach enables the model to exploit the representational benefits of data-driven classes derived from entire trajectory segments. The predictive framework is described in detail in Sec. 7.3.

Across most datasets, predictors conditioned on trajectory classes derived from K-means, TS K-means, and FP SC GAN demonstrate comparable per-

Table 6.6: Top-1 ADE/FDE metrics (in meters) in the test sets for our future-driven SC GAN and the observable class-conditional counterpart. Bold values highlight the best prediction scores.

| Dataset   | cGAN            | FD SC GAN                         |
|-----------|-----------------|-----------------------------------|
| THÖR      | $0.66 \pm 0.01$ | <b><math>0.59 \pm 0.01</math></b> |
|           | $1.11 \pm 0.01$ | <b><math>0.94 \pm 0.02</math></b> |
| Argoverse | $1.92 \pm 0.02$ | <b><math>1.79 \pm 0.01</math></b> |
|           | $3.32 \pm 0.03$ | <b><math>2.89 \pm 0.04</math></b> |

Table 6.7: Top-3 ADE and FDE ( $\downarrow$ ) metrics in the HOTEL test set for a conditioned GAN predictor conditioned on classes derived from different clustering algorithms. Bold values highlight the best prediction scores.

| Method     | ADE                               | FDE                               |
|------------|-----------------------------------|-----------------------------------|
| Baseline   | $0.87 \pm 0.07$                   | $1.64 \pm 0.12$                   |
| K-means    | $1.03 \pm 0.05$                   | $1.90 \pm 0.09$                   |
| TS K-means | $1.04 \pm 0.05$                   | $1.81 \pm 0.06$                   |
| FP SC GAN  | <b><math>0.80 \pm 0.10</math></b> | <b><math>1.44 \pm 0.12</math></b> |

formance. This result suggests that all three clustering methods are capable of capturing motion patterns relevant to trajectory forecasting and of transferring this structure to the predictor. However, an exception is observed in the HOTEL scene from the ETH/UCY benchmark. As previously reported by [88], the training set in this scene is dominated by horizontal motion, whereas the test set primarily contains vertical trajectories. This domain shift is illustrated in Fig. 6.13, which shows that the majority of test samples satisfy the inequality  $|y_{t+1} - y_t| > |x_{t+1} - x_t|, \forall t \in \{1, T_P\}$ , in contrast to the training distribution. Such a shift hinders models that rely heavily on environment-dependent patterns, highlighting the importance of robustness to spatial biases in predictive systems. Tab. 6.7 summarizes the prediction performance of GAN-based models conditioned on clusters derived from various algorithms. Among them, FP SC GAN consistently outperforms traditional clustering-based predictors, demonstrating robustness under distribution drifts. These results suggest that FP SC GAN clusters can better generalize across variations in navigational styles by capturing underrepresented modes in the data.

Contrary to traditional clustering methods, FP SC GAN stands out not only as a clustering tool but also as a generative model. As described in Sec. 6.4, its dual mechanism allows clustering to guide generation, thus avoiding mode collapse, while the generative process reinforces meaningful clustering. To illustrate this advantage, we compare FP SC GAN with a variant named Full

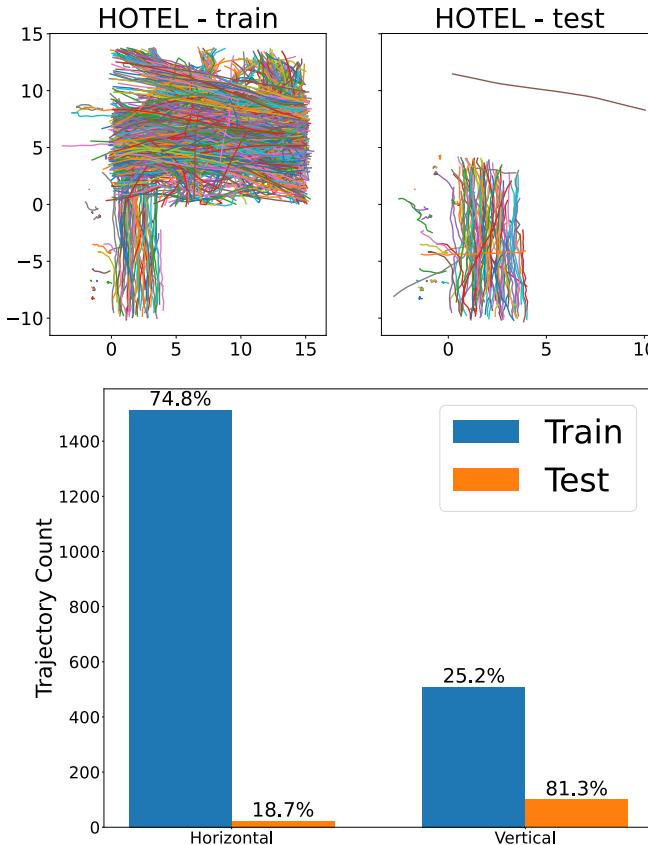


Figure 6.13: Trajectories and directional statistics in the HOTEL dataset. **Top:** Sampled trajectories from the training and test splits. **Bottom:** Proportion of trajectories showing vertical ( $|y_{t+1} - y_t| > |x_{t+1} - x_t|$ ) versus horizontal motion, highlighting the distributional drift between train and test sets.

Path GAN (FP GAN), which shares the same architecture and training protocol but omits the clustering conditioning mechanism. Fig. 6.14 shows generated trajectories (translated to the origin) for the UNIV dataset using variety loss with  $N_c = 7$ . The vanilla FP GAN faces mode collapse, generating only a narrow set of behaviors. In contrast, FP SC GAN produces a more diverse and representative set of trajectories, covering the broader motion spectrum of the dataset.

We depict the differences in cluster formation across methods in Fig. 6.15 using the THÖR dataset. While K-means identifies only three clusters, TS K-means and FP SC GAN capture a richer and more fine-grained structure in the

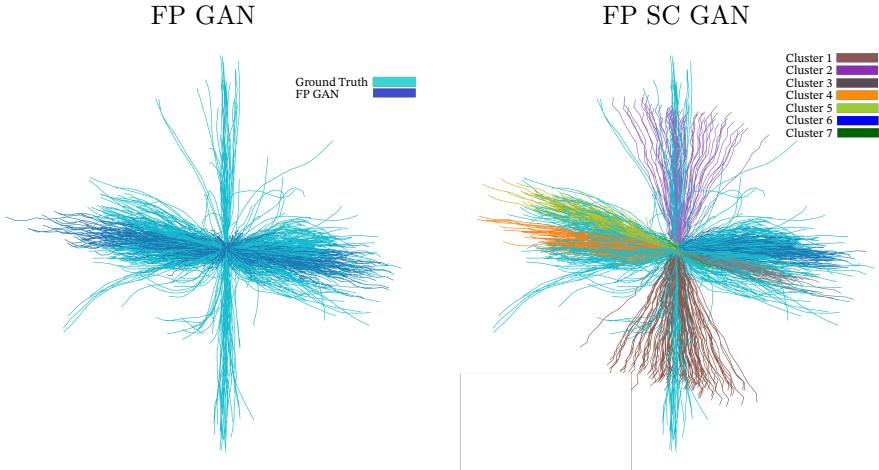


Figure 6.14: Overlapping between ground truth and generated samples from: FP GAN (**left**) without conditioning and FP SC GAN (**right**).

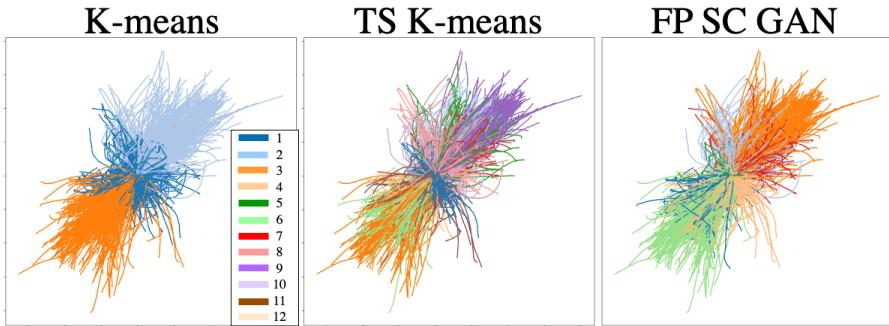


Figure 6.15: Clusters from each clustering method: K-means (**left**), TS K-means (**middle**), and FP SC GAN (**right**).

data. This discrepancy reflects the inability of K-means to capture temporal dynamics, as it operates on flattened representations, whereas TS K-means and FP SC GAN preserve time-series characteristics.

To better understand what makes specific trajectory patterns more difficult to predict, Fig. 6.16 shows examples from the most and least challenging clusters in THÖR. The most challenging clusters, defined by the highest ADE/FDE, primarily contain non-linear trajectories with turns or abrupt directional changes. In contrast, the least challenging clusters are composed of nearly straight trajectories. This observation reinforces the connection between trajectory complexity and prediction error, as complex motions are harder to

model and tend to correspond to underrepresented modes in GAN training. This information is crucial to improve the prediction of the most challenging groups of trajectories detailed in Sec. 7.2.

Lastly, Fig. 6.17 presents randomly selected clusters from THÖR and Argoverse, providing qualitative insights into the types of structure captured by FP SC GAN. These examples demonstrate that the FP SC GAN effectively groups trajectories with shared characteristics. In the THÖR dataset, which features a constrained indoor environment, trajectories are less linear compared to road environments like Argoverse, where agents typically move along straighter paths with fewer directional changes. The most distinguishing trajectory motion patterns differ between the datasets: in THÖR, movement direction is the dominant feature, while in Argoverse, navigation distance plays a more important role.

In summary, our results demonstrate that future-driven SC GAN consistently outperforms a GAN model based on observable classes (cGAN). This indicates that the data-driven classes found in the discriminator’s feature space of SC GAN, grounded in the dynamics of future motion, constitute a more informative and discriminative representation than human-annotated observable labels in both the THÖR and Argoverse datasets. Furthermore, while full-driven SC GAN achieves predictive performance comparable to traditional clustering techniques such as K-means and TS K-means, it yields more generalizable cluster structures, especially under distributional drifts. Finally, our visual analyses confirm that SC GAN clusters not only capture meaningful behavioral patterns but also reflect underlying structural complexity in the trajectory space. These insights are further leveraged in the predictive architectures developed in Chapter 7, where we exploit these learned clusters to enhance trajectory forecasting.

## 6.5 Conclusions and Outlook

Observable classes and actions per frame enhance trajectory prediction and provide semantic interpretability. However, observable classes are sometimes ambiguous, and actions rely on external perception modules susceptible to detection errors and noise. This chapter introduced an alternative approach: data-driven trajectory classes derived solely from motion trajectory dynamics without requiring external semantic annotations. Although less interpretable, these classes show higher consistency and predictive utility in synthetic and real-world environments, depending on future trajectory states. Consequently, we also introduced the concept of predictors conditioned on future insights, which assumes access to future trajectory information. While not deployable in practice, these models are theoretical baselines that demonstrate the utility of data-driven trajectory classes. They also provide valuable signals that can

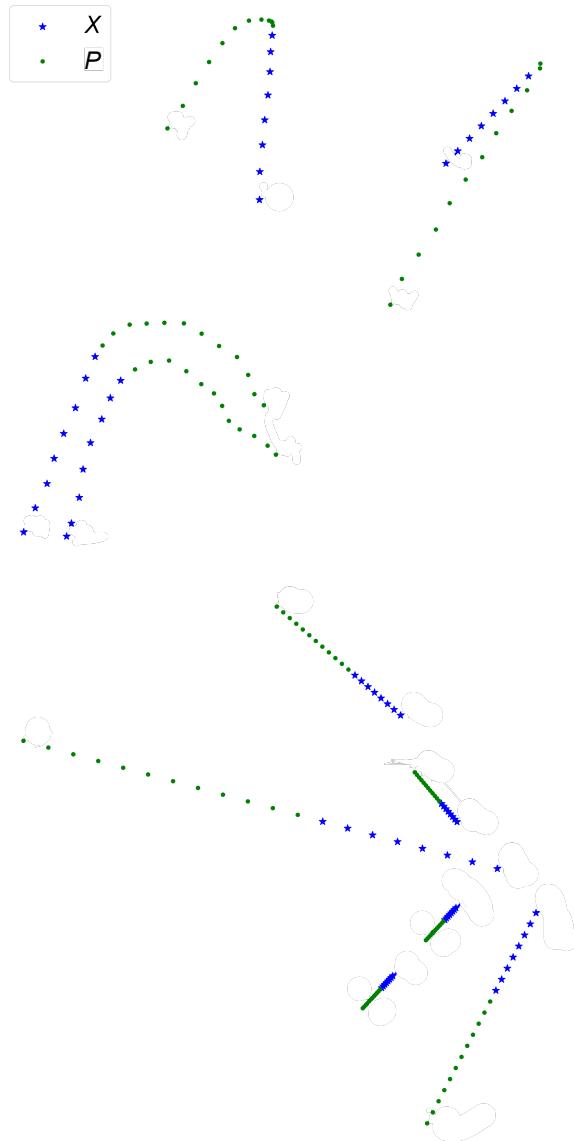


Figure 6.16: Examples of trajectories randomly sampled from THÖR’s training set for the *most challenging* cluster (**top**) and the *least challenging* cluster (**bottom**). The most challenging cluster encompasses non-linear trajectories, while the least challenging trajectories are predominantly linear.

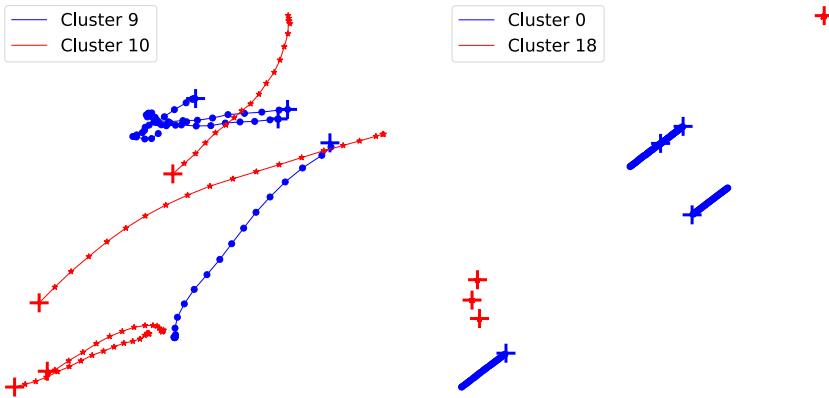


Figure 6.17: Examples of trajectories randomly sampled from the THÖR (**left**) and Argoverse (**right**) datasets, with crosses denoting the starting points of each tracklet. In THÖR, trajectories from cluster 9 predominantly move from right to left, while those from cluster 10 feature from left to right movements. In Argoverse, trajectories from cluster 0 are substantially longer compared to those from cluster 18.

be exploited in practical systems (see Secs. 7.2 and 7.3). A comprehensive empirical analysis of such predictors showed that:

- Clusters derived from future or full trajectory segments outperform those based on observed segments or observable classes, confirming their capacity to encode latent dynamics.
- Clusters derived from statistical features yield lower FDE than raw displacement-based representations, particularly in real-world data.
- Data-driven classes substantially improve predictions for complex, non-linear trajectories, such as those in the *zigzag-walker* and *random-walker* categories of the synthetic dataset in Sec. 6.3.2.

Furthermore, we proposed an alternative deep learning-based framework to traditional clustering methods, which are either limited in capturing misaligned temporal dynamics (K-means) or are computationally expensive (TS K-means). Our SC GAN leverages a GAN-based architecture underlying LSTM networks for trajectory data modeling and deep feature space clustering. Empirical results demonstrate that SC GAN outperforms traditional clustering techniques, especially under data drifting, as illustrated in the HOTEL dataset from the ETH/UCY benchmark [60, 76].

In summary, data-driven trajectory classes outperform semantically defined observable labels under ideal conditions where class labels from future states

are available. Our results highlight the potential of data-driven classes in enhancing trajectory prediction. We extend this work in Chapter 7 by integrating these data-driven classes into practical trajectory prediction systems.



## Chapter 7

# Trajectory Prediction with Data-driven Classes

*Everyone sees what you appear to be, few experience what you really are.*

— Niccolò Machiavelli

Data-driven contextual information provides a compelling alternative for overcoming the limitations of observable classes in trajectory prediction. As established in Chapter 6, data-driven classes derived from future or full trajectory segments offer informative representations that capture motion patterns and latent behavioral cues crucial to enhance trajectory predictions. The challenge, however, lies in how to effectively integrate these classes into trajectory prediction frameworks in a manner that is both robust and practically useful. In Chapter 6, we have also shown that future- and full-driven clusters can be used to improve the representation of the observed tracklet, but such clusters are not available during the inference phase. Specifically, future-driven clusters represent the distribution of future trajectories, while full-driven clusters are more holistic representations integrating both observed and future trajectory information. Therefore, this chapter presents two strategies for incorporating data-driven trajectory classes into trajectory prediction systems to enhance robustness, accuracy, and generalization, benefiting from the representation value of both future- and full-driven clusters. First, we address the limitations of context-agnostic GAN-based predictors, which often suffer from the *mode collapse* problem where the generator overfits dominant patterns and fails to capture the full diversity of the data distribution. To mitigate this issue, we introduce novel training strategies exploiting the future-driven clusters obtained from SC GAN (see Sec. 6.4.2). By employing a weighted loss function and balanced batch sampling informed by data-driven class distributions and prediction errors, we encourage broader coverage of trajectory modes and improved generalization. Second, we propose a multi-stage probabilistic pre-

diction framework that explicitly conditions generative models on full-driven trajectory classes. This approach comprises four stages: transforming trajectories to the displacement space, clustering full-driven trajectory states via full-driven SC GAN, training class-conditioned generative models, and ranking the generated predictions. To improve efficiency and scalability, we propose novel distance-based ranking techniques for trajectory selection, which outperform neural scoring modules in terms of computational cost while maintaining competitive accuracy. The framework outperforms class-agnostic baselines on datasets of both human and road agents and performs competitively against deterministic single-output models when evaluated on the most probable prediction. Together, these strategies provide a comprehensive study of how data-driven classes can be leveraged to improve trajectory prediction, balancing predictive performance, scalability, and representational clarity.

## 7.1 Introduction

### 7.1.1 Motivation and Contributions

In this thesis, data-driven classes can be derived from three trajectory segments: (1) the observed tracklet (*observation-driven clusters*), (2) the future segment (*future-driven clusters*), or (3) the full trajectory (*full-driven clusters*). As established in Chapter 6, clusters based solely on observed trajectories inherently reflect information already available to the predictor. Since trajectory prediction models are trained directly on this observed input, observation-driven clusters do not introduce novel priors or additional structure, and thus, this thesis does not further explore such clusters as part of practical trajectory predictors. In contrast, future-driven clusters provide valuable forward-looking information, capturing the diversity and structure of potential future outcomes. A predictor conditioned on these clusters benefits from privileged information on the multiple possible future trajectories. Building on these insights, we propose novel training strategies that force the learning of groups of trajectories that are hard to predict. Specifically, we leverage the future-driven clusters obtained from the Self-Conditioned GAN described in Sec. 6.4 to identify challenging modes in the data distribution and improve the training of a vanilla GAN-based predictor. This approach mitigates the mode collapse problem, enhancing the generator’s ability to cover broader trajectory patterns.

Full-driven clusters encompass information from observed and future trajectories, providing a more comprehensive representation of trajectory patterns. These clusters can act as a bridge, linking the observed tracklet to future outcomes. This connection is crucial for trajectory prediction, allowing the model to leverage a holistic representation of the trajectory data. In this context, we integrate full-driven clusters into a multi-stage probabilistic prediction framework, where the first stage involves clustering the full tra-

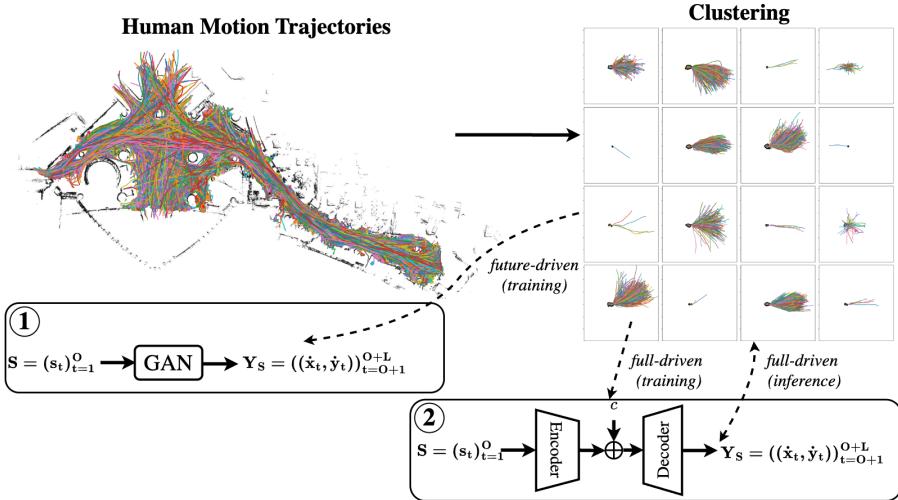


Figure 7.1: Data-driven classes for trajectory prediction: human motion trajectories are clustered into data-driven classes that are either (1) future-driven, used to guide the training of GAN-based predictors, or (2) full-driven, explicitly used to condition predictors during training. In the latter case, the clustering space is used to assign probabilities to the predicted trajectories during inference.

jectory states. This multi-stage approach is more effective than data-driven class-agnostic baselines in several trajectory prediction benchmarks.

In summary, the distinction between future- and full-driven clusters lies in their respective scopes: future-driven clusters focus on capturing the distribution of possible future trajectories, supporting multimodal trajectory prediction (see Sec. 1.2.2). In contrast, full-driven clusters integrate the context of the observed tracklet with future trajectory information, offering a more comprehensive view of the entire trajectory. Understanding the utility and trade-offs of these clustering approaches is critical for advancing trajectory prediction methods. This chapter studies future- and full-driven clusters for trajectory prediction. Fig. 7.1 illustrates the key contributions of this chapter:

- We show how future-driven clusters mitigate the *mode collapse* problem on GAN-based trajectory predictors, improving their ability to model diverse future trajectory patterns.
- We demonstrate how full-driven trajectory classes can seamlessly and explicitly integrate into a multi-stage probabilistic prediction framework.

### 7.1.2 Outline

This chapter is organized as follows. Sec. 7.2 introduces novel training strategies for GAN-based trajectory predictors, leveraging future-driven clusters to improve mode coverage and mitigate mode collapse. This section also details the experimental setup and provides empirical results demonstrating the effectiveness of the proposed strategies. Sec. 7.3 presents a multi-stage probabilistic prediction framework that explicitly conditions generative models on full-driven trajectory clusters. The section includes implementation details, evaluation methodology, and results that validate the performance and generalizability of the proposed approach. Finally, Sec. 7.4 summarizes the chapter’s key contributions and insights, outlining the broader implications of integrating data-driven trajectory classes into prediction systems.

## 7.2 Future-driven Clusters for Prediction

### 7.2.1 SC GAN for Diverse Prediction

GAN-based frameworks are prone to mode collapse, where the generator fails to capture less dominant or underrepresented modes in the data. Therefore, the goal is to improve the vanilla GAN training process by mitigating mode collapse and enhancing the generator’s ability to cover the most challenging modes from the input data distribution. While future-driven clusters may be challenging to integrate directly into end-to-end trajectory prediction, the training process of the future-driven Self-Conditioned GAN (FD SC GAN described in Sec. 6.4.2) offers valuable insights into the most challenging modes to recover. The clusters produced by the FD SC GAN divide the future trajectory data into distinct regions, offering a mechanism to identify regions where the predictor struggles to accurately recover the future of observed tracklets. Leveraging this information, we propose an improved vanilla GAN training approach that prioritizes the harder-to-predict examples identified by the FD SC GAN. In practice, we propose to penalize and sample more examples that are harder to predict during the vanilla GAN training. To guide this process, we define a set of *soft-assumptions* – “soft” because they may be erroneous due to the natural errors from FD SC GAN’s training – driven from FD SC GAN’s clustering space:

- We assume that FD SC GAN’s clusters group similar future ( $\mathbf{Y}_S$ ) based on trajectories’ state representation ( $S$ ).
- Clusters associated with higher prediction errors, such as Average and Final Displacement Errors (defined in Sec. 1.2.3), represent modes that are harder to recover and consequently require additional focus during training.

An overview of the entire process is depicted in Fig. 7.2, and we integrate the described assumptions into three training strategies for a vanilla GAN:

1. Weighted MSE loss (*wMSE*): This approach adjusts the generator’s Mean Squared Error term to emphasize trajectories from challenging subspaces representing modes less likely to be recovered. The weight applied to the MSE term in the generator’s loss of the trajectories in cluster  $i$ , initially defined in Eq. (4.5), is modified as follows:

$$\Lambda^{(i)} = \lambda_{\text{ADE}} \frac{\text{ADE}^{(i)}}{\text{ADE}_{\max}} + \lambda_{\text{FDE}} \frac{\text{FDE}^{(i)}}{\text{FDE}_{\max}} + \lambda_{\text{Dist}} \frac{\#^{(i)}}{\#_{\text{Tot}}}, \quad (7.1)$$

where  $\text{ADE}^{(i)}$  and  $\text{FDE}^{(i)}$  are the Average and the Final Displacement Errors of cluster  $i$ , respectively, obtained by the FD SC GAN;  $\text{ADE}_{\max} = \max_{j \in \{1, \dots, N_c\}} \{\text{ADE}^{(j)}\}$  and  $\text{FDE}_{\max} = \max_{j \in \{1, \dots, N_c\}} \{\text{FDE}^{(j)}\}$ ;  $\#^{(i)}$  and  $\#_{\text{Tot}}$  are the number of samples in cluster  $i$  and the total number of samples in the clustering space, respectively, mitigating the influence of outliers. Finally,  $\lambda_{\text{ADE}}$ ,  $\lambda_{\text{FDE}}$ , and  $\lambda_{\text{Dist}}$  are weights applied to the ADE, FDE, and clustering distribution terms.

2. Weighted batch sampler (*wB*): This strategy uses a multinomial distribution to sample trajectories during training, from which the probability of sampling trajectories from cluster  $i$  is given by:

$$p^{(i)} = \frac{\Lambda^{(i)}}{\sum_{j=1}^{N_c} \Lambda^{(j)}}, \quad (7.2)$$

where  $\Lambda^{(i)}$  is the weight assigned to cluster  $i$ , as defined in Eq. (7.1). *wB* ensures that harder-to-predict clusters are seen more often during training.

3. Combination of weighted MSE loss and batch sampler (*wMSE+wB*): This approach combines the weighted MSE loss and the weighted batch sampling strategies to adjust the loss function and data sampling process simultaneously.

## 7.2.2 Experiments

### Datasets, Baselines and Metrics

We evaluate our proposed methods on two distinct domains: the indoor human motion dataset THÖR [83], and the urban traffic dataset Argoverse [17], which contains road user trajectories. Both datasets and corresponding preprocessing are described in Sec. 6.4.4. To emphasize the learning of underrepresented

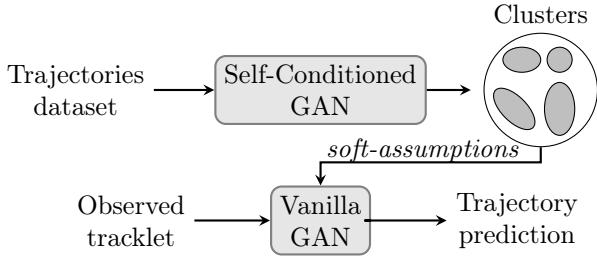


Figure 7.2: Proposed framework’s overview based on future-driven trajectory classes. First, FD SC GAN learns the future-driven clusters (or modes) of the input data. Then, this information is used as training settings (via *soft-assumptions*) to improve the prediction of specific modes.

agent classes in Argoverse, we applied a constraint when constructing the training set: 2600 trajectories from *av*, 2600 from *agents*, and 526 from *others* (factor of  $\approx 5$  times). We evaluate our proposed training configurations against two baseline approaches: the RED method [7], and a Vanilla GAN model inspired by Trajnet++ [52]. These baselines are comparable to those used in Chapter 4. We report the Top-1 ADE/FDE scores in meters in the test set as the average of five runs.

### 7.2.3 Quantitative Results

The Self-Conditioned GAN framework leverages clustering to capture similar and discriminative features in the trajectory data. Since it relies on K-means clustering (see Sec. 6.4.1), the number of clusters needs to be predefined. We determined this number through a grid search on the validation sets, selecting the configuration that maximized ADE performance. The number of clusters was 13 for the THÖR dataset and 19 for the Argoverse dataset.

In the first experiment, we show the intra-observed class performance of our approaches. Tab. 7.1 presents the prediction results for the evaluated methods, where observable classes in *italic* are the least predominant in the training set. According to the ADE and FDE metrics in the least representative agent classes, our approach based on the weighted batch sampler setting (GAN wB) and the combination of the two settings (GAN wMSE + wB) outperforms the baselines in both datasets. Therefore, the proposed training settings drive the generator to learn the most challenging unsupervised subspaces, enhancing the prediction of the least representative observable classes. We speculate that the trajectories of the least predominant agent classes lie on the most challenging clusters. Furthermore, in the THÖR dataset, the difference between the results for the *inspector* is not as clear as for the *workers* role. This difference might

Table 7.1: Intra-observable classes ADE/FDE metrics in the test sets. Bold values highlight superior performance of our methods in the least predominant observable classes.

| Dataset | Labels<br>(# samples) | Baselines      |                                                                            | Ours                                                     |                                                          |                                                                            |
|---------|-----------------------|----------------|----------------------------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------------------------|
|         |                       | LSTM<br>[7]    | GAN<br>[52]                                                                | GAN<br>wMSE                                              | GAN<br>wB                                                | GAN<br>wMSE + wB                                                           |
| THÖR    | workers<br>(413)      | 0.695<br>1.064 | 0.642 $\pm$ 0.006<br>1.033 $\pm$ 0.005                                     | 0.629 $\pm$ 0.005<br>1.009 $\pm$ 0.014                   | 0.644 $\pm$ 0.012<br>1.044 $\pm$ 0.028                   | <b>0.625 <math>\pm</math> 0.009</b><br><b>1.006 <math>\pm</math> 0.019</b> |
|         | visitors<br>(1379)    | 0.664<br>1.139 | 0.660 $\pm$ 0.001<br><b>1.105 <math>\pm</math> 0.090</b>                   | <b>0.657 <math>\pm</math> 0.003</b><br>1.107 $\pm$ 0.007 | 0.668 $\pm$ 0.005<br>1.124 $\pm$ 0.018                   | 0.657 $\pm$ 0.003<br>1.113 $\pm$ 0.013                                     |
|         | inspector<br>(260)    | 0.796<br>1.582 | 0.735 $\pm$ 0.007<br>1.474 $\pm$ 0.019                                     | 0.736 $\pm$ 0.008<br><b>1.473 <math>\pm</math> 0.013</b> | <b>0.729 <math>\pm</math> 0.013</b><br>1.479 $\pm$ 0.049 | 0.734 $\pm$ 0.003<br>1.476 $\pm$ 0.015                                     |
|         | others<br>(526)       | 1.864<br>3.029 | 1.815 $\pm$ 0.031<br>2.969 $\pm$ 0.034                                     | 1.799 $\pm$ 0.007<br>2.944 $\pm$ 0.022                   | <b>1.789 <math>\pm</math> 0.012</b><br>2.927 $\pm$ 0.020 | 1.801 $\pm$ 0.027<br><b>2.919 <math>\pm</math> 0.032</b>                   |
|         | av<br>(2600)          | 1.512<br>2.278 | <b>1.467 <math>\pm</math> 0.007</b><br><b>2.269 <math>\pm</math> 0.023</b> | 1.482 $\pm$ 0.009<br>2.292 $\pm$ 0.010                   | 1.480 $\pm$ 0.003<br>2.282 $\pm$ 0.006                   | 1.493 $\pm$ 0.010<br>2.298 $\pm$ 0.028                                     |
|         | agent<br>(2600)       | 2.371<br>4.690 | <b>2.349 <math>\pm</math> 0.012</b><br><b>4.654 <math>\pm</math> 0.016</b> | 2.362 $\pm$ 0.013<br>4.700 $\pm$ 0.029                   | 2.368 $\pm$ 0.020<br>4.721 $\pm$ 0.044                   | 2.371 $\pm$ 0.012<br>4.724 $\pm$ 0.027                                     |

be due to the dataset design since *workers* had to carry materials (e.g., boxes), whereas the *inspector* and *visitors* have much more similar motion patterns [83].

Tab. 7.2 presents the results for two trajectory groups based on the Self-Conditioned GAN’s clustering space. We report the performance for the most challenging and least predominant clusters (clusters 9 and 10 for THÖR and Argoverse, respectively), where the Self-Conditioned GAN shows the worst intra-cluster results, and for the most dominant clusters (clusters 0 and 18 for THÖR and Argoverse, respectively), which contain the largest number of samples. As expected, the results indicate a correlation between the number of trajectories within a cluster and the metrics achieved by each method: smaller clusters with fewer trajectories yield higher ADE/FDE scores, while dominant clusters with more trajectories achieve better prediction performance. That is, clusters with fewer trajectories present worse ADE/FDE scores, while dominant clusters present better prediction performance. Finally, our proposed training settings outperform the baselines in the most challenging clusters, supporting the described *soft assumptions* and achieving comparable results in the dominant trajectory groups.

Finally, Tab. 7.3 presents the overall results for both datasets. In the THÖR dataset, our weighted MSE training setting improved the average performance across both ADE and FDE metrics. However, in the Argoverse dataset, where the number of samples for the least representative class (*others*) was intentionally kept very small, the average scores did not improve. Nevertheless, our approach consistently enhanced the performance for the least dominant clus-

Table 7.2: ADE/FDE metrics for 2 clusters of the test set. Bold values indicate the superior performance of our methods in the least dominant clusters across both datasets, as well as in the most dominant cluster in the Argoverse dataset.

| Dataset   | Cluster ID<br>(# samples) | Baselines       |                                                                            | Ours                                    |                                         |                                                                             |
|-----------|---------------------------|-----------------|----------------------------------------------------------------------------|-----------------------------------------|-----------------------------------------|-----------------------------------------------------------------------------|
|           |                           | LSTM<br>[7]     | GAN<br>[52]                                                                | GAN<br>wMSE                             | GAN<br>wB                               | GAN<br>wMSE + wB                                                            |
| THÖR      | 9<br>(23)                 | 1.203<br>2.456  | 1.120 $\pm$ 0.025<br>2.758 $\pm$ 0.082                                     | 1.054 $\pm$ 0.048<br>2.505 $\pm$ 0.134  | 1.124 $\pm$ 0.038<br>2.811 $\pm$ 0.136  | <b>1.039 <math>\pm</math> 0.055</b><br><b>2.505 <math>\pm</math> 0.105</b>  |
|           | 0<br>(1003)               | 0.325<br>0.447  | <b>0.311 <math>\pm</math> 0.003</b><br><b>0.403 <math>\pm</math> 0.010</b> | 0.321 $\pm$ 0.004<br>0.419 $\pm$ 0.015  | 0.317 $\pm$ 0.007<br>0.416 $\pm$ 0.021  | 0.315 $\pm$ 0.002<br>0.424 $\pm$ 0.017                                      |
| Argoverse | 10<br>(16)                | 7.394<br>19.075 | 7.184 $\pm$ 0.178<br>18.402 $\pm$ 0.415                                    | 7.105 $\pm$ 0.123<br>18.233 $\pm$ 0.297 | 7.122 $\pm$ 0.055<br>18.276 $\pm$ 0.113 | <b>7.047 <math>\pm</math> 0.088</b><br><b>18.128 <math>\pm</math> 0.194</b> |
|           | 18<br>(1542)              | 0.912<br>1.148  | 0.809 $\pm$ 0.016<br>1.100 $\pm$ 0.017                                     | 0.807 $\pm$ 0.010<br>1.088 $\pm$ 0.027  | 0.805 $\pm$ 0.007<br>1.079 $\pm$ 0.012  | <b>0.795 <math>\pm</math> 0.008</b><br><b>1.055 <math>\pm</math> 0.030</b>  |

Table 7.3: ADE/FDE metrics in the test sets. Bold values indicate superior performance of our methods in the THÖR dataset.

| Dataset | Baselines      |                                                                            | Ours                                                                       |                                                          |                                                          |
|---------|----------------|----------------------------------------------------------------------------|----------------------------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------|
|         | LSTM<br>[7]    | GAN<br>[52]                                                                | GAN<br>wMSE                                                                | GAN<br>wB                                                | GAN<br>wMSE + wB                                         |
| THÖR    | 0.685<br>1.163 | 0.663 $\pm$ 0.002<br>1.123 $\pm$ 0.019                                     | <b>0.658 <math>\pm</math> 0.003</b><br><b>1.119 <math>\pm</math> 0.009</b> | 0.668 $\pm$ 0.003<br><b>1.119 <math>\pm</math> 0.009</b> | <b>0.658 <math>\pm</math> 0.002</b><br>1.122 $\pm$ 0.013 |
|         | 1.948<br>3.330 | <b>1.912 <math>\pm</math> 0.012</b><br><b>3.298 <math>\pm</math> 0.014</b> | 1.915 $\pm$ 0.012<br>3.307 $\pm$ 0.017                                     | 1.915 $\pm$ 0.009<br>3.310 $\pm$ 0.018                   | 1.923 $\pm$ 0.010<br>3.307 $\pm$ 0.014                   |

ters in both datasets, demonstrating its effectiveness in recovering challenging subspaces of the trajectory data.

#### 7.2.4 Qualitative Results

We present qualitative results of the prediction methods in Fig. 7.3, focusing on complex trajectories. In the proposed Self-Conditioned GAN framework, such complex trajectories pertain to small and non-dominant clusters. Our approach, leveraging the weighted loss function and batch sampling strategies, produces predictions that align more closely with the ground truth in these challenging scenarios. By leveraging the weighted objective function and batch sampling strategies, our approach prioritizes these complex trajectories during training, enabling the model to effectively address their intricacies and recover their modes, as demonstrated in Fig. 7.3.

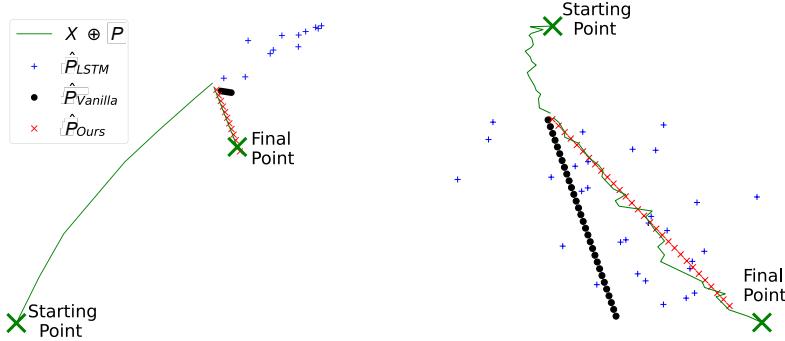


Figure 7.3: Examples of trajectory forecasting on the THÖR (**left**) and Argoverse (**right**) datasets for complex trajectories. Here,  $\mathbf{X}$  represents the observed 2D positions. The results demonstrate that our approach generates predictions that align more closely with the ground truth than the baselines.

## 7.3 Full-driven Clusters for Prediction

### 7.3.1 SC GAN for Probabilistic Prediction

The goal is to leverage full-driven SC GAN (FP SC GAN described in Sec. 6.4.3) and the resulting clusters to explicitly condition a trajectory predictor. We propose a model-agnostic multi-stage system to achieve a similar objective to class-conditioned methods (see Sec. 4.2): predicting the future of a given tracklet conditioned on a specific class. In this case, the classes are data-driven, representing groups of similar trajectories. Additionally, our system ( $\psi_{\text{D-TP}}$ ) ensures that predictions are probabilistically informed during inference. The task, referred to as D-TP (defined in Sec. 1.2), can be formally expressed as:

$$\psi_{\text{D-TP}} : \mathbf{S}_k, c_{\mathbf{S}_k \oplus \mathbf{Y}_{\mathbf{S}_k}} \mapsto \mathbf{Y}_{\mathbf{S}_k}, p_{\mathbf{S}_k \oplus \hat{\mathbf{Y}}_{\mathbf{S}_k}}, \quad (7.3)$$

where  $\mathbf{S}_k$  represents the observed tracklet,  $c_{\mathbf{S}_k \oplus \mathbf{Y}_{\mathbf{S}_k}}$  denotes the class of the full trajectory's states,  $\mathbf{Y}_{\mathbf{S}_k}$  is the future trajectory states, and  $p_{\mathbf{S}_k \oplus \hat{\mathbf{Y}}_{\mathbf{S}_k}}$  is the probability associated with the observed tracklet concatenated with the predicted future  $\mathbf{S}_k \oplus \hat{\mathbf{Y}}_{\mathbf{S}_k}$ .

To address this problem, we propose the multi-stage framework illustrated in Fig. 7.4. This framework begins by transforming the input trajectories' 2D positions into the displacement space, represented as the finite differences of the 2D positions. Having the input states ( $\mathbf{S}_k$ ) as the displacements (equivalent to the velocities), the predictor avoids dependence on the spatial context of the input data, facilitating transferability to new domains. In the second

step, the ground truth displacement vectors ( $\mathbf{S}_k \oplus \mathbf{Y}_{\mathbf{S}_k}$ ) are partitioned into  $N_c$  clusters using a clustering method, such as FP SC GAN. This clustering step is essential as it groups akin trajectories' states (in this case, the displacements), producing trajectory clusters, which then condition the prediction process. In the third step, we train conditional Deep Generative Models (cDGMs), such as conditional Variational Autoencoders (cVAEs) or conditional Generative Adversarial Networks (cGANs), to predict future displacements based on the observed states and their corresponding cluster class  $c_{\mathbf{S}_k \oplus \mathbf{Y}_{\mathbf{S}_k}}$  (i.e., the identifier for the cluster derived from the ground truth displacements). Here, we use cDGMs and corresponding training settings similar to the ones described in Sec. 4.3. As we cluster the entire trajectory's states, the cDGM establishes a connection between the prediction, the observed displacement tracklet, and the cluster class. This connection ensures that the predicted trajectory for a given cluster class  $c_{\mathbf{S}_k \oplus \mathbf{Y}_{\mathbf{S}_k}}$  is more similar to the trajectories within the same cluster than trajectories from other clusters. During inference, the system generates candidates for each cluster. Then, it employs a ranking method to assign probabilities to the proposed predictions, ensuring informed and accurate trajectory forecasts.

The final step of the system aims at finding the mapping:

$$\mathbf{S}_k \oplus \hat{\mathbf{Y}}_{\mathbf{S}_k} \mapsto p_{\mathbf{S}_k \oplus \hat{\mathbf{Y}}_{\mathbf{S}_k}}, \quad (7.4)$$

which assigns probabilities to the  $N_c \times n_{\text{inf}}$  predicted samples corresponding to  $n_{\text{inf}}$  predictions per cluster. This mapping ensures that samples from the correct cluster are assigned higher probabilities compared to those from other clusters. One approach to achieve this is to train a deep neural network to directly learn the mapping in Eq. (7.4), as demonstrated in [90, 21]. Alternatively, we propose using distance-based similarity measures, leveraging two methods: *centroids* and *neighbors*. We assume an inverse relationship between distance and similarity in the clustering space in both methods. For instance, smaller distances to a cluster's centroid indicate higher similarity to that cluster's samples, leading to a greater probability of belonging to it. Fig. 7.5 provides a visual representation of our method: assuming the ground truth cluster class is cluster 1 ( $\mathbf{Y}^{c(1)}$ ), the intuition is that the generated samples conditioned on the ground truth cluster ( $\hat{\mathbf{Y}}^{c(1)}$ ) are closer to cluster 1 (in blue) compared to the samples generated under other clusters and their respective conditioning cluster's centroid. Similarly, in the *neighbors* method, smaller average distances to the  $N_{\text{neig}}$  nearest neighbors of the same cluster increase the likelihood of a sample belonging to that cluster.

Formally, in distance-based methods, the probability of a predicted trajectory belonging to its conditioning cluster  $c^{(i)}$  is computed using a *soft-argmax* function over the inverse distances:

$$\hat{p}_{c^{(i)}} = \frac{\exp(\frac{1}{m_{c^{(i)}}}/\tau)}{\sum_j^{N_c} \exp(\frac{1}{m_{c^{(j)}}}/\tau)}, \quad (7.5)$$

where  $m$  represents the distance measure, and  $\tau$  is a temperature parameter controlling the sharpness of the probability distribution. In the *centroids* method,  $m_{c^{(i)}}$  is the L2-distance between the sample embeddings  $Enc_D(\mathbf{S}_k \oplus \hat{\mathbf{Y}}_{\mathbf{S}_k})$  and the centroid of cluster  $c^{(i)}$ . In the *neighbors* method,  $m_{c^{(i)}}$  is the average L2-distance to the  $N_{\text{neig}}$  nearest neighbors from cluster  $c^{(i)}$ . Alg. 2 summarizes our framework's training and inference pipelines.

### 7.3.2 Experiments

#### Datasets and Baselines

We evaluate our methods using two experimental setups: a train-test split and a leave-one-dataset-out approach. For the train-test split, we use two datasets: THÖR [83] and Argoverse [17], as described in Sec. 6.4.4. These datasets provide diverse scenarios representing indoor robotics and road-based environments, respectively. In the leave-one-dataset-out approach, we perform experiments on the widely adopted ETH/UCY benchmark [60, 76], as described in Sec. 6.4.4.

Furthermore, we conduct the experiments with baselines widely used in scientific works in the trajectory prediction field:

- Constant Velocity Model (*CVM*) [88]: Heuristic model that assumes that humans move with constant velocity and direction. In this work, we also include it for road agents' data (Argoverse). In [88], the velocity is given by the projection of the last displacement, but we use a weighted sum of the previous displacements based on a Gaussian kernel provided by [82]. Both methods achieve similar results.
- RED-LSTM predictor (*RED*) [7]: Deep learning-based model that stacks an LSTM (64 hidden dimensions) with a 2-layer MLP (hidden dimensions of 32 and 16). Inputs are linearly embedded displacements processed through a linear layer with 16 hidden dimensions, and a PreLU activation function is applied after each layer.
- GAN (*GAN*) and VAE (*VAE*) [52]: We adapted these models to remove mechanisms for social interaction modeling, focusing solely on trajectory-related information. These generative models, along with our proposed methods described in Sec. 6.4.1, share the same network configuration: an

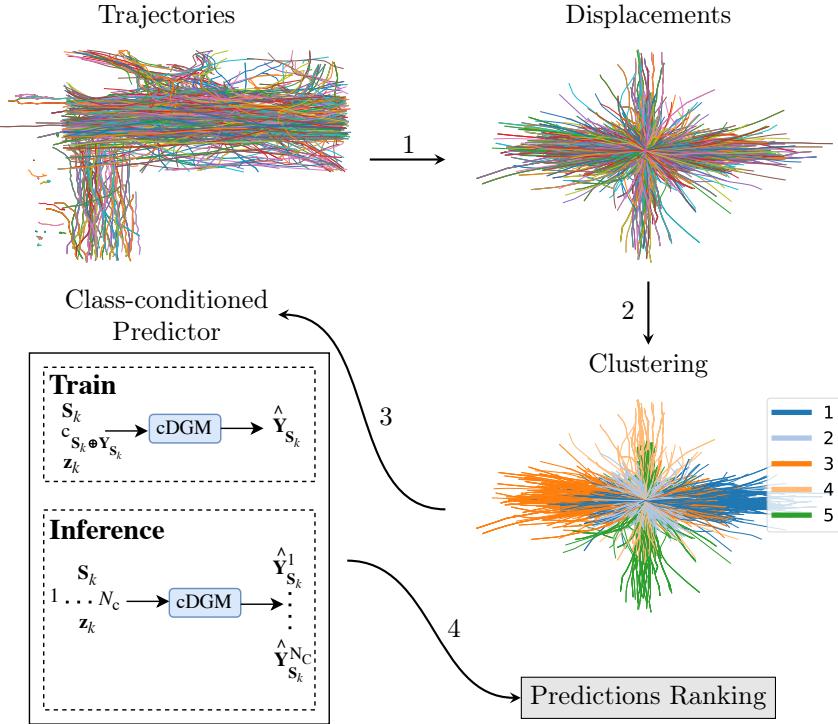


Figure 7.4: System overview. (1) We transform 2D positions into the displacement space (finite differences of positions). (2) We cluster the full-driven displacement data into  $N_c$  partitions. (3) We train a data-driven class-conditioned deep generative model that takes as input the trajectory’s states ( $\mathbf{S}_k$ ), the respective cluster class ( $c_k$ ), and white noise ( $\mathbf{z}_k$ ). During inference, the prediction model proposes  $N_c$  predictions. (4) The ranking step assigns probabilities to the predicted trajectories of each cluster.

initial linear layer with 16 hidden dimensions, an LSTM with 64 hidden dimensions, and a final linear layer with 32 hidden dimensions for decoding temporal features. To evaluate the ability of these models to generate a wide range of plausible trajectories, we incorporate the *k-varietiy loss* introduced in [37]. This loss function evaluates the sample closest to the ground truth out of  $k_{\text{variety}}$  generated trajectories, encouraging multi-modal diversity during training. In our experiments,  $k_{\text{variety}} = 3$ .

As described in Sec. 7.3.1, we compare two cDGMs: cVAE (*OURS-VAE*) and cGAN (*OURS-GAN*). Additionally, we evaluate three clustering algorithms: K-means, TS K-means, and the proposed FP SC GAN (see Sec. 6.4.3).

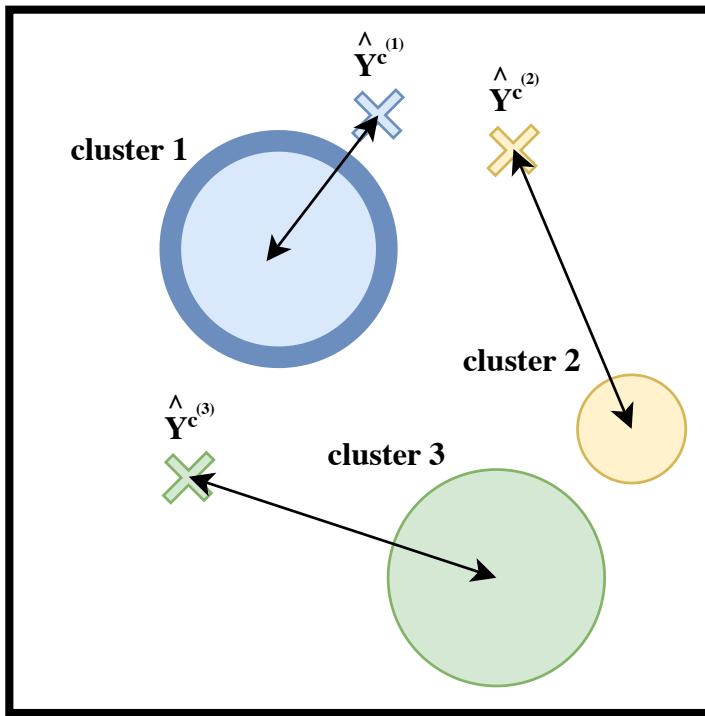


Figure 7.5: Illustration of our centroid distance-based method to rank trajectories during inference. The bold blue circle (cluster 1) represents the ground truth cluster. After training, samples generated conditioned on cluster 1 are closer to the corresponding centroid than those generated under other clusters and their corresponding conditioning cluster's centroid (in this case, clusters 2 or 3).

Furthermore, we have three predictions ranking methods: (1) based on the Euclidean distance to the centroids (*cent*); (2) based on the Euclidean distance to the  $N_{\text{neig}}$  closest neighbors (set to 20) in the displacement space and feature space, denoted as *neigh-ds* and *neigh-fs*, respectively. It is important to note here that *neigh-fs* is only used in FP SC GAN as it is the only method that makes use of a deep feature space; on the other hand, K-means and TS K-means require the predictions ranking based on the displacement space, as there is no deep feature space in these traditional methods; (3) based on the classification provided by the auxiliary network (*anet*). We evaluate the predictions ranking methods for one prediction per cluster class,  $n_{\text{inf}} = 1$ .

---

**Algorithm 2** Multi-stage framework for probabilistic trajectory prediction

---

**Require:** Training data  $\{(\mathbf{S}_k, \mathbf{Y}_{\mathbf{S}_k}, \mathbf{P}_{\mathbf{S}_k})\}_k$  (tracklets, futures, and ground truth positions), number of clusters  $N_c$

- 1: Initialize all components of  $\psi_{D-TP}$   $\triangleright$  Clustering and prediction methods.
- 2: **Step 1: Full-driven Clustering**
- 3: **for** each trajectory  $(\mathbf{S}_k, \mathbf{Y}_{\mathbf{S}_k})$  **do**
- 4:     Concatenate observed and future states:  $\mathbf{S}_k \oplus \mathbf{Y}_{\mathbf{S}_k}$ .
- 5:     Assign cluster label:  $c_k = \text{Cluster}(\mathbf{S}_k \oplus \mathbf{Y}_{\mathbf{S}_k})$ .
- 6:     Save cluster assignments  $\{c_k\}$ .
- 7: **end for**
- 8: **Step 2: Training the Generative Predictor**
- 9: **while** training not converged **do**
- 10:     Sample minibatch  $\{(\mathbf{S}_k, \mathbf{Y}_{\mathbf{S}_k}, \mathbf{P}_{\mathbf{S}_k}, c_k)\}_k$ .
- 11:     Update the generative model (GAN or VAE) to minimize the respective losses.
- 12: **end while**
- 13: **Step 3: Inference via Cluster Sampling**
- 14:     Initialize list of predictions  $\mathcal{P} \leftarrow \emptyset$
- 15:     **for**  $i \leftarrow 1$  to  $n_{inf}$  **do**  $\triangleright$  Sample  $n_{inf}$  futures from the generative model.
- 16:         **for**  $j \leftarrow 1$  to  $N_c$  **do**  $\triangleright$  Iterate over cluster classes.
- 17:             Generate prediction:

$$\hat{\mathbf{Y}}_{\mathbf{S}_k}^{c^{(j)}} = \text{Predictor}(\mathbf{S}_k, c^{(j)})$$

- 18:             Add  $\hat{\mathbf{Y}}_{\mathbf{S}_k}^{c^{(j)}}$  to prediction set  $\mathcal{P}$ .
- 19:         **end for**
- 20:     **end for**
- 21:     Rank predictions in  $\mathcal{P}$  with ou *centroids* or *neighbors* methods.
- 22: **return** Trained framework  $\psi_{D-TP}$  and ranked predictions  $\mathcal{P}$

---

## Metrics

The metrics used to evaluate the performance of the methods are based on Top- $K$  ADE and FDE, both measured in meters, as defined in Eqs. (1.1) and (1.2). These metrics enable a fair comparison between single-output predictors (*CVM* and *RED*) and stochastic multi-output models (*GAN*, *VAE*, *VAE-OURS*, and *GAN-OURS*). The evaluation framework includes:

- Top-3 ADE/FDE: This metric assesses the multimodal trajectory estimates produced by generative models. For *GAN* and *VAE*, three trajectories are sampled, and the one closest to the ground truth is evaluated. For *GAN-OURS* and *VAE-OURS*, the three most likely trajectories are selected, and the closest one to the ground truth is evaluated.

- Top-1 ADE/FDE: This metric compares point estimate predictors with stochastic multimodal models. For *GAN* and *VAE*, the first trajectory prediction is evaluated, while for *GAN-OURS* and *VAE-OURS*, the most likely trajectory is selected for evaluation.

To further assess the performance of the predictions ranking methods, we use accuracy, as defined in Eq. (1.3). This metric compares the outputs of the ranking methods to the soft labels derived from the clustering step, providing an additional perspective on the effectiveness of the probabilistic assignment and ranking strategies. The results for deep learning-based methods are averaged over five runs.

### 7.3.3 Quantitative Results

#### Prediction Results

Tab. 7.4 presents the Top-3 ADE/FDE results for deep generative approaches. Our proposed framework (*GAN-OURS* and *VAE-OURS*) demonstrates superior prediction performance across all datasets, underscoring the effectiveness of generating multiple plausible trajectories based on the conditioning cluster, regardless of the underlying prediction model (model-agnostic). This approach is more accurate than relying exclusively on the *k*-variety loss. Additionally, the results reveal that various clustering methods and predictions ranking mechanisms yield comparable performance across datasets. However, certain configurations have an edge in specific cases. For instance, in the THÖR dataset with *GAN-OURS*, the *cent* mechanism achieves ADE/FDE values of  $0.45 \pm 0.02$  and  $0.75 \pm 0.03$ , respectively. Similarly, in the ZARA2 dataset, the combination of TS K-means and *neig* achieves ADE/FDE values of  $0.37 \pm 0.01$  and  $0.62 \pm 0.01$ , respectively.

To assess the variability of the generative models against the predictions of single-output baselines, we present the Top-1 ADE/FDE in Tab. 7.5. Although the single-output deep learning-based baseline, *RED*, demonstrates the best overall results, our framework produces comparable scores in the ETH, HOTEL, and ZARA1 datasets. Our framework with the two prediction methods (*GAN-OURS* and *VAE-OURS*) outperforms their counterparts (with the same network structure) in Top-1 scores. Additionally, our approach effectively reduces the performance gap between Top-3 and Top-1 predictions, demonstrating its ability to mitigate the variability often seen in generative models [50]. An important advantage of our framework is its ability to assign meaningful probabilities to predicted trajectories, creating a probabilistic space of likely future locations rather than generating uninformative predictions. Moreover, while *CVM* performs competitively in human trajectory data, its performance declines substantially in road agent scenarios, where it struggles to account for speed variations in the trajectories. These results further emphasize the

Table 7.4: Top-3 ADE/FDE ( $\downarrow$ ) metrics in the test sets. Bold scores highlight superior performance of our framework for both prediction methods.

| Datasets  | <i>GAN</i>      | <i>GAN-OURS<sup>1</sup></i>       | <i>VAE</i>      | <i>VAE-OURS<sup>1</sup></i>       |
|-----------|-----------------|-----------------------------------|-----------------|-----------------------------------|
| THÖR      | $0.57 \pm 0.01$ | <b><math>0.53 \pm 0.01</math></b> | $0.62 \pm 0.02$ | <b><math>0.56 \pm 0.01</math></b> |
|           | $1.04 \pm 0.03$ | <b><math>0.84 \pm 0.03</math></b> | $1.05 \pm 0.05$ | <b><math>0.89 \pm 0.02</math></b> |
| Argoverse | $1.62 \pm 0.07$ | <b><math>1.56 \pm 0.02</math></b> | $1.96 \pm 0.02$ | <b><math>1.62 \pm 0.02</math></b> |
|           | $2.81 \pm 0.14$ | <b><math>2.69 \pm 0.02</math></b> | $3.44 \pm 0.06$ | <b><math>2.82 \pm 0.04</math></b> |
| ETH       | $0.84 \pm 0.03$ | <b><math>0.77 \pm 0.04</math></b> | $0.94 \pm 0.02$ | <b><math>0.82 \pm 0.02</math></b> |
|           | $1.64 \pm 0.06$ | <b><math>1.60 \pm 0.11</math></b> | $1.84 \pm 0.04$ | <b><math>1.60 \pm 0.04</math></b> |
| HOTEL     | $0.87 \pm 0.07$ | <b><math>0.81 \pm 0.10</math></b> | $1.08 \pm 0.04$ | <b><math>0.97 \pm 0.08</math></b> |
|           | $1.64 \pm 0.12$ | <b><math>1.46 \pm 0.11</math></b> | $1.95 \pm 0.06$ | <b><math>1.72 \pm 0.14</math></b> |
| UNIV      | $0.56 \pm 0.01$ | <b><math>0.51 \pm 0.01</math></b> | $0.61 \pm 0.01$ | <b><math>0.56 \pm 0.01</math></b> |
|           | $1.08 \pm 0.02$ | <b><math>0.98 \pm 0.03</math></b> | $1.16 \pm 0.01$ | <b><math>1.06 \pm 0.02</math></b> |
| ZARA1     | $0.43 \pm 0.02$ | <b><math>0.37 \pm 0.01</math></b> | $0.48 \pm 0.01$ | <b><math>0.44 \pm 0.01</math></b> |
|           | $0.82 \pm 0.07$ | <b><math>0.72 \pm 0.03</math></b> | $0.98 \pm 0.04$ | <b><math>0.91 \pm 0.03</math></b> |
| ZARA2     | $0.46 \pm 0.01$ | <b><math>0.40 \pm 0.01</math></b> | $0.49 \pm 0.01$ | <b><math>0.43 \pm 0.01</math></b> |
|           | $0.81 \pm 0.05$ | <b><math>0.65 \pm 0.03</math></b> | $0.86 \pm 0.05$ | <b><math>0.73 \pm 0.01</math></b> |

<sup>1</sup><sub>FP SC GAN + neig-fs</sub>

Table 7.5: Top-1 ADE/FDE ( $\downarrow$ ) metrics in the test sets. Bold scores highlight the superior performance of single-output methods for Top-1 predictions.

| Datasets  | <i>CVM</i>  | <i>RED</i>                        | <i>CF-GAN</i>   | <i>GAN-OURS<sup>1</sup></i> | <i>VAE</i>      | <i>VAE-OURS<sup>1</sup></i>       |
|-----------|-------------|-----------------------------------|-----------------|-----------------------------|-----------------|-----------------------------------|
| THÖR      | 0.79        | <b><math>0.65 \pm 0.01</math></b> | $0.85 \pm 0.09$ | $0.76 \pm 0.02$             | $0.71 \pm 0.02$ | $0.71 \pm 0.02$                   |
|           | 1.28        | <b><math>1.06 \pm 0.01</math></b> | $1.68 \pm 0.20$ | $1.31 \pm 0.05$             | $1.30 \pm 0.04$ | $1.21 \pm 0.04$                   |
| Argoverse | 2.57        | <b><math>1.84 \pm 0.01</math></b> | $2.41 \pm 0.09$ | $1.92 \pm 0.02$             | $2.69 \pm 0.02$ | $1.95 \pm 0.01$                   |
|           | 3.93        | <b><math>3.18 \pm 0.02</math></b> | $4.53 \pm 0.17$ | $3.27 \pm 0.03$             | $4.94 \pm 0.02$ | $3.39 \pm 0.04$                   |
| ETH       | <b>0.95</b> | <b><math>0.95 \pm 0.01</math></b> | $1.08 \pm 0.08$ | $0.96 \pm 0.03$             | $1.08 \pm 0.03$ | <b><math>0.95 \pm 0.02</math></b> |
|           | 2.11        | $1.94 \pm 0.02$                   | $2.18 \pm 0.19$ | $1.99 \pm 0.05$             | $2.15 \pm 0.08$ | $1.95 \pm 0.03$                   |
| HOTEL     | <b>0.42</b> | $1.03 \pm 0.04$                   | $1.03 \pm 0.05$ | $0.95 \pm 0.13$             | $1.18 \pm 0.05$ | $1.09 \pm 0.12$                   |
|           | <b>0.74</b> | $1.84 \pm 0.06$                   | $1.99 \pm 0.11$ | $1.72 \pm 0.19$             | $2.21 \pm 0.08$ | $1.96 \pm 0.18$                   |
| UNIV      | <b>0.65</b> | <b><math>0.65 \pm 0.01</math></b> | $0.74 \pm 0.03$ | $0.74 \pm 0.01$             | $0.71 \pm 0.01$ | $0.69 \pm 0.01$                   |
|           | 1.29        | <b><math>1.27 \pm 0.01</math></b> | $1.49 \pm 0.05$ | $1.44 \pm 0.04$             | $1.41 \pm 0.01$ | $1.36 \pm 0.01$                   |
| ZARA1     | 0.54        | <b><math>0.45 \pm 0.01</math></b> | $0.60 \pm 0.11$ | $0.47 \pm 0.01$             | $0.64 \pm 0.01$ | $0.53 \pm 0.01$                   |
|           | 1.05        | <b><math>0.87 \pm 0.01</math></b> | $1.21 \pm 0.24$ | $0.92 \pm 0.03$             | $1.33 \pm 0.03$ | $1.10 \pm 0.02$                   |
| ZARA2     | 0.55        | <b><math>0.50 \pm 0.01</math></b> | $0.65 \pm 0.02$ | $0.52 \pm 0.01$             | $0.60 \pm 0.02$ | $0.54 \pm 0.01$                   |
|           | 0.91        | <b><math>0.80 \pm 0.01</math></b> | $1.24 \pm 0.09$ | $0.86 \pm 0.04$             | $1.13 \pm 0.07$ | $0.93 \pm 0.01$                   |

<sup>1</sup><sub>FP SC GAN + neig-fs</sub>

robustness of our proposed deep learning-based methods in diverse environments.

Table 7.6: Accuracy ( $\uparrow$ ) of the predictions ranking methods in the test sets of the train/test split setting.

| Clustering method | Ranking method  | THÖR                              | Argoverse                         |
|-------------------|-----------------|-----------------------------------|-----------------------------------|
| K-means           | <i>cent</i>     | $0.83 \pm 0.01$                   | <b><math>0.95 \pm 0.01</math></b> |
|                   | <i>neigh-ds</i> | $0.81 \pm 0.01$                   | $0.94 \pm 0.01$                   |
|                   | <i>anet</i>     | <b><math>0.87 \pm 0.01</math></b> | <b><math>0.95 \pm 0.01</math></b> |
| TS K-means        | <i>cent</i>     | $0.66 \pm 0.01$                   | $0.94 \pm 0.01$                   |
|                   | <i>neigh-ds</i> | $0.66 \pm 0.01$                   | $0.95 \pm 0.01$                   |
|                   | <i>anet</i>     | <b><math>0.73 \pm 0.01</math></b> | <b><math>0.96 \pm 0.01</math></b> |
| FP SC GAN         | <i>cent</i>     | $0.64 \pm 0.06$                   | <b><math>0.97 \pm 0.01</math></b> |
|                   | <i>neigh-fs</i> | $0.68 \pm 0.03$                   | $0.96 \pm 0.05$                   |
|                   | <i>anet</i>     | <b><math>0.70 \pm 0.01</math></b> | $0.95 \pm 0.01$                   |

### Predictions Ranking Results

Tab. 7.6 presents the accuracy of the predictions ranking methods in the test sets of the train/test split settings (THÖR and Argoverse). The results demonstrate that the accuracy of these ranking methods directly impacts both the Top-3 and Top-1 ADE/FDE scores reported in Tab. 7.4 and Tab. 7.5, respectively. A general observation is that higher ranking accuracy correlates with improved ADE/FDE scores, particularly for Top-1 metrics. Within each clustering method and dataset, *anet* achieves superior accuracy only in the THÖR dataset across all clustering methods. We speculate this is because THÖR’s clusters are closer to each other, as the participants operated within the same environment, featuring similar motion patterns across the train and test sets. Conversely, in Argoverse, the behavioral diversity among different agent types (e.g., vehicles and other road agents) leads to more distinct clusters, reducing *anet*’s relative advantage.

In the leave-one-dataset-out setting (see Tab. 7.7), where datasets inherently show diverse behaviors due to the train/test splits, our proposed distance-based methods (*cent*, *neigh-ds*, and *neigh-fs*) outperform the auxiliary neural network (*anet*) in most cases across various clustering methods. While the constant-width MLP-based *anet* provides accurate estimates, its computational complexity scales as  $\mathcal{O}(N_L H_U^2)$ , where  $N_L$  is the number of layers, and  $H_U$  is the number of hidden units per layer. In contrast, distance-based methods such as *centroids* and *neighbors* require only  $\mathcal{O}(N_c)$  and  $\mathcal{O}(N_c N_{\text{neig}})$  computations, respectively, where  $N_c$  is the number of clusters and  $N_{\text{neig}}$  is the number of neighbors. Thus, distance-based methods offer computational advantages beyond their comparable accuracy to *anet*. They eliminate the need for training an additional neural network and achieve inference in linear time, whereas auxiliary networks operate in quadratic time.

Table 7.7: Accuracy ( $\uparrow$ ) of the predictions ranking methods in the test sets in the leave-one-dataset-out setting.

| Clustering method | Ranking method  | ETH                               | HOTEL                             | UNIV                              | ZARA1                             | ZARA2                             |
|-------------------|-----------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| K-means           | <i>cent</i>     | $0.95 \pm 0.01$                   | <b><math>1.00 \pm 0.00</math></b> | <b><math>0.84 \pm 0.01</math></b> | $0.93 \pm 0.01$                   | $0.94 \pm 0.01$                   |
|                   | <i>neigh-ds</i> | <b><math>0.98 \pm 0.01</math></b> | $0.99 \pm 0.01$                   | $0.81 \pm 0.01$                   | $0.90 \pm 0.01$                   | <b><math>0.96 \pm 0.01</math></b> |
|                   | <i>anet</i>     | $0.93 \pm 0.02$                   | $0.99 \pm 0.01$                   | $0.83 \pm 0.01$                   | <b><math>0.94 \pm 0.01</math></b> | $0.95 \pm 0.01$                   |
| TS K-means        | <i>cent</i>     | $0.95 \pm 0.01$                   | <b><math>1.00 \pm 0.00</math></b> | <b><math>0.87 \pm 0.01</math></b> | <b><math>0.93 \pm 0.01</math></b> | <b><math>0.95 \pm 0.01</math></b> |
|                   | <i>neigh-ds</i> | <b><math>0.98 \pm 0.01</math></b> | $0.99 \pm 0.00$                   | $0.82 \pm 0.01$                   | $0.91 \pm 0.01$                   | <b><math>0.95 \pm 0.01</math></b> |
|                   | <i>anet</i>     | $0.95 \pm 0.01$                   | $0.98 \pm 0.01$                   | $0.84 \pm 0.01$                   | <b><math>0.93 \pm 0.01</math></b> | <b><math>0.95 \pm 0.01</math></b> |
| FP SC GAN         | <i>cent</i>     | $0.70 \pm 0.11$                   | $0.79 \pm 0.09$                   | <b><math>0.65 \pm 0.02</math></b> | <b><math>0.91 \pm 0.04</math></b> | <b><math>0.91 \pm 0.04</math></b> |
|                   | <i>neigh-fs</i> | <b><math>0.75 \pm 0.05</math></b> | <b><math>0.91 \pm 0.06</math></b> | <b><math>0.65 \pm 0.03</math></b> | $0.90 \pm 0.05$                   | $0.89 \pm 0.02$                   |
|                   | <i>anet</i>     | $0.69 \pm 0.08$                   | $0.83 \pm 0.08$                   | $0.56 \pm 0.03$                   | <b><math>0.91 \pm 0.03</math></b> | $0.90 \pm 0.03$                   |

### 7.3.4 Qualitative Results

In Fig. 7.6, we present qualitative examples of the Top-3 predictions from our proposed methods and the baselines across the THÖR (left), Argoverse (center) and ZARA1 (right) test sets. The THÖR example illustrates a particularly challenging scenario described by a sharp heading change. Despite this complexity, our method successfully captures this uncommon behavior within the three most probable trajectories, assigning a likelihood of  $p = 0.26$ . The most probable trajectory ( $p = 0.48$ ) aligns with the movement’s general trend, which is reasonable given the most likely constant velocity profile of human walking [88].

In the Argoverse example, the auxiliary network shapes hierarchical predictions, yielding more information about our models’ predictions. The Top-1 prediction ( $p_1 = 0.29$ ) is also the closest to the ground truth. The second most likely trajectory ( $p_2 = 0.23$ ) follows the same direction but with a shorter distance, while the third most likely trajectory ( $p_3 = 0.16$ ) diverges in direction but remains a plausible alternative. Finally, in the ZARA2 example, while the baseline *GAN* fails to fully capture the static behavior, our framework successfully generates this underrepresented behavior and assigns it the highest probability.

## 7.4 Conclusions

This chapter presented prediction frameworks that leverage data-driven trajectory classes to improve trajectory prediction diversity and accuracy. Building upon the SC GAN framework developed in Sec. 6.4.2, we exploited unsupervised clustering in the discriminator’s feature space to discover distinct behavioral modes, each corresponding to unique trajectory characteristics. Using these future-driven clusters, we proposed three novel training strategies incorporating FD SC GAN-derived information into the prediction process. Our experimental results demonstrated that these strategies outperform baseline

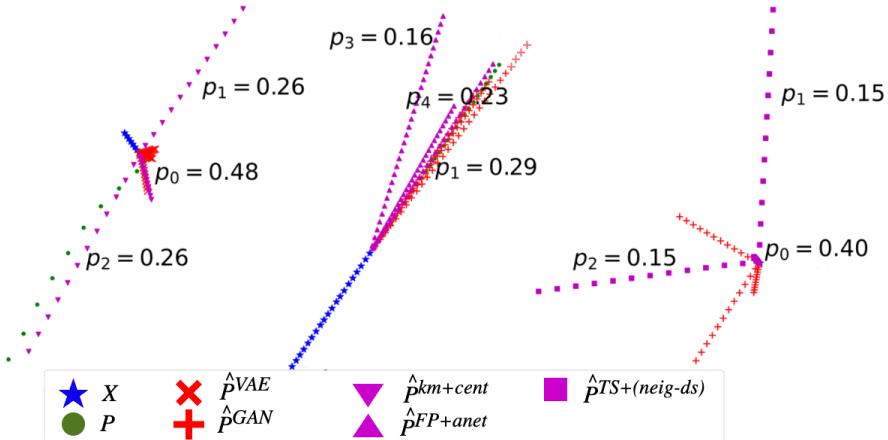


Figure 7.6: Top-3 predictions in test samples from THÖR (**left**), Argoverse (**center**), and ZARA2 (**right**).  $p_i \in [0, N_c - 1]$  the probabilities provided by our predictions ranking methods.

models, particularly for predicting the most challenging clusters while maintaining competitive performance across the broader distribution of trajectories. In addition, FD SC GAN generates effective trajectory clusters and provides valuable insights into the structure and impact of these clusters on the forecasting task. We further extended the utility of data-driven classes by introducing a multi-stage probabilistic prediction framework based on full-driven clustering. This system operates through four key stages: displacement space transformation, clustering through full-driven SC GAN (FP SC GAN), data-driven class-conditioned trajectory prediction, and final prediction ranking. Transforming trajectories into displacement space removes dependencies on absolute spatial positioning, enhancing generalizability across different environments and deployment contexts. For the clustering stage, we employed FP SC GAN, which combines the strengths of clustering and generative modeling to produce holistic trajectory classes. For ranking and selecting predictions, we introduced efficient distance-based scoring mechanisms that outperformed auxiliary neural network approaches, offering substantial computational advantages by scaling linearly with the number of candidates rather than quadratically as an MLP. By combining these components, our system underlying VAE- and GAN-based models surpassed their counterparts equipped with k-variety loss in Top-3 ADE/FDE scores while achieving comparable or superior Top-1 ADE/FDE scores. Our experimental results across diverse datasets validate the accuracy and robustness of the proposed framework.

Finally, this chapter contributed two frameworks: (1) training strategies for GAN-based predictors that enhance diversity and mitigate the mode collapse

problem and (2) a multi-stage prediction framework that efficiently integrates data-driven classes to produce more accurate and probabilistically calibrated trajectory forecasts. By integrating meaningful data-driven trajectory classes and employing efficient distance-based mechanisms, our frameworks establish a solid foundation for advancing trajectory forecasting and modeling in heterogeneous data settings.

# Chapter 8

## Conclusions

*Education must enable one to sift and weigh evidence, to discern the true from the false, the real from the unreal, and the facts from the fiction.*  
— Martin Luther King Jr., *The Purpose of Education*

Understanding and predicting heterogeneous human motion is a fundamental capability for robots and intelligent systems deployed in complex, dynamic environments. In industrial settings, autonomous mobile robots must navigate safely and efficiently while sharing space with human workers, adapting their behavior to diverse agent roles, tasks, or activities. In autonomous driving scenarios, accurately forecasting the trajectories of surrounding agents, including vulnerable road users such as pedestrians and cyclists, is crucial for safety and decision-making. Similarly, in smart home environments, intelligent systems rely on trajectory understanding and prediction to anticipate human behavior and respond proactively.

This thesis investigates trajectory heterogeneity by modeling how differences in agent roles, activities, and motion patterns influence and improve prediction accuracy. The final chapter summarizes the key contributions of this work in Sec. 8.1 and outlines promising directions for future research in Sec. 8.2 aimed at further advancing our knowledge of trajectory classes for trajectory prediction in human-centered environments.

### 8.1 Thesis Contributions and Summary

This thesis investigates trajectory heterogeneity in human-centered environments, addressing the whole pipeline from data collection to integrating observable and data-driven classes into trajectory prediction frameworks. The central concept introduced is trajectory classes, which are groups of trajectories that share common characteristics. These classes can be defined either through semantic, human-interpretable labels (observable classes) or learned

directly from trajectory data (data-driven classes). The overarching research goal is to leverage both trajectory classes to enhance the analysis and prediction of human motion.

A central contribution of this thesis is the development of novel datasets that enable the study of heterogeneous human behavior in robot-shared environments. We introduce the THÖR-MAGNI dataset in Chapter 3, a large-scale, richly annotated dataset recorded in a mock industrial facility. It captures diverse human activities, including group navigation, object transportation, and interactions with a mobile robot. We found that this dataset enables the study of observable classes and their influence on trajectory prediction.

Building on THÖR-MAGNI, we investigate the representational power of observable classes defined by human roles in our dataset for trajectory prediction. We propose class-conditioned deep learning baselines based on LSTM, Transformer, GAN, and VAE architectures using the methods in Chapter 4. We evaluated these models across imbalanced and low-data regimes in both indoor (THÖR-MAGNI) and outdoor (Stanford Drone Dataset) environments. Our results show that observable classes generally improve predictive accuracy. Moreover, in low-data regimes or class-imbalanced scenarios, pattern-based approaches such as Maps of Dynamics outperform deep learning models, highlighting the need for data-efficient prediction strategies. These findings inform a model selection framework tailored to data availability and class distribution in heterogeneous scenarios.

Despite the capability demonstrated by observable classes, they have also shown limitations. They are statically assigned to agents and remain constant across all associated trajectories, even when the underlying behaviors vary. This semantic ambiguity arises when a single observable class contains multiple distinct motion behaviors, or when similar behaviors appear across different class labels, reducing their predictive discriminability. To mitigate this, in Chapter 5, we extend THÖR-MAGNI with frame-level action annotations, resulting in THÖR-MAGNI *Act*. These fine-grained action labels provide rich contextual cues that augment the predictor’s input state. When integrated through multi-task learning or direct conditioning, they reduce semantic ambiguity and enhance prediction accuracy by capturing intra-agent behavioral variability.

The thesis also explores data-driven classes: trajectory clusters derived directly from trajectory data without relying on external perception. In Chapter 6, we propose Self-Conditioned GAN (SC GAN), a generative framework that learns meaningful trajectory embeddings through self-conditioning. These embeddings form trajectory clusters that provide privileged information for downstream predictors. By exposing models to rare yet semantically relevant modes, such as stopping behavior, SC GAN helps address challenges like mode collapse in generative models with the approaches described in Sec. 7.2.

In Sec. 7.3, we further extend the clustering of trajectory data to develop a multi-stage probabilistic prediction framework, which leverages clusters de-

rived from entire trajectory sequences to generate, condition, and rank multiple future trajectory predictions. Although these clusters depend on future states and are unavailable during inference, we introduce lightweight ranking mechanisms that leverage their benefits at test time. This framework enables robust, probabilistically informed predictions while avoiding the complexity of explicit perception-based class detection.

In summary, the key contributions of this thesis are:

- **The THÖR-MAGNI dataset:** A novel dataset capturing heterogeneous human motion in industrial settings, enabling analysis of observable classes.
- **Class-conditioned deep learning trajectory predictors analysis:** LSTM-, Transformer-, VAE-, and GAN-based predictors that efficiently and effectively incorporate class labels, tested across imbalanced and low-data regimes.
- **Action-augmented state representation for trajectory modeling:** THÖR-MAGNI *Act*, an extension of THÖR-MAGNI with fine-grained action annotations that improve predictive accuracy and mitigate class ambiguity. Efficient and effective action-conditioned predictors and joint action and trajectory prediction methods based on the Transformer architecture.
- **Self-Conditioned GAN for learning data-driven classes:** a generative model that discovers trajectory clusters used to enhance GAN-based predictors.
- **Multi-stage probabilistic prediction framework:** a clustering-based prediction strategy leveraging full trajectory clusters to improve inference-time prediction via ranking mechanisms.

Collectively, these contributions provide a principled and empirically validated framework for incorporating trajectory classes, both observable and data-driven, into human trajectory prediction. Specifically, in this work, we demonstrate how understanding and exploiting motion trajectory heterogeneity can lead to more accurate, generalizable, and context-aware trajectory predictors for intelligent systems deployed in human-centered environments.

## 8.2 Future Work

While this thesis presents a comprehensive foundational investigation of trajectory heterogeneity through observable and data-driven classes, several open questions and research challenges remain. Integrating trajectory classes into predictive frameworks inherently involves trade-offs between interpretability,

scalability, and robustness. Observable classes offer high interpretability by relying on human-labeled categories but depend on upstream perception systems that may introduce noise or fail in complex environments. In contrast, data-driven classes depend solely on the trajectory data, providing flexible and adaptive representations, yet often lack semantic clarity.

Future research should investigate these trade-offs, with particular attention to hybrid approaches that combine the interpretability of observable classes with the representational power of data-driven classes. We elaborate on this idea in Sec. 8.2.1. Additionally, observable classes rely on perception and detection systems, which raise concerns about robustness under sensor noise or misclassification. The impact of such failures on predictive performance should be systematically evaluated, as discussed in Sec. 8.2.2.

Finally, while this thesis explored data-driven classes derived from fixed trajectory segments (e.g., observed, future, or entire trajectories), a promising extension involves learning trajectory classes across variable time horizons, which would allow for the discovery of fine-grained, temporally localized motion patterns that are not constrained by predefined and static trajectory segmentation. We outline this direction in Sec. 8.2.3.

### 8.2.1 Hybrid Observable and Data-driven Class Conditioning

Observable classes encode human-understandable semantics that support interpretable reasoning in trajectory prediction. For instance, one may reason: “If the observed trajectory shows this pattern and the agent is a pedestrian, then it is likely to continue in this manner.” Such reasoning enables the incorporation of prior knowledge about agent types (e.g., pedestrians typically move more slowly than vehicles and tend to remain on sidewalks), providing privileged information that can enhance both the safety, reliability, and accuracy of prediction and planning processes.

In contrast, data-driven classes are derived directly from motion cues. They are agnostic to semantic categories and thus better capture motion-specific patterns, such as stopping behavior (see Sec. 7.3) that may not be distinguishable through observable classes alone. These representations are less ambiguous and can capture subtle variations in motion that static observable labels may overlook.

Given their complementary strengths, combining observable and data-driven trajectory classes presents an opportunity to unify semantic interpretability with motion-based expressiveness. Specifically, we propose a hierarchical conditioning strategy in which clustering is applied within each observable class. This approach yields hybrid class labels that encode human-interpretable semantic attributes and motion pattern similarity, offering a richer and more discriminative input for trajectory prediction models. Fig. 8.1 illustrates this

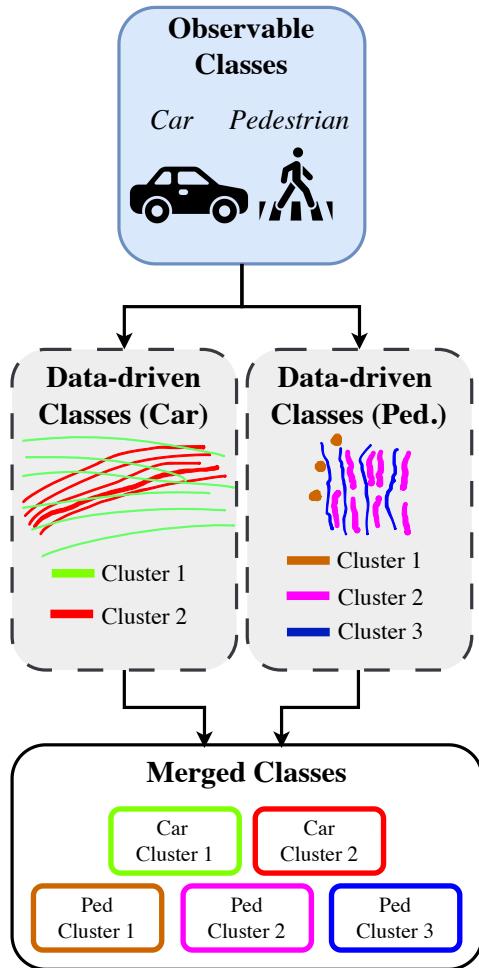


Figure 8.1: From top to bottom: clustering separately each observable class yields hybrid class labels that retain semantic interpretability while capturing data-driven patterns.

process, which can be seamlessly integrated into the prediction frameworks developed in this thesis.

### 8.2.2 Robustness of Trajectory Classes

While trajectory classes have demonstrated strong potential to enhance predictive performance, their effectiveness in real-world systems depends critically on robustness to noise, detection errors, and data imperfections.

Observable classes, in particular, rely on upstream perception systems (e.g., class detection, tracking, and activity recognition pipelines) to assign semantic labels such as “pedestrian” or “worker”. These systems are often subject to domain shifts, occlusions, and sensor noise, which lead to misclassifications or inconsistent class assignments. For instance, an individual carrying an object might be misclassified as a different role, or an agent might switch classes erroneously due to detection jitter. Since trajectory predictors condition on these labels, their misclassification can propagate through the system, degrading prediction accuracy or even leading to unsafe behavior in robotics applications.

In parallel, trajectory data is often affected by noise in measurement acquisition, including errors in localization and delayed or missing detections. Such issues impact data-driven class discovery (where noise can distort clustering) and, consequently, the performance of the predictors trained on such data.

Future work should include a systematic robustness analysis to support the deployment of trajectory class-aware predictors in real-world systems. This includes evaluating how prediction performance degrades under increasing levels of:

- **Perception-induced class noise** by simulating misclassification rates in observable class assignments;
- **Trajectory noise** is achieved by injecting perturbations into the observed motion data and quantifying their effect on both class detection and prediction accuracy.

Such studies would provide insights into the sensitivity of trajectory class-based models and inform the development of robust architectures that can either tolerate noisy inputs or adaptively reweight uncertain signals. Furthermore, this analysis could motivate the integration of uncertainty-aware mechanisms, e.g., Bayesian inference or confidence-weighted conditioning, to mitigate the impact of unreliable trajectory class signals.

### 8.2.3 Time-Granularity in Data-driven Classes

Throughout this thesis, data-driven trajectory classes have been defined at the granularity of entire trajectory segments or fixed observation/prediction windows. While this formulation captures broad motion patterns, it overlooks the temporal evolution of motion patterns within a single trajectory. In practice,

human trajectories often comprise a succession of distinct motion primitives, such as stopping, accelerating, turning, or weaving, that unfold over time.

To address this, future work should explore the temporal decomposition of trajectories into short-horizon segments, each associated with a data-driven label representing a localized motion pattern. This fine-grained decomposition would allow trajectory prediction models to treat *motion* as a temporally structured sequence of latent behavioral modes rather than a single global category (see Fig. 8.2). Moreover, such decomposition must account for the non-uniform temporal spacing of motion transitions, as real-world motion patterns may vary in duration and timing. In summary, time-granular representation introduces several advantages, such as (1) fine-grained decomposition, where predictors can identify and respond to local behaviors such as an upcoming stop or turn, (2) improved generalization, as shorter segments are more generalizable than longer segments across agents and contexts, supporting transfer learning and modular prediction, and (3) greater realism and interpretability, by reflecting the inherently uneven temporal structure of human motion.

This notion of time-varying data-driven labels aligns conceptually with the use of fine-grained action labels explored in Chapter 5, where the goal is to capture the dynamic nature of human behavior in a finer representation. Both represent temporally evolving annotations over the trajectory sequence, where each time step or segment reflects a potentially distinct motion intent or behavioral mode. However, while action labels are manually defined and rely on perception modules, time-varying data-driven labels offer a fully unsupervised, trajectory-centric alternative. Consequently, time-varying data-driven labels are robust to perceptual noise and adaptable to diverse contexts without requiring task-specific action semantics.

To enable this paradigm, future research must address several open challenges. First, robust methods for unsupervised segmentation and clustering of short-horizon motion primitives are needed. Unlike full-trajectory clustering, this task requires accurate temporal partitioning to ensure that motion patterns are well-isolated and semantically meaningful. Second, researchers should adapt predictive frameworks to model sequences of short-term behaviors by extending action-conditioned models (see Chapter 5) to operate over sequences of learned data-driven primitives.

Ultimately, time-granular data-driven classes can augment the state representation of trajectory predictors in a manner analogous to action labels and may be further integrated into hybrid class-conditioning frameworks as described in Sec. 8.2.1. Fine-grained data-driven labels would enable more expressive, context-aware predictors capable of capturing the fine-grained variability of real-world motion.

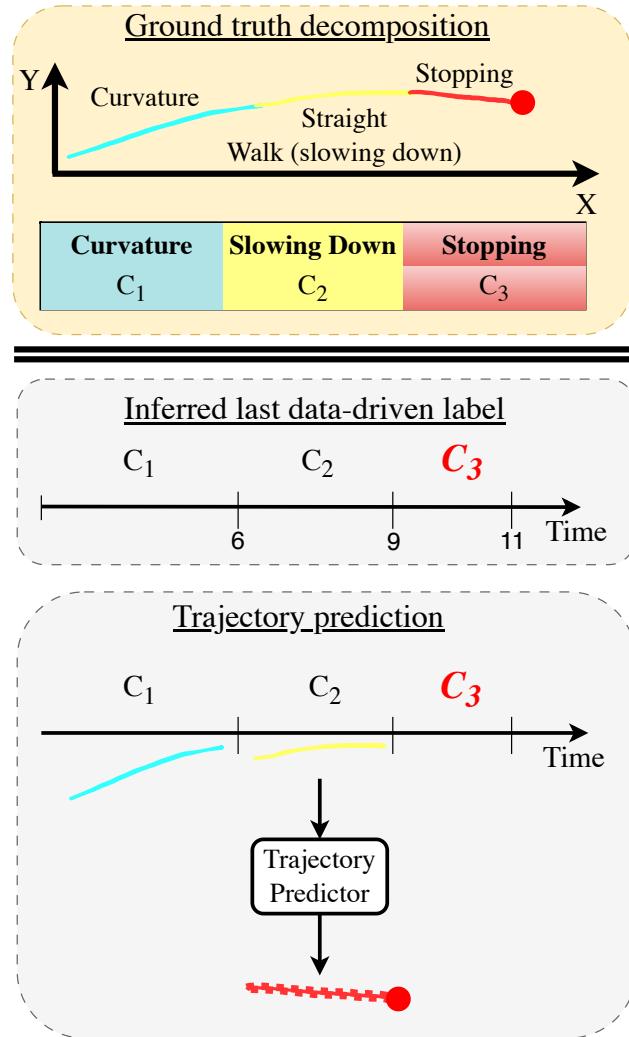


Figure 8.2: Overview of trajectory prediction with temporally decomposed data-driven labels. **Top:** Ground-truth sequence of data-driven labels, representing short-term trajectory patterns. **Middle:** Inferred data-driven label based on an unevenly spaced temporal decomposition. **Bottom:** Predicted future trajectory conditioned on the observed trajectory and the associated sequence of data-driven labels, including the most recent inferred label.

# References

- [1] Javad Amirian, Bingqing Zhang, Francisco Valente Castro, Juan José Baldeolmar, Jean-Bernard Hayet, and Julien Pettré. Opentraj: Assessing prediction complexity in human trajectories datasets. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi, editors, *Computer Vision – ACCV 2020*, pages 566–582, Cham, 2021. Springer International Publishing. (Cited on pages 52 and 54.)
- [2] Josh Andle, Nicholas Soucy, Simon Socolow, and Salimeh Yasaei Sekeh. The stanford drone dataset is more complex than we think: An analysis of key characteristics. *IEEE Transactions on Intelligent Vehicles*, 8(2):1863–1873, 2023. (Cited on page 29.)
- [3] Rabbia Asghar, Manuel Diaz-Zapata, Lukas Rummelhard, Anne Spalanzani, and Christian Laugier. Vehicle motion forecasting using prior information and semantic-assisted occupancy grid maps. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 49–54, 2023. (Cited on pages 34 and 37.)
- [4] Inhwan Bae, Jean Oh, and Hae-Gon Jeon. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10017–10029, October 2023. (Cited on pages 33 and 75.)
- [5] Inhwan Bae, Young-Jae Park, and Hae-Gon Jeon. Singulartrajectory: Universal trajectory predictor using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17890–17901, June 2024. (Cited on page 33.)
- [6] Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International Journal of Social Robotics*, 1(1):71–81, 2009. (Cited on page 49.)
- [7] Stefan Becker, Ronny Hug, Wolfgang Hübner, and Michael Arens. Red: A simple but effective baseline predictor for the trajnet benchmark. In

- European Conference on Computer Vision Workshops*, 2018. (Cited on pages 62, 140, 141, 142, and 145.)
- [8] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, pages 3457–3464. IEEE, 2011. (Cited on page 32.)
  - [9] Toufik Benmessabih, Rim Slama, Vincent Havard, and David Baudry. Online human motion analysis in industrial context: A review. *Engineering Applications of Artificial Intelligence*, 131:107850, 2024. (Cited on page 88.)
  - [10] 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. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1929–1934. IEEE, 2020. (Cited on page 40.)
  - [11] Dražen Bršić, Takayuki Kanda, Tetsushi Ikeda, and Takahiro Miyashita. Person tracking in large public spaces using 3-D range sensors. *IEEE Transactions on Human-Machine Systems*, 43(6):522–534, 2013. (Cited on pages 30, 32, and 40.)
  - [12] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. (Cited on page 33.)
  - [13] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 86–99. PMLR, 30 Oct–01 Nov 2020. (Cited on page 33.)
  - [14] Rohan Chandra, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8475–8484, 2019. (Cited on pages 14, 34, and 60.)
  - [15] Rohan Chandra, Tianrui Guan, Srujan Panuganti, Trisha Mittal, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Forecasting

- trajectory and behavior of road-agents using spectral clustering in graph-lstms. *IEEE Robotics and Automation Letters*, 2020. (Cited on page 33.)
- [16] Rohan Chandra, Tianrui Guan, Srujan Panuganti, Trisha Mittal, Utтарan Bhattacharya, Aniket Bera, and Dinesh Manocha. Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms. *IEEE Robotics and Automation Letters*, 5(3):4882–4890, 2020. (Cited on page 125.)
- [17] Ming-Fang Chang, John 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8740–8749, 2019. (Cited on pages 29, 30, 104, 124, 139, and 145.)
- [18] George Charalambous, Sarah Fletcher, and Philip Webb. The development of a scale to evaluate trust in industrial human-robot collaboration. *International Journal of Social Robotics*, 8:193–209, 4 2016. (Cited on page 49.)
- [19] Guangyi Chen, Junlong Li, Nuoxing Zhou, Liangliang Ren, and Jiwen Lu. Personalized trajectory prediction via distribution discrimination. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15580–15589, October 2021. (Cited on pages 35, 36, and 37.)
- [20] Jiahe Chen, Jinkun Cao, Dahua Lin, Kris M. Kitani, and Jiangmiao Pang. MGF: Mixed gaussian flow for diverse trajectory prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. (Cited on pages 35, 36, and 37.)
- [21] Weihuang Chen, Zhigang Yang, Lingyang Xue, Jinghai Duan, Hongbin Sun, and Nanning Zheng. Multimodal pedestrian trajectory prediction using probabilistic proposal network. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2022. (Cited on pages 20, 36, 37, and 144.)
- [22] Yuying Chen, Congcong Liu, Bertram E. Shi, and Ming Liu. Comogen: Coherent motion aware trajectory prediction with graph representation. In *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020. (Cited on page 33.)
- [23] Yuxiang Cui, Haodong Zhang, Yue Wang, and Rong Xiong. Learning world transition model for socially aware robot navigation. In *2021*

- IEEE International Conference on Robotics and Automation (ICRA)*, pages 9262–9268, 2021. (Cited on pages 14 and 60.)
- [24] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. (Cited on pages 12 and 113.)
- [25] Tiago Rodrigues de Almeida and Oscar Martinez Mozos. Likely, light, and accurate context-free clusters-based trajectory prediction. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1269–1276, 2023. (Cited on pages 7, 33, 36, and 37.)
- [26] Tiago Rodrigues de Almeida, Andrey Rudenko, Tim Schreiter, Yufei Zhu, Eduardo Gutierrez Maestro, Lucas Morillo-Mendez, Tomasz P. Kucner, Oscar Martinez Mozos, Martin Magnusson, Luigi Palmieri, Kai O. Arras, and Achim J. Lilienthal. Thor-magni: Comparative analysis of deep learning models for role-conditioned human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2200–2209, October 2023. (Cited on page 35.)
- [27] Tiago Rodrigues de Almeida, Yufei Zhu, Andrey Rudenko, Tomasz P. Kucner, Johannes A. Stork, Martin Magnusson, and Achim J. Lilienthal. Trajectory prediction for heterogeneous agents: A performance analysis on small and imbalanced datasets. *IEEE Robotics and Automation Letters*, 9(7):6576–6583, 2024. (Cited on pages 8, 16, 35, 37, 40, 42, and 94.)
- [28] Patrick Dendorfer, Aljoša Ošep, and Laura Leal-Taixé. Goal-GAN: Multimodal Trajectory Prediction Based on Goal Position Estimation. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi, editors, *Computer Vision – ACCV 2020*, pages 405–420, Cham, 2021. Springer International Publishing. (Cited on page 54.)
- [29] Christian Dondrup, Nicola Bellotto, Ferdian Jovan, Marc Hanheide, et al. Real-time multisensor people tracking for human-robot spatial interaction. *ICRA ’15 Workshop on Machine Learning for Social Robotics*, 2015. (Cited on page 32.)
- [30] Mahsa Ehsanpour, Fatemeh Sadat Saleh, Silvio Savarese, Ian D. Reid, and Hamid Rezatofighi. JRDB-act: A large-scale dataset for spatio-temporal action, social group and activity detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20951–20960, 2022. (Cited on pages 29, 32, 40, and 88.)
- [31] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases

- with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996. (Cited on page 33.)
- [32] Mark Nicholas Finean, Luka Petrovič, Wolfgang Merkt, Ivan Markovič, and Ioannis Havoutis. Motion planning in dynamic environments using context-aware human trajectory prediction. *Robotics and Autonomous Systems*, 166:104450, 2023. (Cited on page 32.)
- [33] Maosi Geng, Junyi Li, Chuangjia Li, Ningke Xie, Xiqun Chen, and Der-Horng Lee. Adaptive and simultaneous trajectory prediction for heterogeneous agents via transferable hierarchical transformer network. *IEEE Transactions on Intelligent Transportation Systems*, 24(10):11479–11492, 2023. (Cited on pages 15, 34, 35, 37, and 60.)
- [34] Harshayu Girase, Haiming Gang, Srikanth Malla, Jiachen Li, Akira Kanehara, Karttikeya Mangalam, and Chiho Choi. Loki: Long term and key intentions for trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9803–9812, October 2021. (Cited on page 88.)
- [35] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10335–10342, 2021. (Cited on page 62.)
- [36] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2672–2680, 2014. (Cited on pages 11 and 12.)
- [37] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. (Cited on pages 35 and 146.)
- [38] Marah Halawa, Olaf Hellwich, and Pia Bideau. Action-based contrastive learning for trajectory prediction. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 143–159, Cham, 2022. Springer Nature Switzerland. (Cited on pages 35 and 37.)
- [39] Sandra G Hart. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*,

- volume 50.9, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006. (Cited on page 49.)
- [40] Sandra G Hart and Lowell E Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988. (Cited on page 49.)
  - [41] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. (Cited on pages 11 and 12.)
  - [42] Ronny Hug, Stefan Becker, Wolfgang Hübner, and Michael Arens. Quantifying the complexity of standard benchmarking datasets for long-term human trajectory prediction. *IEEE Access*, 9:77693–77704, 2021. (Cited on page 33.)
  - [43] Boris Ivanovic, Kuan-Hui Lee, Pavel Tokmakov, Blake Wulfe, Rowan McAllister, Adrien Gaidon, and Marco Pavone. Heterogeneous-agent trajectory forecasting incorporating class uncertainty. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12196–12203, 2022. (Cited on pages 34 and 37.)
  - [44] Simindokht Jahangard, Zhixi Cai, Shiki Wen, and Hamid Rezatofighi. Jrdb-social: A multifaceted robotic dataset for understanding of context and dynamics of human interactions within social groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. (Cited on page 30.)
  - [45] Miao Kang, Jingwen Fu, Sanping Zhou, Songyi Zhang, and Nanning Zheng. Learning to predict diverse trajectory from human motion patterns. *Neurocomputing*, 504:123–131, 2022. (Cited on pages 20, 36, and 37.)
  - [46] Suhyeon Kim, Haecheong Park, and Junghye Lee. Word2vec-based latent semantic analysis (w2v-lsa) for topic modeling: A study on blockchain technology trend analysis. *Expert Systems with Applications*, 152:113401, 2020. (Cited on page 75.)
  - [47] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. (Cited on page 70.)
  - [48] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. (Cited on pages 11 and 12.)

- [49] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990. (Cited on page 33.)
- [50] 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*. Curran Associates Inc., Red Hook, NY, USA, 2019. (Cited on page 149.)
- [51] Parth Kothari and Alexandre Alahi. Safety-compliant generative adversarial networks for human trajectory forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 24(4):4251–4261, 2023. (Cited on pages 10, 11, 62, 64, and 120.)
- [52] Parth Kothari, Sven Kreiss, and Alexandre Alahi. Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 2022. (Cited on pages 54, 70, 92, 94, 124, 140, 141, 142, and 145.)
- [53] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125. IEEE Press, 2018. (Cited on page 29.)
- [54] Philipp Kratzer, Simon Bihlmaier, Niteesh Balachandra Midlagajni, Rohit Prakash, Marc Toussaint, and Jim Mainprice. Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze. *IEEE Robotics and Automation Letters (RAL)*, 2020. (Cited on pages 30, 32, and 40.)
- [55] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, Nov 2016. (Cited on page 60.)
- [56] Tomasz Piotr Kucner, Martin Magnusson, Sariah Mghames, Luigi Palmieri, Francesco Verdoja, Chittaranjan Srinivas Swaminathan, Tomáš Krajiník, Erik Schaffernicht, Nicola Bellotto, Marc Hanheide, and Achim J Lilienthal. Survey of maps of dynamics for mobile robots. *The International Journal of Robotics Research (IJRR)*, 42(11):977–1006, 2023. (Cited on pages 15, 60, and 69.)
- [57] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. (Cited on page 121.)
- [58] Doi Thi Lan and Seokhoon Yoon. Trajectory clustering-based anomaly detection in indoor human movement. *Sensors*, 23(6), 2023. (Cited on page 33.)

- [59] 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2165–2174, 2017. (Cited on page 35.)
- [60] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007. (Cited on pages 29, 30, 32, 104, 125, 132, and 145.)
- [61] Timm Linder and Kai O. Arras. *People Detection, Tracking and Visualization Using ROS on a Mobile Service Robot*, pages 187–213. Springer International Publishing, Cham, 2016. (Cited on page 3.)
- [62] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. NTU RGB+D 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019. (Cited on page 40.)
- [63] Mingyu Liu, Ekim Yurtsever, Jonathan Fossaert, Xingcheng Zhou, Walter Zimmer, Yuning Cui, Bare Luka Zagar, and Alois C. Knoll. A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook. *IEEE Transactions on Intelligent Vehicles*, pages 1–29, 2024. (Cited on page 29.)
- [64] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14274–14283, 2020. (Cited on pages 17, 118, and 121.)
- [65] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinrong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7573–7582, 2021. (Cited on page 33.)
- [66] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. (Cited on pages 17, 19, 33, 75, and 103.)
- [67] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6120–6127, 2019. (Cited on pages 15, 34, 37, and 60.)
- [68] Barbara Majecka. Statistical models of pedestrian behaviour in the forum. *Master's thesis, School of Informatics, University of Edinburgh*, 2009. (Cited on pages 30 and 32.)

- [69] Srikanth Malla, Behzad Dariush, and Chiho Choi. Titan: Future forecast using action priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. (Cited on pages 13, 40, and 88.)
- [70] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. (Cited on page 29.)
- [71] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014. (Cited on page 118.)
- [72] Xiaoyu Mo, Zhiyu Huang, Yang Xing, and Chen Lv. Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 23(7), 2022. (Cited on pages 15, 34, 35, 37, and 60.)
- [73] Matteo Munaro and Emanuele Menegatti. Fast RGB-D people tracking for service robots. *Autonomous Robots*, 37:227–242, 2014. (Cited on pages 30 and 32.)
- [74] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR 2011*, pages 3153–3160. IEEE, 2011. (Cited on pages 29 and 32.)
- [75] Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. (Cited on page 107.)
- [76] S. Pellegrini, A. Ess, K. Schindler, and L. 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, 2009. (Cited on pages 29, 30, 32, 104, 125, 132, and 145.)
- [77] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. (Cited on page 120.)
- [78] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and

- trajectory prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6261–6270, 2019. (Cited on pages 29 and 88.)
- [79] 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, 2016. (Cited on pages 13, 29, 30, 32, 40, 60, and 66.)
- [80] Tiago Rodrigues de Almeida, Eduardo Gutierrez Maestro, and Oscar Martinez Mozos. Context-free self-conditioned gan for trajectory forecasting. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1218–1223, 2022. (Cited on pages 8, 13, 14, 18, 33, and 34.)
- [81] Tiago Rodrigues de Almeida, Tim Schreiter, Andrey Rudenko, Luigi Palmieri, Johannes A. Stork, and Achim J. Lilienthal. THÖR-MAGNI act: Actions for human motion modeling in robot-shared industrial spaces. In *Proceedings of the 2025 ACM/IEEE International Conference on Human-Robot Interaction, HRI ’25*, pages 1083–1087. IEEE Press, 2025. (Cited on pages 8 and 31.)
- [82] Andrey Rudenko, Wanting Huang, Luigi Palmieri, Kai O Arras, and Achim J Lilienthal. Atlas: a benchmarking tool for human motion prediction algorithms. In *Robotics: Science and Systems (RSS) Workshop on Social Robot Navigation*, 2021. (Cited on pages 124 and 145.)
- [83] Andrey Rudenko, Tomasz P. Kucner, Chittaranjan S. Swaminathan, Ravi T. Chadalavada, Kai O. Arras, and Achim J. Lilienthal. THÖR: Human-robot navigation data collection and accurate motion trajectories dataset. *IEEE Robotics and Automation Letters*, 5(2):676–682, 2020. (Cited on pages 13, 30, 32, 41, 43, 52, 104, 124, 139, 141, and 145.)
- [84] 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. (Cited on page 40.)
- [85] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision – ECCV 2020*, pages 683–700. Springer International Publishing, 2020. (Cited on pages 11, 54, and 62.)
- [86] Tim Schreiter, Tiago Rodrigues de Almeida, Yufei Zhu, Eduardo Gutierrez Maestro, Lucas Morillo-Mendez, Andrey Rudenko, Luigi Palmieri,

- Tomasz P. Kucner, Martin Magnusson, and Achim J. Lilienthal. THÖRMAGNI: A large-scale indoor motion capture recording of human movement and robot interaction, 2024. (Cited on pages 3, 7, 30, 35, 60, and 66.)
- [87] Tim Schreiter, Lucas Morillo-Mendez, Ravi T Chadalavada, Andrey Rudenko, Erik Billing, Martin Magnusson, Kai O Arras, and Achim J Lilienthal. Advantages of Multimodal versus Verbal-Only Robot-to-Human Communication with an Anthropomorphic Robotic Mock Driver. In *32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2023. (Cited on page 57.)
- [88] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. What the constant velocity model can teach us about pedestrian motion prediction. *IEEE Robotics and Automation Letters*, 5(2):1696–1703, 2020. (Cited on pages 127, 145, and 152.)
- [89] Tianmin Shu, Dan Xie, Brandon Rothrock, Simisa Todorovic, and Song-Chun Zhu. Joint inference of groups, events and human roles in aerial videos. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4576–4584, 2015. (Cited on pages 29 and 32.)
- [90] Jianhua Sun, Yuxuan Li, Haoshu Fang, and Cewu Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13230–13239, 2021. (Cited on pages 20, 33, 36, 37, and 144.)
- [91] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tslearn, a mach. learning toolkit for time series data. *Journal of Mach. Learning Research*, 21(118):1–6, 2020. (Cited on pages 17, 19, 103, and 109.)
- [92] Tobii AB. Tobii Pro Lab User Manual v1.217. [https://go.tobii.com/tobii\\_pro\\_lab\\_user\\_manual](https://go.tobii.com/tobii_pro_lab_user_manual), Accessed: 2024-02-02. (Cited on page 90.)
- [93] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, pages 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. (Cited on pages 11, 12, and 94.)

- [94] Edward Vendrow, Duy Tho Le, Jianfei Cai, and Hamid Rezatofighi. Jrdb-pose: A large-scale dataset for multi-person pose estimation and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. (Cited on page 29.)
- [95] Ao Wang, Hui Chen, Lihao Liu, Kai CHEN, Zijia Lin, Jungong Han, and Guiguang Ding. YOLOv10: Real-time end-to-end object detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. (Cited on page 35.)
- [96] Yuning Wang, Pu Zhang, Lei Bai, and Jianru Xue. Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1400–1409, 2023. (Cited on pages 33, 36, and 37.)
- [97] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemeyer Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Advances in Neural Information Processing Systems*, 2021. (Cited on pages 3, 7, 13, 29, and 40.)
- [98] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6488–6497, June 2022. (Cited on page 33.)
- [99] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 511–528, Cham, 2022. Springer Nature Switzerland. (Cited on page 33.)
- [100] Hao Xue, Du Q. Huynh, and Mark Reynolds. Poppl: Pedestrian trajectory prediction by lstm with automatic route class clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):77–90, 2021. (Cited on page 33.)
- [101] Zhi Yan, Tom Duckett, and Nicola Bellotto. Online learning for human classification in 3D LiDAR-based tracking. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 864–871. IEEE, 2017. (Cited on pages 30 and 32.)
- [102] Zhi Yan, Simon Schreiberhuber, Georg Halmetschlager, Tom Duckett, Markus Vincze, and Nicola Bellotto. Robot perception of static and

- dynamic objects with an autonomous floor scrubber. *Intelligent Service Robotics*, 13(3):403–417, 2020. (Cited on page 32.)
- [103] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 376–394, Cham, 2022. Springer Nature Switzerland. (Cited on page 54.)
- [104] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2022. (Cited on page 88.)
- [105] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a Mixture model of Dynamic pedestrian-Agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2871–2878, 2012. (Cited on page 32.)
- [106] Yufei Zhu, Andrey Rudenko, Tomasz P Kucner, Achim J Lilienthal, and Martin Magnusson. A data-efficient approach for long-term human motion prediction using maps of dynamics. In *IEEE International Conference on Robotics and Automation Workshop (LHMP)*, 2023. (Cited on page 69.)
- [107] Yufei Zhu, Andrey Rudenko, Tomasz P. Kucner, Luigi Palmieri, Kai O. Arras, Achim J. Lilienthal, and Martin Magnusson. Cliff-lhmp: Using spatial dynamics patterns for long- term human motion prediction. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3795–3802, 2023. (Cited on pages 15, 60, 62, and 69.)