

Trajectory Prediction for Heterogeneous Agents: A Performance Analysis on Small and Imbalanced Datasets

Tiago Rodrigues de Almeida¹, Yufei Zhu¹, Andrey Rudenko²,
Tomasz P. Kucner³, Johannes A. Stork¹, Martin Magnusson¹, Achim J. Lilienthal^{1,4}

Abstract—Robots and other intelligent systems navigating in complex dynamic environments should predict future actions and intentions of surrounding agents to reach their goals efficiently and avoid collisions. The dynamics of those agents strongly depends on their tasks, roles, or observable labels. Class-conditioned motion prediction is thus an appealing way to reduce forecast uncertainty and get more accurate predictions for heterogeneous agents. However, this is hardly explored in the prior art, especially for mobile robots and in limited data applications. In this paper, we analyse different class-conditioned trajectory prediction methods on two datasets. We propose a set of conditional pattern-based and efficient deep learning-based baselines, and evaluate their performance on robotics and outdoors datasets (THÖR-MAGNI and Stanford Drone Dataset). Our experiments show that all methods improve accuracy in most of the settings when considering class labels. More importantly, we observe that there are significant differences when learning from imbalanced datasets, or in new environments where sufficient data is not available. In particular, we find that deep learning methods perform better on balanced datasets, but in applications with limited data, e.g., cold start of a robot in a new environment, or imbalanced classes, pattern-based methods may be preferable.

Index Terms—Human and Humanoid Motion Analysis and Synthesis, Human Detection and Tracking, Datasets for Human Motion, Deep Learning Methods

I. INTRODUCTION

RELIABLE and safe robot navigation in dynamic human-centered environments relies on anticipating the future behavior of other agents. In several domains, including autonomous driving (AD) and industrial mobile robotics, the

Manuscript received: February, 13, 2024; Revised April, 27, 2024; Accepted May, 23, 2024. This paper was recommended for publication by Editor Gentiane Venture upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) and by the EU Horizon 2020 No. 101017274 (DARKO). *Tiago Rodrigues de Almeida and Yufei Zhu contributed equally to this work. (Corresponding author: Tiago Rodrigues de Almeida.)*

¹Tiago Rodrigues de Almeida, Yufei Zhu, Johannes A. Stork, Martin Magnusson, and Achim J. Lilienthal are with the Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden {tiago.almeida, yufei.zhu, johannesandreas.stork, martin.magnusson, achim.lilienthal}@oru.se

²Andrey Rudenko is with the Robert Bosch GmbH, Corporate Research, Stuttgart, Germany andrey.rudenko@de.bosch.com

³Tomasz P. Kucner is with the School of Electrical Engineering Aalto University and with the Finnish Center for Artificial Intelligence (FCAI), Finland tomasz.kucner@aalto.fi

⁴Achim J. Lilienthal is also with the Technical University of Munich, Germany. achim.j.lilienthal@tum.de

Digital Object Identifier (DOI): see top of this page.

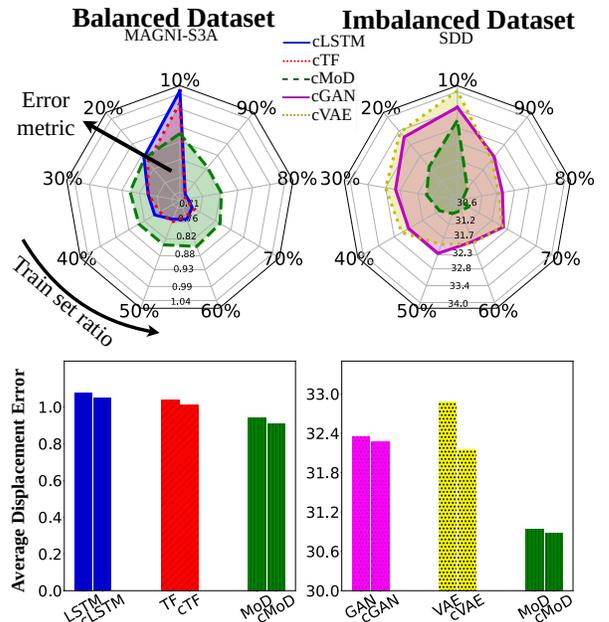


Fig. 1: Average Displacement Error of class-conditioned trajectory prediction methods across balanced and imbalanced datasets. **Top:** In balanced datasets, at low data regimes (10% train set ratio), the pattern-based method (cMoD) is more accurate than deep learning methods (**left**). In imbalanced datasets, cMoD is more accurate than deep generative models (**right**). Thus, the pattern-based method, cMoD, is more suitable for low training data regimes and imbalanced datasets than its deep learning counterparts. **Bottom:** Class-conditioned methods (c***) consistently outperform their unconditional counterparts across both datasets.

motion planner must proactively consider the future positions of many heterogeneous agents to maintain safety standards [1]. In autonomous urban driving, various entities with distinct dynamic patterns, such as pedestrians, cyclists, and cars, navigate in a shared space. In industrial environments, humans engaged in different tasks, such as transporting objects, interacting with robots, or walking in groups, may also present different motion patterns within the same spatial layout [2]. The diversity of agents and their corresponding motion patterns pose significant challenges to current trajectory predictors [3], leading to high prediction uncertainty, lower prediction accuracy, and thus to overly conservative motion planning [4].

In the context of mobile robots in dynamic environments, prior art has hardly explored class-aware prediction approaches, which is particularly evident given that human motion datasets with class or activity labels are still rare [5]. Moreover, existing heterogeneous trajectory prediction meth-

ods tailored for AD do not transfer well to robotics settings, as they depend on domain-specific contextual features [6]. 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 [7]. Additionally, both robotics and AD domains may feature non-uniform class distributions, leading to decreased performance of deep learning-based trajectory prediction methods [8]. It is important to understand whether class-conditioned prediction methods can benefit in applications with scarce or imbalanced data, and if so, to what extent and under which specific circumstances.

In this paper, we present an in-depth study of class-conditioned trajectory prediction methods under different conditions. We extend a pattern-based approach CLiFF-LHMP [9], which uses Maps of Dynamics [10] (MoD), to introduce a class-conditioned variant, and similarly adapt several deep learning methods to include class labels. In contrast to previous methods [11, 12, 13], 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), and various amounts of training data. The study of imbalanced datasets is significant 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. We analyse heterogeneous agents prediction in two distinct datasets: the Stanford Drone Dataset (SDD) [14] with diverse road users outdoors, and the THÖR-MAGNI dataset [5], with mobile robots and human agents in a mockup indoor industrial environment. 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. Fig. 1 outlines the main results of our study.

In summary, we make the following contributions:

- We establish a set of conditional Maps of Dynamics (MoD) and deep learning-based trajectory prediction baselines¹ for outdoor mixed traffic scenarios (SDD) and an indoor mobile robot dataset (THÖR-MAGNI).
- We analyse the performance of four deep learning methods and an MoD approach that consider activity labels or agent classes in THÖR-MAGNI and SDD.
- We show that class-conditioned methods outperform their unconditional counterparts in most cases. 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.

II. RELATED WORK

A. Motion Prediction for Heterogeneous Agents

The task of heterogeneous trajectory prediction involves estimating the future positions of an agent based on an observed trajectory, augmented by features describing the agent class, and optionally incorporating additional contextual factors, such as the obstacle maps. Deep learning has been widely applied to solve this problem [6, 15, 12], in particular Graph Neural Networks in the context of Autonomous Driving [11, 16, 17, 15, 13]. However, methods developed for predicting the motion of road agents do not transfer their assumptions when applied to other environments, such as intralogistic or public spaces. For instance, *TraPhic* [6] uses the shape of the road agent as a discriminative input feature for the various classes, which does not scale well to datasets with diverse human activities, such as THÖR [18] or THÖR-MAGNI [5]. Conversely, *HAICU* [19] uses the output of the perception module as a representation of the road agent's class, incorporating a continuous label distribution instead of a discrete value as the agent's type. Finally, [20] explores the use of dynamic Occupancy Grid Maps (OGMs) combined with semantic attributes to predict vehicle trajectories. However, it does not account for the heterogeneous entities typically present in road environments (e.g., pedestrians and cyclists). This limitation highlights the need for models that can be applied to different road users.

Alternative approaches involve individual deep learning modules for each agent class [11, 13, 12] to account for the heterogeneity of the dataset. These methods require individual encoders and/or decoders for each class, which presents scalability challenges as the number of classes increases. Conversely, in this paper we condition deep learning-based trajectory predictors on class embeddings, leading to a single model encompassing all classes. *Semantics-STGCNN* [17] addresses imbalanced class proportions in heterogeneous motion trajectory datasets through a class-balancing loss function, yet this strategy struggles to achieve top performance in all classes. To address issues of deep learning methods in imbalanced datasets, we propose a class-conditioned method based on Maps of Dynamics that is significantly less sensitive to class proportions than deep learning approaches and thus improves on its unconditional counterpart.

In this paper, we conduct an analysis of class-conditioned methods that are agnostic to the scene environment, allowing for applicability across different settings. Specifically, we evaluate four deep learning models and a Maps of Dynamics method [9], along with their respective conditional counterparts, on two datasets of human motion (THÖR-MAGNI) and road agent trajectories (SDD). We argue these methods are well-suited for application in new environments to support safe mobile robot navigation.

B. Heterogeneous Motion Trajectory Datasets

Motion trajectory datasets reflect a variety of factors that describe the dynamics of the agents' movements. These factors commonly relate to (1) agent-agent interactions [21, 22], providing insights into the social and interactive motion;

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

(2) agent-environment interactions [23], describing specific environment-related events in trajectory data or activities performed by the agent; (3) human-robot interaction (HRI) [24] supporting the development of social navigation methods. This paper focuses on trajectory prediction in heterogeneous motion datasets containing various classes of agents. These classes include labeled agents such as cars, pedestrians, and bicyclists[14], or diverse human activities that influence the motion dynamics in a working environment [5].

Heterogeneous human motion datasets have been gathered across diverse environments, including road scenes [25, 26], university campuses [14, 27, 28], surveilled outdoor areas [29, 30], and indoor settings [18, 5, 23]. In this paper, we tested our prediction methods on two datasets: the well-established outdoor SDD [14] and the novel indoor THÖR-MAGNI dataset [5]. We chose SDD and THÖR-MAGNI due to their distinct settings: imbalanced outdoor road agents and balanced human tasks in a robotics environment, respectively.

III. METHODS

A. Problem Statement

We frame the task of trajectory prediction as inferring a sequence of future states \mathcal{T} with the input of an observation sequence X , and the class of the agent c . A state $s \in X$ of an agent is represented by the 2D Cartesian coordinates (x, y) and the corresponding velocity vector (v_x, v_y) , i.e., $s = (x, y, v_x, v_y)$. Velocity can also be decomposed into 2D speed and orientation. The future sequence \mathcal{T} can be composed of velocities Y and positions P , depending on the method formulation. After observing O_p time steps, T_p future states are predicted, i.e. $O_p = |X|$ and $T_p = |\mathcal{T}|$.

B. Deep Learning Models

In this section, we present the deep learning methods to predict motion trajectories considering agent classes. Our analysis includes both single-output trajectory predictors (one prediction per observed trajectory), namely Long Short-Term Memory (LSTM), specifically the RED method [31], and Transformer-based, denoted by TF [2], as well as multiple-output approaches (multiple predictions per observed trajectory), including Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). We use the single-output methods, RED and TF, and their respective class-conditioned counterparts cRED and cTF, as outlined in [2]. The multiple-output approaches, GAN, VAE, and their respective conditioned counterparts cGAN and cVAE use Transformers-based encoders [32] (as described in this section).

1) *Single-Output Trajectory Predictors*: The first step is embedding X using a Multi-Layer Perceptron (MLP) network (see Fig. 2). Additionally, cRED and cTF 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

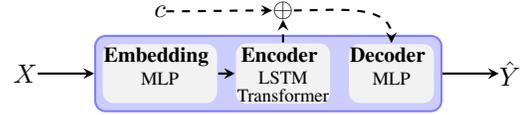


Fig. 2: Single-output unconditional and conditional methods (dashed lines).

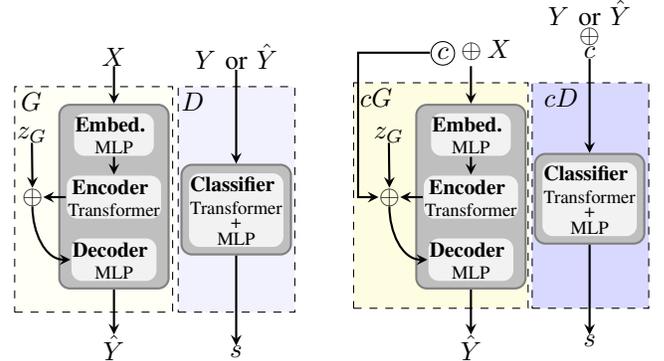


Fig. 3: GAN-based models: unconditional GAN (left) and cGAN (right).

features before decoding. We train single-output networks with the Mean Squared Error (MSE) loss:

$$L_T(P, \hat{P}) = \frac{1}{T_p} \sum_j \|p^j - \hat{p}^j\|_2, \quad (1)$$

where \hat{P} represents the estimated sequence of positions, p^j the ground truth position at time step j and \hat{p}^j the corresponding predicted position.

2) *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 X and a latent random vector \mathbf{z}_G to a realistic future set of velocities Y . We sample the latent vector from a standard normal Gaussian distribution. Simultaneously, the discriminator differentiates both real and generated future velocity vectors, Y and \hat{Y} , respectively. This adversarial training scenario is essential for producing multiple plausible trajectories. In cGAN, both 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:

$$L_G = \lambda_1 L_T + \lambda_2 \left(\frac{1}{2} \mathbb{E}[(D(Y) - 1)^2] + \frac{1}{2} \mathbb{E}[D(\hat{Y})^2] \right), \quad (2)$$

where λ_1 and λ_2 are the weights applied to the MSE term L_T (Eq. 1) 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.

Fig. 3 illustrates the network configurations for GAN and cGAN models. In general, the generators 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 [22], the discriminator comprises a transformer-encoder network and a MLP in the last layer. For cGAN, both the generator and the discriminator concatenate X to the agent's class embedding. The generator also concatenates the class embeddings to the input of the decoder.

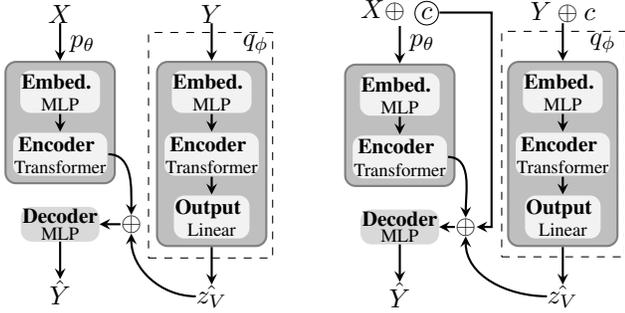


Fig. 4: VAE-based models: unconditional VAE (left) and cVAE (right). The recognition network (q_ϕ , enclosed with dashed border) is solely available during training.

3) *VAE-based Trajectory Predictors*: Generative 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 X and a latent vector z_V to the predicted trajectory Y , while the recognition network learns to map the ground truth trajectory Y 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:

$$L_V = \beta_1 L_T - \beta_2 D_{KL}[q_\phi(z_V|Y)||p_\theta(z_V|X)], \quad (3)$$

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 .

Fig. 4 shows the network configurations for the VAE and cVAE models. The predictor’s network configuration is identical to the generator in the GAN and cGAN models. The difference lies in the training process, where the latent vector 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.

C. Pattern-based Trajectory Predictors

Maps of Dynamics encode spatial or spatio-temporal motion patterns as a feature of the environment [33, 10]. By generalizing velocity observations, human dynamics can be represented through flow models. Prior work proposes CLiFF-LHMP [9], which exploits MoDs 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 50s ahead, [9], even with sparse, incomplete, and very limited training data [34].

In [9], a single CLiFF-map is used for all predicted trajectories, irrespective of the agent class. However, their motion patterns often differ, as shown in Fig. 5 and further detailed in Fig. 6. To address this, we introduce a class-

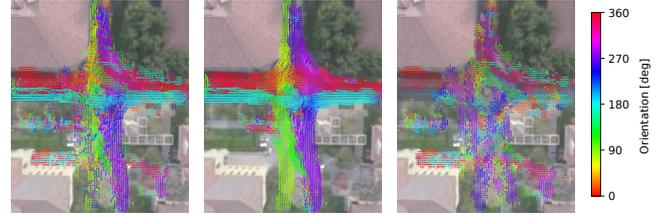


Fig. 5: General and class-conditioned CLiFF-maps in the *DeathCircle* scene of the SDD dataset. **Left**: all classes combined, **middle**: *Bicyclist* class, **right**: *Pedestrian* class. Colored arrows depict the mean speed (length) and direction (orientation) within the SWGMM of CLiFF-map, highlighting distinct motion patterns for different classes.

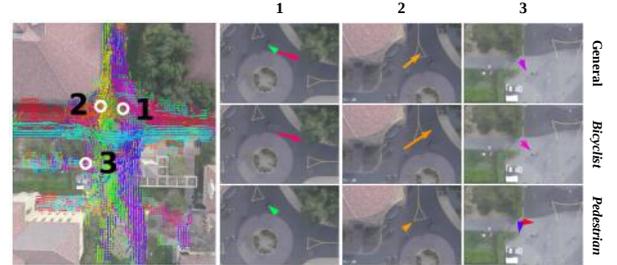


Fig. 6: CLiFF-maps at example locations in SDD [14]. Both general and class-conditioned CLiFF-maps of *Bicyclist* and *Pedestrian* of three locations are shown on the **right**. General CLiFF maps may depict combinations of multiple classes (point 1) or median speed and orientation (points 2 and 3).

conditioned CLiFF-map that differentiates the motion patterns representation to specific agent classes.

In a class-conditioned CLiFF-map, individual CLiFF-maps Ξ_c are built for each agent class using their specific trajectories. For agent class c , we estimate \mathcal{T} by sampling a velocity from Ξ_c within the sampling radius r_s for each prediction time step t . This velocity is then refined using a biased version of the Constant Velocity Model (CVM), following the same estimation process as the original CLiFF-LHMP, which is briefly described in the following. We refer the reader to [9] for more details. The velocity prediction at time step t is updated by biasing the last time step velocity with the sampled one as $\rho_t = \rho_{t-1} + (\rho_s - \rho_{t-1}) \cdot Kn(\rho_s - \rho_{t-1})$, $\theta_t = \theta_{t-1} + (\theta_s - \theta_{t-1}) \cdot Kn(\theta_s - \theta_{t-1})$, where ρ and θ represent speed and heading orientation of the agent, respectively. The kernel function Kn , defined as $Kn(x) = e^{-\beta\|x\|^2}$, modulates the influence of the sampled velocity. Using kernel Kn , the MoD term is scaled by the deviation between sampled and current velocities according to the CVM. The MoD is trusted less if it deviates more from the current velocity. Parameter β controls the reliance on the MoD versus the CVM, with a lower β favoring the velocity sampled from the MoD.

IV. EXPERIMENTS

A. Datasets

In this study, we evaluate and compare the performance of the trajectory prediction methods described in Sec. III on two datasets, THÖR-MAGNI [5] and SDD [14]. Importantly for our analysis, there is a substantial difference in class proportions between the two datasets. Specifically, SDD shows a noticeable class imbalance compared to THÖR-MAGNI. This inter-dataset class imbalance poses a significant challenge to accurate trajectory prediction. We analyze how this challenge

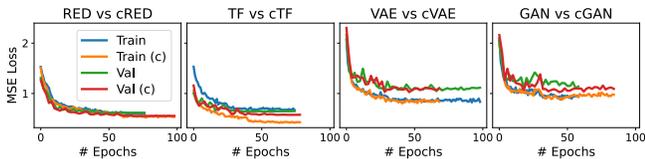


Fig. 7: MSE training curves examples for the deep learning models, where (c) denotes conditional variants and *Val* the validation curve.

is handled by the two categories of predictors: deep learning models and MoDs approaches.

THÖR-MAGNI includes 3.5 hours of human motion data in a laboratory setting with static and mobile robots. In some scenarios, people are assigned tasks such as moving objects (boxes, buckets, poster stands) which significantly influence their motion patterns, especially the velocity profiles [2]. This paper focuses on Scenarios 2, 3A, and 3B, which include 30 participants in 1.5 hours of 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%.

SDD encompasses 5 hours of heterogeneous trajectory data from 60 videos recorded on the Stanford University campus. It includes trajectories of bicyclists, pedestrians, skateboarders, carts, cars, and buses. Notably, certain classes such as *Bicyclist* and *Pedestrian* coexist in shared spaces 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 above 10 trajectories per class, resulting in 7 scenes with cumulatively 3 agent classes: *Pedestrian*, *Bicyclist* and *Car* with corresponding sample proportions of 64.6%, 34.3%, and 1.1%.

B. Implementation Details

To evaluate the predictors, we employed a repeated random sub-sampling validation method. For each iteration, we randomly selected $p\%$ of the dataset for training and used the remaining $(100 - p)\%$ 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. V-A), we set $p = 90$. In the data efficiency analysis (Sec. V-B), we decreased the percentage of data used for training from $p = 90$ to $p = 10$ in steps of 10. Following current trajectory prediction benchmarks [35], we set $O_p = 8$ and $T_p = 12$.

For deep learning-based predictors: We maintained a uniform hyperparameter setting to ensure a fair comparison. The training process for all networks extended to a maximum of 100 epochs with early stopping after 20 epochs with no improvement. We optimize the networks with the Adam optimizer [36], 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.

These hyperparameters and the networks’ configurations described in Sec. III-B allow training without overfitting, as shown by the loss curve examples in Fig. 7. Finally, 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: Identical parameters are used for both the class-conditioned and the general CLiFF-LHMP. The CLiFF-map grid resolutions for the SDD dataset and the THÖR-MAGNI dataset are 20 pixels and 0.2m, respectively. The sampling radius r_s 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.

C. Evaluation Metrics

To compare the trajectory predictors, we use the *Top-K Average and Final Displacement Errors* (Top-K ADE and FDE, in pixels for SDD and meters for THÖR-MAGNI), as in [16, 22]. Top-K ADE measures the average ℓ_2 distance between the ground truth track and the closest prediction (out of K samples), and FDE measures the distance between the last predicted position and the corresponding ground truth. We present the results of Top-1 and Top-3 ADE/FDE. When $K = 1$, we use the most likely output trajectory. We measure the mean and standard deviation of these metrics across iterations in the validation.

V. RESULTS

In our analysis, we aim to (1) quantify the improvement in trajectory prediction performance when using class attributes, and (2) evaluate trajectory prediction performance based on the specific characteristics of the dataset. The latter provides insight into the appropriate trajectory prediction method selection for a particular application. Due to space limitations, we only report the results for THÖR-MAGNI Scenario 2 in Table I, Fig. 9 and Fig. 12, and for Scenarios 3A and 3B in Fig. 10. We observe similar trends in all scenarios.

A. Accuracy Analysis Conditioned on Class Balance

Table I shows the prediction accuracy results separately for each class on Scenario 2 of the THÖR-MAGNI and SDD datasets. It also shows the global results for all trajectories (last rows 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). When predicting trajectories from the *Visitors–Group*, this difference is least pronounced. We speculate that this may be due to the fact that the motion patterns of these agents are less structured compared to the other classes, as shown in Fig. 9. This also highlights the

TABLE I: Top-1 ADE/FDE scores (ADE above, FDE below) in THÖR-MAGNI Scenario 2 and SDD datasets with a 90% train ratio. Bold values highlight superior performance of conditional models over their unconditional counterparts across most settings.

Data	Class	RED	cRED	TF	cTF	GAN	cGAN	VAE	cVAE	MoD	cMoD
THÖR-MAGNI Scenario 2	<i>Carrier-Box</i>	0.64±0.07 1.23±0.14	0.60±0.07 1.10±0.14	0.66±0.07 1.24±0.15	0.60±0.07 1.10±0.13	0.76±0.07 1.50±0.19	0.70±0.10 1.33±0.19	0.68±0.06 1.31±0.13	0.66±0.05 1.26±0.11	0.81±0.11 1.59±0.25	0.73±0.07 1.40±0.17
	<i>Carrier-Bucket</i>	0.71±0.06 1.35±0.18	0.67±0.06 1.21±0.15	0.65±0.05 1.24±0.13	0.60±0.06 1.12±0.16	0.78±0.04 1.48±0.18	0.73±0.06 1.37±0.14	0.73±0.09 1.44±0.20	0.74±0.08 1.43±0.19	0.92±0.18 1.78±0.37	0.72±0.10 1.30±0.17
	<i>Visitors-Alone</i>	0.81±0.05 1.53±0.12	0.78±0.06 1.48±0.13	0.79±0.04 1.52±0.12	0.75±0.04 1.45±0.14	0.88±0.07 1.72±0.14	0.85±0.08 1.67±0.19	0.84±0.06 1.62±0.16	0.83±0.05 1.61±0.14	0.94±0.06 1.97±0.20	0.92±0.09 1.95±0.22
	<i>Visitors-Group</i>	0.72±0.05 1.34±0.17	0.72±0.07 1.35±0.18	0.74±0.06 1.40±0.15	0.68±0.05 1.29±0.13	0.80±0.08 1.52±0.15	0.80±0.06 1.57±0.16	0.78±0.08 1.59±0.16	0.75±0.06 1.56±0.13	0.82±0.10 1.78±0.24	0.83±0.10 1.80±0.24
	<i>Carrier-LO</i>	0.73±0.05 1.44±0.12	0.69±0.03 1.38±0.10	0.69±0.05 1.41±0.12	0.64±0.04 1.31±0.08	0.78±0.08 1.59±0.16	0.75±0.06 1.56±0.13	0.77±0.07 1.50±0.15	0.72±0.06 1.44±0.12	0.83±0.10 1.73±0.22	0.75±0.08 1.61±0.19
	Global	0.74±0.01 1.41±0.04	0.71±0.03 1.34±0.06	0.72±0.02 1.39±0.04	0.67±0.02 1.29±0.05	0.81±0.05 1.59±0.09	0.78±0.05 1.53±0.09	0.77±0.03 1.49±0.05	0.76±0.02 1.47±0.06	0.87±0.05 1.79±0.10	0.80±0.05 1.67±0.09
	SDD	<i>Ped.</i>	18.63±0.54 37.55±1.22	18.76±0.54 37.69±1.08	18.99±0.89 37.60±1.76	19.00±0.79 37.72±1.30	20.26±0.69 40.27±1.25	20.21±0.49 40.31±0.92	20.92±1.25 41.49±2.32	20.09±0.77 39.90±1.44	19.88±0.46 40.02±1.07
<i>Car</i>		8.44±7.48 16.63±15.03	8.66±7.05 16.55±14.30	9.36±6.91 17.79±14.36	9.64±7.62 18.11±15.32	10.99±7.39 20.78±14.83	10.51±7.12 19.55±14.23	10.83±7.00 20.81±13.94	10.41±7.28 18.72±13.86	9.95±11.05 20.61±23.07	8.73±9.56 18.48±20.16
<i>Byc.</i>		64.08±2.49 137.42±5.31	64.38±2.53 137.56±4.97	65.33±2.37 142.08±4.60	64.01±2.67 139.97±4.75	67.32±2.75 145.50±5.04	67.04±2.41 144.50±4.61	68.22±3.58 147.00±6.32	67.80±3.36 145.13±6.64	64.35±2.02 142.51±4.40	63.60±2.02 141.01±4.24
Global		33.95±0.91 71.23±2.07	34.14±1.00 71.38±1.99	34.62±0.90 72.87±1.70	34.19±1.00 72.23±1.63	36.14±1.07 75.79±1.96	36.01±0.75 75.46±1.55	36.87±1.67 77.09±3.08	36.19±1.45 75.40±2.81	34.60±0.62 74.03±1.50	34.21±0.73 73.25±1.60

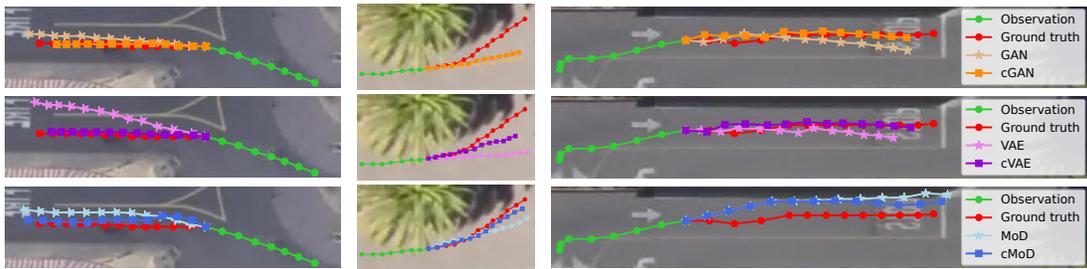


Fig. 8: Prediction examples of *Bycyclist* (left), *Pedestrian* (middle) and *Car* (right) in SDD with 4.8s prediction horizon.

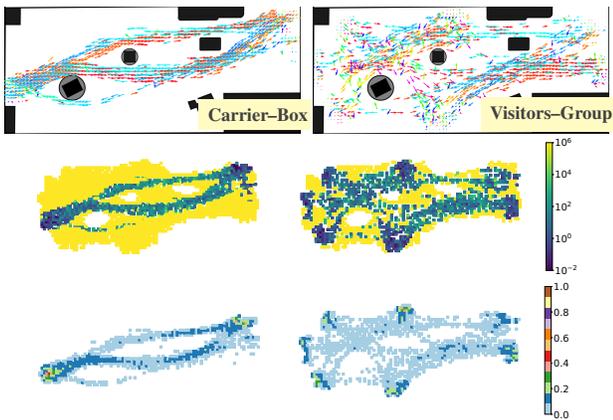


Fig. 9: Comparison of motion patterns of *Carrier-Box* and *Visitors-Group* in THÖR-MAGNI, Scenario 2. Class-conditioned CLiFF-maps (first row) show that *Carrier-Box* has a more distinct and structured motion pattern compared to *Visitors-Group*. The KL divergence heatmap (middle row) quantifies the 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.

importance of suitable class labels, such that each class encompasses specific motion patterns, and dataset-imposed 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 is able to 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. The THÖR-MAGNI dataset highlights differences in spatial patterns among classes, as shown in Fig. 9. Prediction accuracy improvements were more pronounced in classes with distinct motion patterns, such as *Carrier-Box* and *Carrier-Bucket*, which deviate more from the general motion pattern. In SDD, variances in speed are observed among different classes, as depicted in Fig. 5. A single CLiFF-map struggles to accurately model variations across multiple classes, leading to inaccurate predictions compared to the class-conditioned MoD-aware method.

B. Data Efficiency Analysis

To assess how training data volume affects model performance, we conducted a data efficiency analysis aimed at identifying optimal models for various data settings. Fig. 10 shows the performance of single-output methods (RED, TF, and MoD, along with their conditioned variants) in THÖR-MAGNI Scenarios 3A and 3B. cMoD outperforms deep learning methods in Top-1 ADE in low data regimes, where 10% of data is available during training. Moreover, performance for deep

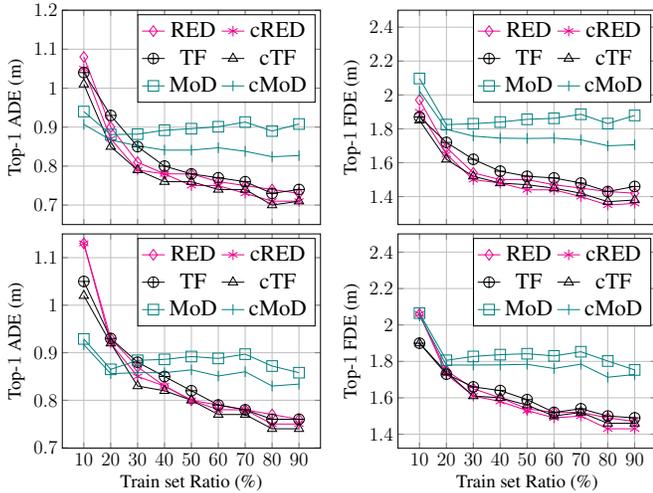


Fig. 10: Top-1 ADE/FDE scores in THÖR-MAGNI Scenario 3A (top row) and 3B (bottom row). In this class-balanced setting, deep learning methods surpass MoD approaches. However, MoD methods (MoD and cMoD) maintain stability even with reduced training data.

learning methods declines with less training data, whereas MoD approaches (MoD and cMoD) are more stable across different data regimes. The MoD model we employ, CLiFF-map, efficiently captures major human motion patterns with limited training data. Beyond a 30% training data increase, improvements in CLiFF-map 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 significant performance improvements. This stability highlights the MoD approach’s advantage in scenarios where extensive data collection is impractical. Fig. 11 presents the performance of multiple-output methods (VAE, GAN, and MoD, along with their respective conditioned variants) on both datasets. In THÖR-MAGNI, deep generative methods prove more effective in generating one out of K trajectories compared to MoD-based methods. Conversely, 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.

C. Qualitative Results

We provide qualitative Top-1 trajectory prediction comparisons for each multiple-output approach in Fig. 8 and for each single-output method in Fig. 12 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.

D. Limitations

In this work we analyze the effect of the dataset-imposed classes on motion prediction accuracy and compare deep

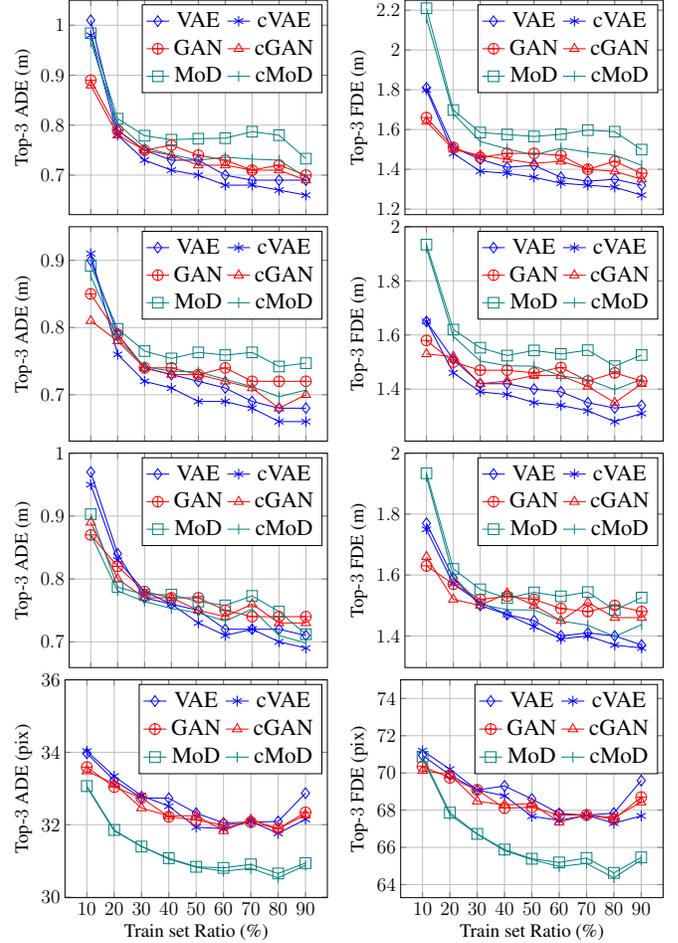


Fig. 11: Top-3 ADE/FDE scores across THÖR-MAGNI Scenarios 2, 3A, 3B (top to third rows), and SDD (bottom row). In the class-balanced THÖR-MAGNI, deep generative methods excel over MoD. In the imbalanced SDD, MoD methods outperform deep generative methods across all data regimes.

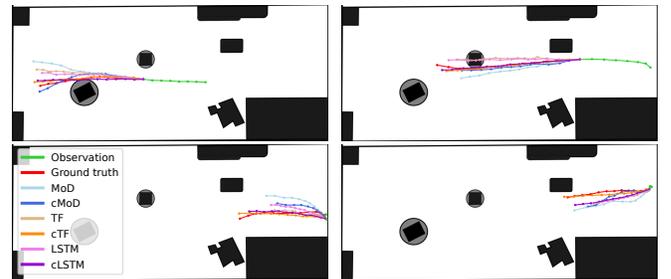


Fig. 12: Prediction examples of Carrier-Box (top left), Carrier-Bucket (top right), Visitors-Alone (bottom left) and Carrier-Large Object (bottom right) in THÖR-MAGNI with 4.8s prediction horizon.

learning with pattern-based methods across various data settings. However, our methods do not explicitly consider agent interactions, due to the own complexity of evaluating and comparing the interaction models [37]. We aim to address this challenge in the future work.

VI. CONCLUSIONS & FUTURE WORK

The challenge of making accurate trajectory predictions in dynamic environments is further complicated when facing heterogeneous agents with diverse dynamics and distinct motion patterns. Considering the classes of agents can help

lowering uncertainty in motion forecasts, an issue that arises when attempting to generalize across different classes. In this paper, we analyze how prior art in deep learning-based and pattern-based prediction can be adapted to consider class labels, concluding that class-conditioned methods generally outperform their unconditioned counterparts. The choice of a specific method, on the other hand, strongly depends on the available training data and the intended downstream application: in new environments with limited data, or where some classes are underrepresented and require multimodal predictions (sometimes critically so, e.g., vulnerable road users in automated driving), pattern-based methods may have an edge over the deep learning models. In future work, we plan to explore unsupervised trajectory and dynamics clustering to create more natural and informative class definitions. This approach aims to address the limitations of dataset-imposed classes (i.e., unstructured motion patterns within a class) and improve model performance in handling class imbalances.

REFERENCES

- [1] P. Teja Singamaneni, A. Favier, and R. Alami. “Human-Aware Navigation Planner for Diverse Human-Robot Interaction Contexts”. In: *Proc. of the Int. Conf. on Intell. Robots and Syst. (IROS)*. 2021.
- [2] T. R. de Almeida et al. “THÖR-Magni: Comparative Analysis of Deep Learning Models for Role-Conditioned Human Motion Prediction”. In: *Proc. of the Int. Conf. on Comp. Vision Worksh.* 2023.
- [3] J. Fang, C. H. Zhu, P. Zhang, H. Yu, and J. Xue. “Heterogeneous Trajectory Forecasting via Risk and Scene Graph Learning”. In: *Trans. on Intell. Transp. Syst. (TITS)* (2022).
- [4] L. Heuer, L. Palmieri, A. Rudenko, A. Mannucci, M. Magnusson, and K. O. Arras. “Proactive Model Predictive Control with Multi-Modal Human Motion Prediction in Cluttered Dynamic Environments”. In: *Proc. of the Int. Conf. on Intell. Robots and Syst. (IROS)*. 2023.
- [5] T. Schreiter et al. “The Magni Human Motion Dataset: Accurate, Complex, Multi-Modal, Natural, Semantically-rich and Contextualized”. In: *Proc. of the IEEE RO-MAN Worksh. (SIRRW)*. 2022.
- [6] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha. “TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions”. In: *Proc. of the Conf. on Comp. Vis. and Pat. Rec. (CVPR)*. 2019.
- [7] Y. Cui, H. Zhang, Y. Wang, and R. Xiong. “Learning World Transition Model for Socially Aware Robot Navigation”. In: *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. 2021.
- [8] T. Rodrigues de Almeida, E. Gutierrez Maestro, and O. Martinez Mozos. “Context-free Self-Conditioned GAN for Trajectory Forecasting”. In: *Proc. of the Int. Conf. on Mach. Learning and App. (ICMLA)*. 2022.
- [9] Y. Zhu et al. “CLiFF-LHMP: Using Spatial Dynamics Patterns for Long-Term Human Motion Prediction”. In: *Proc. of the Int. Conf. on Intell. Robots and Syst. (IROS)*. 2023.
- [10] T. P. Kucner et al. “Survey of maps of dynamics for mobile robots”. In: *Int. J. of Robotics Research* (2023).
- [11] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. “TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents”. In: *Proc. of the AAAI Conf. on Artificial Intelligence (AAAI)*. 2019.
- [12] M. Geng, J. Li, C. Li, N. Xie, X. Chen, and D.-H. Lee. “Adaptive and Simultaneous Trajectory Prediction for Heterogeneous Agents via Transferable Hierarchical Transformer Network”. In: *Trans. on Intell. Transp. Syst. (TITS)* (2023).
- [13] X. Mo, Z. Huang, Y. Xing, and C. Lv. “Multi-Agent Trajectory Prediction With Heterogeneous Edge-Enhanced Graph Attention Network”. In: *Trans. on Intell. Transp. Syst. (TITS)* (2022).
- [14] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. “Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes”. In: *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*. 2016.
- [15] F. Zheng et al. “Unlimited Neighborhood Interaction for Heterogeneous Trajectory Prediction”. In: *Proc. of the Int. Conf. on Computer Vision (ICCV)*. 2021.
- [16] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone. “Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data”. In: *Proc. of the Europ. Conf. on Comp. Vision (ECCV)*. 2020.
- [17] B. Rainbow, Q. Men, and H. P. H. Shum. “Semantics-STGCNN: A Semantics-guided Spatial-Temporal Graph Convolutional Network for Multi-class Trajectory Prediction”. In: *Proc. of the Conf. on Systems, Man, and Cybernetics (SMC)*. 2021.
- [18] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal. “THÖR: Human-Robot Navigation Data Collection and Accurate Motion Trajectories Dataset”. In: *Robotics and Automation Letters* (2020).
- [19] B. Ivanovic et al. “Heterogeneous-Agent Trajectory Forecasting Incorporating Class Uncertainty”. In: *Proc. of the Int. Conf. on Intell. Robots and Syst. (IROS)*. 2022.
- [20] R. Asghar, M. Diaz-Zapata, L. Rummelhard, A. Spalanzani, and C. Laugier. “Vehicle Motion Forecasting Using Prior Information and Semantic-Assisted Occupancy Grid Maps”. In: *Proc. of the Int. Conf. on Intell. Robots and Syst. (IROS)*. 2023.
- [21] A. Salatiello, M. Hovaidi-Ardestani, and M. A. Giese. “A Dynamical Generative Model of Social Interactions”. In: *Frontiers in Neuro-robotics* (2021).
- [22] P. Kothari and A. Alahi. “Safety-Compliant Generative Adversarial Networks for Human Trajectory Forecasting”. In: *Trans. on Intell. Transp. Syst. (TITS)* (2023).
- [23] P. Kratzer, S. Bihlmaier, N. B. Midlagajni, R. Prakash, M. Toussaint, and J. Mainprice. “MoGaze: A Dataset of Full-Body Motions that Includes Workspace Geometry and Eye-Gaze”. In: *Robotics and Automation Letters* (2021).
- [24] H. Karnan et al. “Socially Compliant Navigation Dataset (SCAND): A Large-Scale Dataset of Demonstrations for Social Navigation”. In: *Robotics and Automation Letters* (2022).
- [25] M.-F. Chang et al. “Argoverse: 3D Tracking and Forecasting with Rich Maps”. In: *Proc. of the Conf. on Comp. Vis. and Pat. Rec. (CVPR)*. 2019.
- [26] B. Wilson et al. “Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting”. In: *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*. 2021.
- [27] Z. Yan, T. Duckett, and N. Bellotto. “Online learning for human classification in 3D LiDAR-based tracking”. In: *Proc. of the Int. Conf. on Intell. Robots and Syst. (IROS)*. 2017.
- [28] M. Ehsanpour, F. Saleh, S. Savarese, I. Reid, and H. Rezatofighi. “JRDB-Act: A Large-scale Dataset for Spatio-temporal Action, Social Group and Activity Detection”. In: *Proc. of the Conf. on Comp. Vis. and Pat. Rec. (CVPR)*. 2022.
- [29] S. Oh et al. “A large-scale benchmark dataset for event recognition in surveillance video”. In: *Proc. of the Conf. on Comp. Vis. and Pat. Rec. (CVPR)*. 2011.
- [30] T. Shu, D. Xie, B. Rothrock, S. Todorovic, and S.-C. Zhu. “Joint Inference of Groups, Events and Human Roles in Aerial Videos”. In: *Proc. of the Conf. on Comp. Vis. and Pat. Rec. (CVPR)*. 2015.
- [31] S. Becker, R. Hug, W. Hubner, and M. Arens. “RED: A simple but effective Baseline Predictor for the TrajNet Benchmark”. In: *Proc. of the Europ. Conf. on Comp. Vision Worksh.* 2018.
- [32] A. Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Inf. Proc. Syst. (NIPS)*. 2017.
- [33] T. P. Kucner, M. Magnusson, E. Schaffernicht, V. H. Bennets, and A. J. Lilienthal. “Enabling Flow Awareness for Mobile Robots in Partially Observable Environments”. In: *Robotics and Automation Letters* (2017).
- [34] Y. Zhu, A. Rudenko, T. P. Kucner, A. J. Lilienthal, and M. Magnusson. “A Data-Efficient Approach for Long-Term Human Motion Prediction Using Maps of Dynamics”. In: *ICRA Worksh. (LHMP)*. 2023.
- [35] P. Kothari, S. Kreiss, and A. Alahi. “Human Trajectory Forecasting in Crowds: A Deep Learning Perspective”. In: *Trans. on Intell. Transp. Syst. (TITS)* (2021).
- [36] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *Int. Conf. on Learning Representations (ICLR)*. Ed. by Y. Bengio and Y. LeCun. 2015.
- [37] A. Rudenko, L. Palmieri, W. Huang, A. J. Lilienthal, and K. O. Arras. “The Atlas Benchmark: an Automated Evaluation Framework for Human Motion Prediction”. In: *Proc. of the Int. Symp. on Robot and Human Interactive Comm. (RO-MAN)*. 2022.