Multi-class Trajectory Prediction in Urban Traffic using the View-of-Delft Dataset

Bruno K.W. Martens

Master of Science Thesis

TU Delft Delft University of Technology

Multi-class Trajectory Prediction in Urban Traffic using the View-of-Delft Dataset

Master Thesis

For the degree of Master of Science in Robotics at Delft University of Technology

Bruno K.W. Martens

4389956

Thesis committee:	Prof. Dr. D.M. Gavrila	TU Delft, supervisor, chair
	Ir. H.J. Boekema	TU Delft, supervisor, member
	Dr. J.F.P. Kooij	TU Delft, external member

May 28, 2023

To be defended publicly on 01-06-2023.

Multi-class Trajectory Prediction in Urban Traffic using the View-of-Delft Dataset

Bruno K.W. Martens[1]

***Abstract*—Critical to the safe application of autonomous vehicles is the ability to accurately predict the future motion of agents surrounding the vehicle. This is especially important - and challenging - in urban traffic, where vehicles share the road with Vulnerable Road Users (VRUs) such as pedestrians and cyclists. However, the majority of the existing on-board prediction datasets focus on predicting future trajectories of vehicles. We therefore present the View-of-Delft Prediction dataset, an extension of the recently-released urban View-of-Delft (VoD) dataset. The proposed dataset contains a large proportion of VRUs and has a good class balance, consisting of 844 prediction scenarios in the city of Delft, with 228 prediction instances for vehicles, 159 for cyclists, and 444 for pedestrians in dense urban traffic. Since state-of-the-art trajectory prediction approaches are primarily developed on car-dominated traffic with little interaction with VRUs, we analyse if the same methodology is suitable for mixed-traffic urban environments with VRUs and vehicles in close proximity. As our baseline for this analysis, we select the graph-based PGP model, for which we propose the addition of encoding motion of surrounding cyclists separately to facilitate its application in dense urban traffic. Since PGP relies on the lane graph topology, we provide novel rich map annotations for the VoD dataset, including lane polylines. Our analysis shows that there is a significant domain gap between the vehicle-dominated nuScenes and VRU-dominated View-of-Delft Prediction datasets, as training only on nuScenes results in a 107.79% higher minADE_{10} on the VoD Prediction test set than training the model on VoD Prediction. Furthermore, we modify the model by adding target agent class information, to make it suited for multi-class trajectory prediction. Our analysis shows that this yields a significant performance improvement of 13.92% in minADE_{10} for a six-second prediction horizon. The View-of-Delft Prediction dataset will be publicly released, enabling novel research on urban mixed-traffic trajectory prediction.**

***Index Terms*—Vectorized Representations, Trajectory Prediction, Graph Neural Networks, Autonomous Driving**

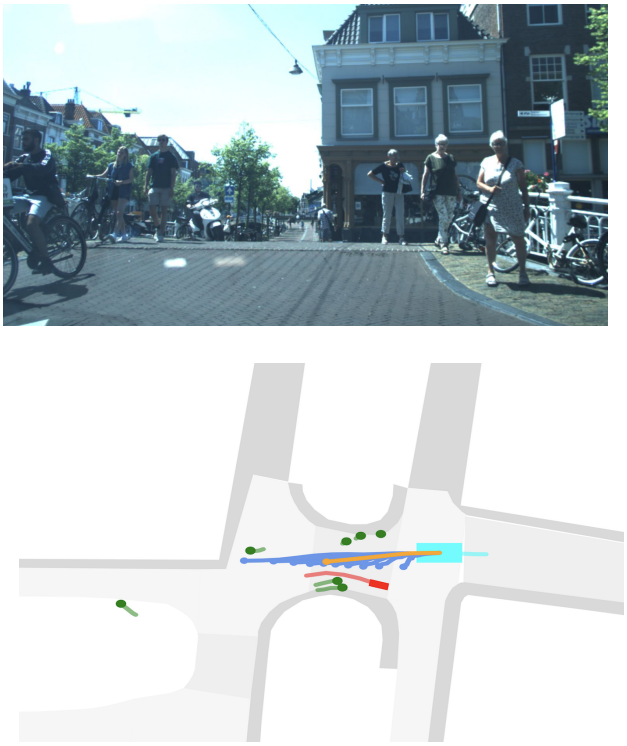


Fig. 1: Camera view and prediction scenario in the VoD Prediction dataset. The vectorised map elements are visualised in the bottom image. The target agent is shown in cyan, pedestrians are shown in green, and cyclists in red. The ground truth is shown in orange and predictions in blue.

I. Introduction

Forecasting trajectories of nearby traffic agents is key to the safe application of autonomous vehicles. In particular, reducing the number of accidents involving Vulnerable Road Users (VRUs) such as pedestrians and cyclists could lead to fewer road deaths, as these agents make up more than half of all traffic fatalities [1].

However, forecasting the motion of VRUs in mixed traffic is especially challenging due to their stochastic and multi-modal behaviour and the unique dynamics and traffic laws for each class of agent. This is of specific concern in urban settings, where traffic is dense, there are many VRUs, and traffic infrastructure is shared between agent classes, leading to complex interactions between agents.

Trajectory prediction performance for vehicles has improved significantly in recent years [2]–[5] as approaches increasingly rely on modelling the combination of semantic cues present in traffic scenarios. These cues range from map information to agent dynamics to interactions between agents. [6]. State-of-the-art methods for this task are deep-learning-based approaches that use an encoder-decoder architecture to encode the variety of contextual cues present in a scenario. Generally, the encoder transforms the past trajectory of agents, their interactions, and the local map information into a feature representation of the scene. The decoder predicts possible future trajectories for the target agent from such representations.

Current state-of-the-art models use 'vectorised' inputs to encode salient information for the prediction task [7]–[9]. This type of input efficiently represents road elements from

[1]Intelligent Vehicles Group, TU Delft, The Netherlands.

vectorised map data and motion states of tracked agents as vectors, which avoids issues associated with rasterised representations of the environment, e.g. lossy rendering and computational inefficiency [7], [9]. Vectorised semantic map information has been used with great success to learn the influence of the static environment on a vehicle's trajectory. This approach not only improves scene compliance but also the accuracy of predictions for vehicles [8]–[10].

Despite these encouraging developments, predicting the trajectories of VRUs remains challenging and most existing datasets primarily focus on vehicle prediction [11]. Besides that, these datasets are largely recorded in suburban or regional locations. While dense inner-city traffic is rich in interactions and tends to have a higher number of VRUs, resulting in more challenging and safety-critical scenarios than many current datasets offer, and thus poses additional challenges such as partial occlusion and complex interactions between agents.

We therefore present the View-of-Delft Prediction dataset, an urban prediction dataset with a good class balance and a high proportion of VRUs such as pedestrians and cyclists in addition to vehicles, to enable research on multi-class trajectory prediction in this challenging setting. This dataset is an extension of the View-of-Delft (VoD) dataset [12], and adds semantic map data which state-of-the-art trajectory prediction methods rely on as a prior for accurate trajectory prediction. An example camera image from the VoD dataset and the corresponding prediction scenario are shown in Figure 1. To indicate the relevance of our proposed dataset, we evaluate whether a domain shift is present between the widely-used nuScenes [13] dataset and VoD prediction, which was recorded in dense urban traffic. For this study, we use Prediction via Graph-based Policy (PGP) [8], a graph-based trajectory prediction approach. We add information of surrounding cyclists as a distinct class to the model, to make it suited for VRU-dense environments. Further, we investigate if the current trajectory prediction methodology for vehicles is also well suited for other agent classes in these VRU-dominated environments. To this extent, we apply PGP on our novel View-of-Delft Prediction dataset. Finally, we propose the addition of target agent class information to the PGP architecture, leading to increased prediction performance.

II. Related Work

A. Trajectory Prediction

A considerable amount of literature exists on trajectory prediction of traffic agents in the context of autonomous driving; see [2]–[5] for surveys on this topic. Leveraged methodologies range from physics-based to planning-based to pattern-based and increasingly rely on contextual cues such as agent information, social interaction and static environmental cues [3]. Physics-based approaches rely on dynamical models and are most effective for short-term prediction horizons as they struggle with capturing the increased complexity of longer prediction horizons or complex interactions. Planning-based approaches tend to rely on the goal of the agent, which is often latent at prediction time in the context of autonomous driving. Nowadays, the majority of work leverages pattern-based methods, aided by the rapid developments in deep learning. Pattern-based methods can more easily account for a wide variety of environmental cues and have an increased ability to handle the latent intent of agents, making them suited for prediction in complex environments with longer prediction horizons [3]. However, the majority of current work focuses on predicting the future trajectory for a single class of road user, e.g. cars [8], [14], [15], cyclists [16], or pedestrians [17]–[19]. Few approaches are designed to forecast the future trajectories of multiple classes of agents [14], [20], [21].

Advancements in deep learning led to the popularity of pattern-based encoder-decoder architectures to encode and account for a variety of environmental cues. Initially, rasterized representations [22]–[25] were a popular choice for encoding contextual cues. In recent years, vectorised representations [7], [8], [26], [27] have gained popularity as a means to encode scene information, as they do not suffer from the lossy rendering, manual tuning and high computational requirements inherent to rasterised representations [7]. Prediction via Graph-based Policy (PGP) [8] effectively leverages a vectorized representation to sample feasible trajectories for vehicles over the lane graph. This is an advantage over popular goal-conditioned prediction methods [14], [15], [28], [29], which only take the feasibility of the selected goal location into account, not of the possible routes to a goal [10]. Additionally, PGP uses Graph Neural Networks (GNNs), which are a natural choice for modelling the interactions between traffic agents and the road topology with this representation and have been shown to improve prediction performance [8], [26], [30].

B. Motion Prediction Datasets

Followed by the recent interest in autonomous driving and related tasks, numerous trajectory prediction datasets have been released in the past few years [11], [13], [31]–[33], driving progress in trajectory prediction. We limit our scope to datasets that were recorded from a moving vehicle - the so-called 'on-board' setting - as this setting is the most relevant to autonomous driving systems. For a dataset to be useful for trajectory prediction in mixed traffic, we argue it should meet the following requirements: 1) contain prediction instances of multiple agent classes, 2) have a high density of interactions between different agent classes, 3) contain accurate vectorised map annotations, and 4) provide sensor data to enable the use of subtle context cues. In Table I we show the compliance of commonly-used datasets with these requirements.

Many of the current datasets that are suitable for multi-class trajectory prediction either have a class imbalance, with vehicles making up the majority (prediction) class or are comprised of highly structured traffic scenarios in suburban or regional locations and thus have limited interactions between agent classes. These datasets are primarily recorded in North America, which is rich in multi-lane traffic and where traffic possesses a limited amount of cases where road elements are shared between agents, which leads to fewer interactions between different agent classes. Lyft level 5 [33] specifically is a large dataset. However, it is recorded in a small geographical domain (less than six kilometres in road length), with recording vehicles driving the same region multiple times.

TABLE I: Overview of motion prediction datasets, with their released sensor information, map information, size of the dataset, number of agents, number of classes for prediction, and recording locations. † We only consider high-quality map annotations, i.e. human-annotated and vectorised. * All agents reported because a breakdown of target agents is not officially reported. N. Am. = North America.

Dataset	Location	Information				Size		Agents for Prediction			
		Camera	LiDAR	Radar	Sem. Map†	Scenes	Duration (s) (hist. - fut.)	Vehicles (# (%))	Cyclists (# (%))	Pedestrians (# (%))	# Pred. Classes
Lyft Level 5 [33]	N. Am.	☐	☐	☐	☐	170k	0.5 - 5	49M* (92)	77k* (6)	605k* (2)	3
WOMD [31]	N. Am.	☐	☐	☐	☑	104k	1 - 8	60k* (72)	620* (0)	23k* (28)	3
Argoverse 2 [32]	N. Am.	☐	☐	☐	☑	250k	5 - 6	>10k (>84)	>1000 (>8)	>1000 (>8)	5
NuScenes [13]	N. Am., Asia	☑	☑	☐	☑	1000	2 - 6	16k (100)	0 (0)	0 (0)	1
Euro-PVI [11]	Europe	☑	☑	☐	☐	1077	0.5 - 3	1077 (12)	1581 (18)	6177 (70)	3
View-of-Delft	Europe	☑	☑	☑	☑	844	1 - 2	228 (28)	159 (19)	444 (53)	3

Accurate and detailed map annotations have become crucial due to the increasing reliance of prediction models on map information [32]. Although many recent datasets provide some form of map annotation, not all of these annotations are of equal quality. Lyft Level 5 [33] provides the location of lane boundaries in a non-vector format. Euro-PVI [11] contains only segmented semantic maps, adopted from OpenStreetMap [1]. We argue that the accuracy of these segmented maps is too limited for autonomous driving tasks, as supported by evidence in Appendix A and [34].

Sharing additional information, besides lane information, such as regions where pedestrians and cyclists can manoeuvre, allows for the development of more universal prediction methods. Only nuScenes [13] provides this information, however, they just provide prediction instances of vehicles in their test set, making the dataset unsuitable for multi-class trajectory prediction.

Lastly, sensor data can be used to develop more effective prediction frameworks, as it provides additional information not available from agent tracks. However, most large-scale trajectory datasets do not provide sensor information such as camera images, LiDAR pointclouds, or radar data.

The only dataset that has a significant percentage of VRUs and consists of dense urban traffic is Euro-PVI [11], which was recorded in Europe. However, we note that Euro-PVI has limitations in the sense that both their map annotations and agent tracks lack sufficient accuracy, as shown in Appendix A. Furthermore, Euro-PVI [11] only contains interactions between the recording vehicle and VRUs, as surrounding vehicle tracks are not provided, which limits the ability to study the interaction between vehicles.

C. Contributions

Our contributions are threefold:

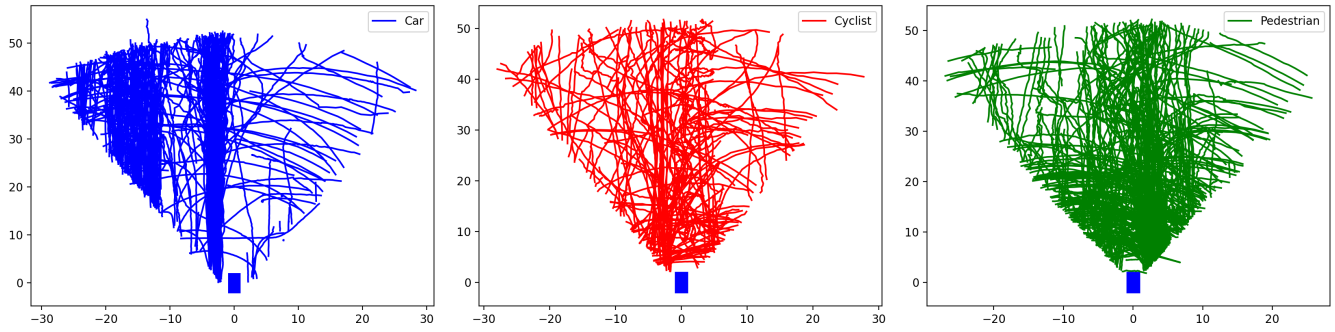
1) We release the naturalistic View-of-Delft Prediction dataset, an extension of the urban View-of-Delft dataset [12]. This dataset has a good class balance and contains a large proportion of VRUs such as pedestrians and cyclists and dense interactions between agent classes. It additionally has high-quality 3D road user annotations, labelled vectorised semantic map elements such as lanes, crosswalks, intersections and off-road areas, plus sensor data from camera, LiDAR, and radar. The dataset is accompanied by a software kit to enable motion prediction for pedestrians, cyclists, and vehicles.
2) To study the relevance of this dataset, we study the domain shift between the vehicle-dominated nuScenes and urban View-of-Delft Prediction datasets. We select the 'vector-based' PGP [8] model for our analysis. We find that there is a significant domain gap between models trained on the nuScenes and View-of-Delft Prediction datasets, as training the model on nuScenes results in a 107.79% higher minADE_{10} on the VoD Prediction test set compared to training the model on VoD Prediction.
3) Finally, we investigate whether vehicle-based trajectory prediction approaches are suitable for mixed-traffic urban settings. We modify the PGP model, by encoding the motion of surrounding cyclists separately, to make it applicable to prediction in urban settings. Additionally, we propose a class-aware version of the model: PGP-CA, by providing agent class information to the trajectory decoder such that it accounts for class-specific dynamics. PGP-CA outperforms PGP for every single agent class, leading to an overall improvement of 13.92% in minADE_{10} for a six-second prediction horizon on our dataset.

III. Dataset

In this section, we present the View-of-Delft Prediction dataset, an extension of the View-of-Delft (VoD) dataset [12] for trajectory prediction in urban environments. The VoD Prediction dataset contains dense interactions between all agent classes. Figure 2 shows the observed agent tracks per agent class, which shows the proximity of surrounding agents in our dataset. Contrary to Euro-PVI [11], the VoD Prediction dataset also provides track information of surrounding vehicles.

The VoD dataset comprises camera, radar, LiDAR, and GPS/IMU information. A summary of the dataset and comparison with major trajectory prediction datasets can be found in Table I. For more details on the sensor setup, we refer readers to the VoD paper [12]. Here, we outline the specific additions that allow the dataset to be used for trajectory prediction for vehicles and VRUs.

[1]https://www.openstreetmap.org/



(a) Vehicles (b) Cyclists (c) Pedestrians

Fig. 2: Tracks of surrounding agents observed in an on-board setting in the VoD dataset. The blue rectangle depicts the recording vehicle.

A. Vector Map Information

As current trajectory prediction methods rely on semantic map data, we provide accurate annotations of lanes, intersections, crosswalks and off-road areas (pedestrian domain) with extensive labels for each road element. This means that our dataset contains information that can aid the prediction performance for the most common agent classes: vehicles, pedestrians, and cyclists. The annotations are created by human annotators from georeferenced aerial images, as further outlined in Appendix B. We label every road element with a unique element identifier, its road type and which road users are allowed to use the road element. The road element-specific labels are summarized in Table II.

Lanes are annotated as polylines demarking the left and right boundary of the driving lane, from which the lane centreline is interpolated. Additionally, we provide the following labels: a lane identifier to match the left and right boundary of the lane, the direction of the lane (one-way or bidirectional), whether the lane has any connections in the form of predecessor or successor lanes and the type of road boundary (i.e. solid/dashed marking). This helps to determine feasible lane switch possibilities.

Intersections are annotated as a polygon that encloses the area of an intersection. As lane boundaries are often not clearly visible at intersections, we use the predecessor and successor labels of adjacent lanes to determine viable connections and interpolate a natural connection line between the two-lane centrelines within the intersection polygon.

Crosswalks are polygons that indicate appointed pedestrian crossing locations.

Off-road areas indicate the pedestrian domain, e.g. sidewalks or city squares. These polygons are included to aid in the prediction of pedestrian future motion.

To the best of our knowledge, we are the first to provide a label for which type of agents are allowed to traverse each instance of road elements. This is relevant in urban areas, where e.g. cyclists might be allowed on some roads but not on others, and may aid further research into urban multi-class trajectory prediction.

TABLE II: Summary of labels of vectorized map information released with the VoD Prediction dataset.

	Lane	Crosswalk	Off-Road Area	Intersection
element id	☑	☑	☑	☑
road type	☑	☑	☑	☑
allowed agents	☑	☑	☑	☑
boundary right	☑	☐	☐	☐
boundary left	☑	☐	☐	☐
boundary type	☑	☐	☐	☐
predecessors	☑	☐	☐	☐
successors	☑	☐	☐	☐
bidirectional lane	☑	☐	☐	☐
lane id	☑	☐	☐	☐

B. Scene selection

The View-of-Delft dataset is recorded with a preference for locations both rich in the number of VRUs and interactions with VRUs. We select all recorded frames from the original VoD dataset to be applicable to the VoD prediction dataset. However, we discard the following scenes:

1) Highway & suburban traffic: We discard recorded frames in high-speed traffic, such as on the high-way and regional roads, as they are sparse in the amount of (interactions with) VRUs. These sequences are thus not within our research scope.

2) Frames affected by noisy ego-localisation: We filter out frames that suffer from localisation inaccuracies due to noisy ego-localisation at the start of recording, by discarding frames where the traversed distance requires the recording vehicle to move with a speed higher than $36.1\ \mathrm{ms}^{-1}$, the maximum legal speed in the Netherlands.

For each scene, we select target agents based on their presence in every recorded frame, besides the recording vehicle itself. This means that we have trajectory information over the complete motion history and prediction horizon for each prediction instance, contrary to e.g. nuScenes [13]. We do take agents that are partially observed into account as surrounding agents. Furthermore, we filter target agents based on the following:

3) Parked Cars: As many parked cars along the canals in the inner city of Delft are naturally present in the dataset, we filter out parked cars as target agents by dismissing cars that have an average speed lower than $0.28\ \mathrm{ms}^{-1}$.

4) Infeasible vehicle trajectories: Additionally, we discard vehicles that move more than $36.1\ \mathrm{ms}^{-1}$ during their motion history or ground truth trajectory or have infeasible deviations in yaw ($\geq \pi/2$) between consecutive frames.

By releasing a modular software kit, as further explained in Appendix C, we allow practitioners to tailor the prediction horizon to their research scope. In this work, we chose to experiment with a two-second and six-second prediction horizon. We use a six-second horizon to be able to compare the prediction performance of PGP between the nuScenes dataset and VoD Prediction dataset. The two-second prediction horizon was selected to evaluate whether trajectory prediction models can cope with complex urban settings. We adhere to the original VoD [12] dataset splits, generating scenes using these splits results in a 53%/12%/34% train/val/test split for a two-second prediction horizon. If a scene has multiple prediction instances, we assign them to the same split.

IV. Methodology

We analyse the performance of trajectory prediction approaches using vectorised representations that are designed using a vehicle-dominated dataset on the VRU-dominated View-of-Delft Prediction dataset. Therefore, we select PGP [8] as the baseline for this analysis, as this is a prediction method that uses vectorised representations and performs well on the vehicle-heavy nuScenes dataset [13]. Since this method was originally developed for the prediction of vehicles only, we modify the model to make it suitable for VRU prediction, and investigate its ability to predict the trajectories of vehicles and VRUs in mixed traffic situations. We discuss here first the regular PGP architecture, how we adjust that to make it suited for trajectory prediction in VRU-dominated dense urban traffic, and then present our PGP-CA model that improves the ability of the model to handle multi-class prediction. The architectures are illustrated in Figure 3.

a) Regular PGP (baseline): PGP [8] consists of three modules: a graph encoder, a policy header and a trajectory decoder. The graph encoder encodes the motion of both the target agent and surrounding agents using Gated Recurrent Units (GRUs) [35]. Social interactions are encoded by using scaled dot product attention on the surrounding agents and their nearby nodes and are represented as node features on the constructed graph. The final node encodings result from using a Graph Neural Network (GNN) consisting of graph attention layers [36] to retrieve the relationships between the nodes on the graph.

Next, the policy header uses the output of the GNN together with the motion encoding of the target agent to output likely and feasible traversals as transitions between nodes in the lane graph, which are then fed to the trajectory decoder to predict smooth motion dynamics in the form of trajectories. The trajectory decoder aggregates the motion encoding of the target agent with policy information for each unique policy. The policy information serves, together with the encoded motion of the target agent and a latent variable, as input to a Multi-Layer Perceptron (MLP) which outputs the future trajectories.

We add a separate motion encoder for surrounding cyclists as a general modification to PGP to make it more suitable for trajectory prediction in dense city environments, which tend to be rich in the number of VRUs. The vanilla PGP model only accounts for surrounding cyclists by treating them as vehicles. However, cyclists influence the behaviour of other agent classes with their unique dynamics in urban traffic. Therefore, we add a separate GRU encoder for cyclists to the model to capture this relationship.

b) Class-aware PGP (PGP-CA): We propose a class-aware variant of the PGP model to make it more suitable for multi-class trajectory prediction, by conditioning the trajectory decoder on the target agent class. We add the target agent class as input to the trajectory decoder to allow the model to weigh the importance of the motion encodings and sampled lane graph traversals differently for each class. We believe that this feature will help the model to more effectively capture the behaviour of each agent class. More specifically, we believe that the predictions for vehicles are more bound to the lane graph, whereas pedestrians have more freedom to move as the model learns to leverage the motion history instead of the lane graph as most important prior to their future trajectory.

V. Experiments

A. Experimental Setup

1) Datasets: In our experiments, we use both the View-of-Delft Prediction dataset and the nuScenes [13] dataset. We evaluate our models on two versions of the View-of-Delft Prediction dataset, with a two-second and six-second prediction horizon. To compare nuScenes to our dataset, we follow their setup and use a two-second observation window and a six-second prediction horizon. As this version of our dataset contains few prediction instances, we investigate pre-training PGP on nuScenes and fine-tuning the models on the View-of-Delft Prediction dataset. We also experiment with a one-second observation window and a two-second prediction horizon for short-term trajectory prediction, as dense urban traffic is challenging due to partial occlusion, uncertainty in the behaviour of VRUs, and dense interactions between agents and road elements that are shared by different agent classes. The number of target agents is shown in Table III for both versions of our dataset.

TABLE III: Breakdown of View-of-Delft target agents.

Agent Class	**# of Targets**	
	$T = 2$ s	$T = 6$ s
Vehicles	228	120
Cyclists	159	62
Pedestrians	444	120
Total	844	306

2) Metrics: We adopt the widely used minimum Average Displacement Error (minADE/mADE) (↓) and Miss Rate

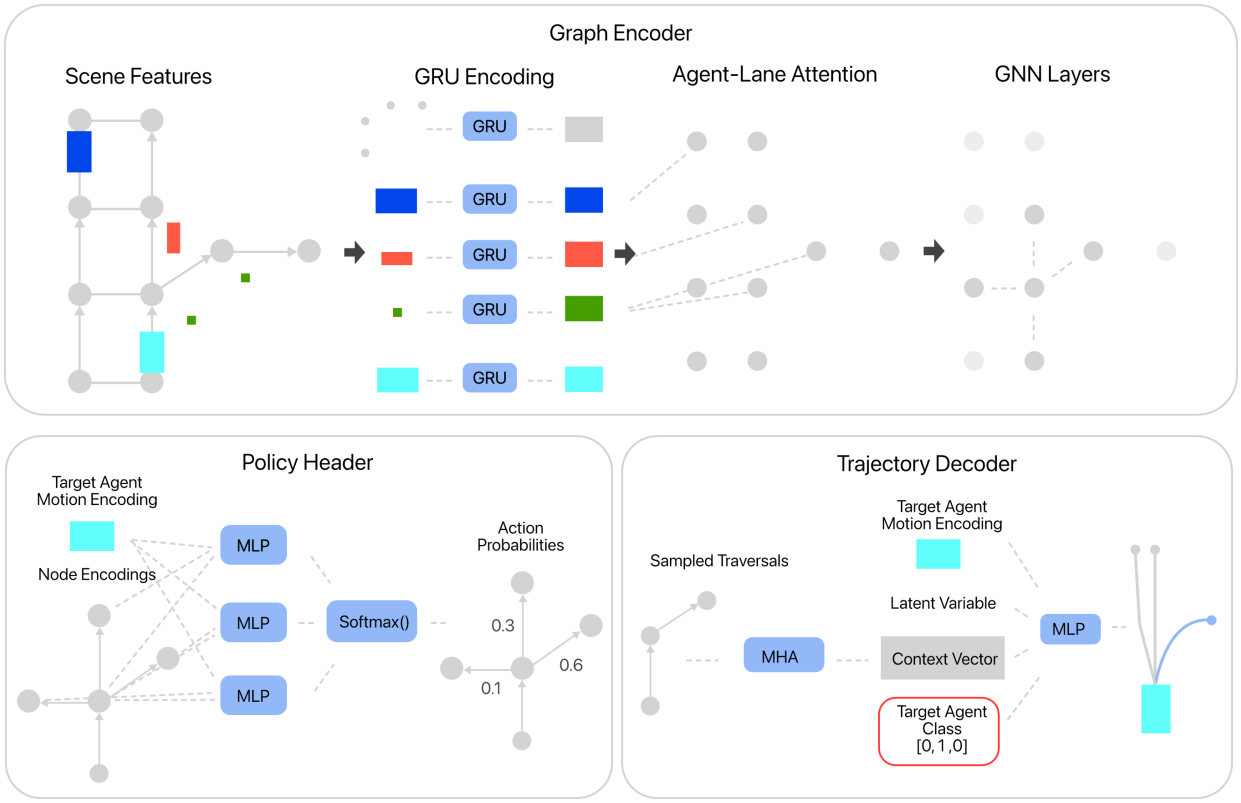


Fig. 3: Architecture of the PGP [8] model with our proposed modifications for View-of-Delft Prediction. Vehicles are depicted in dark blue, cyclists in red, pedestrians in green, and the target agent in cyan. We modify the model to make it suitable for multi-class prediction in urban traffic by adding a separate motion encoder for cyclists and propose an architectural change by providing class labels to the trajectory decoder. The model with the addition of target agent class information (outlined in red) is referred to as PGP-CA.

(MR) (↓) metrics for $K = \{5, 10\}$ predictions. The minADE is the lowest average Euclidean distance between the ground truth trajectory $\mathbf{y}$ and set of K predicted trajectories $\{\hat{\mathbf{y}}^{(1)}, ..., \hat{\mathbf{y}}^{(K)}\}$ over the prediction horizon T:

$$\min \text{ADE}_K = \min_{i \in \{1,...,K\}} \frac{1}{T} \sum_{t=1}^{T} \left\| \mathbf{y}_t - \hat{\mathbf{y}}_t^{(i)} \right\|_2 . \quad (1)$$

The MR is defined as the fraction of scenes where the final location $\hat{\mathbf{y}}_T$ does not lie within a threshold distance r of the ground truth $\mathbf{y}_T$ for any of the K predictions:

$$\text{MR}_K = \frac{1}{N} \sum_{n=1}^{N} H \left(\min_{i \in \{1,...,K\}} \left\| \mathbf{y}_{T,n} - \hat{\mathbf{y}}_{T,n}^{(i)} \right\|_2 - r \right) , \quad (2)$$

where $H(\cdot)$ is the Heaviside step function. We choose thresholds r of 1.5 m and 0.5 m for prediction horizons of 6 s and 2 s, respectively, to account for the distance agents can travel over these horizons.

3) Baselines: In addition to the baseline PGP model, we adopt a constant velocity model (CVM) [37] to assess the necessity for non-linear trajectory prediction in our dataset. A constant velocity model extrapolates the final observed velocity of an agent from the final observed pose over the entire prediction horizon. We select PGP [8] as the state-of-the-art model in our baseline suite. As aforementioned in section IV, this model is modified to encode the motion of surrounding cyclists separately next to vehicles and pedestrians and capture their unique dynamics.

B. Implementation Details

For training models on the VoD Prediction dataset, we use a batch size of 128. We train each model for 1000 epochs, with a constant learning rate of 0.001. No data augmentation is used for training on the VoD Prediction dataset. For pre-training on nuScenes [13], we follow the setup as described in [8]. For each experiment, we perform five evaluation runs with different seeds (1 till 5).

C. Domain Shift

We first evaluate the PGP model on the View-of-Delft Prediction dataset for different training setups, i.e. nuScenes, VoD Prediction or both, to gain insight into the domain shift between the datasets and the level of difficulty of our proposed dataset. Our hypothesis is that urban traffic is more complex given the high amount of interactions between agents and lack of segregated road elements per agent class, and thus only models that are trained on our dataset are able to predict these scenarios accurately.

In Table IV we observe that training the model just on nuScenes [13] results in greater errors than training on VoD Prediction only, e.g. 156.04% higher minADE_5 and 107.79%

TABLE IV: Prediction performance of PGP [8] model on the View-of-Delft Prediction test set for a prediction horizon of $T = 6$ s and various training configurations.

Training Datasets		K=5		K=10	
NuScenes [13]	VoD	mADE ↓	MR ↓	mADE ↓	MR ↓
☑	☐	2.33	0.74	1.69	0.62
☐	☑	**0.91**	**0.43**	0.77	**0.27**
☑	☑	0.97	**0.43**	**0.73**	0.29

higher minADE_{10} when evaluated on VoD prediction, while evaluation with this training setup on nuScenes [13] shows results that are in line with [8]. This indicates that there is a domain shift between the two datasets.

Furthermore, pre-training on nuScenes [13] and fine-tuning on View-of-Delft Prediction results in an overall increase of 6.59% in minADE_5 and decrease of 5.19% in minADE_{10} compared to training on View-of-Delft Prediction only, demonstrating that the two datasets are not fully complementary. This is expected as nuScenes [13] only contains prediction instances for vehicles and has very different traffic dynamics. To gain a better understanding of the effect of pre-training on prediction performance, we present the metrics per agent class in Table V and Table VI.

D. Quantitative Results

In Table V we present the results of different prediction methods for $T = 6$ s and $T = 2$ s prediction horizons. Detailed numerical results including standard deviation are listed in Appendix D. We evaluate the models for a six-second horizon to allow comparison between the nuScenes and VoD Prediction datasets. The poor performance of the constant velocity model on the metrics for both horizons shows that the dynamics in our dataset are highly non-linear, and we thus require additional information to effectively predict dense urban traffic situations.

For $T = 6$ s, both the PGP baseline and PGP-CA model significantly outperform the CVM, showing the ability of the model to learn non-linear dynamics. Comparing the PGP baseline with the PGP-CA model, we denote an overall decrease of 8.49% and 13.92% in minADE_5 and minADE_{10} respectively. Moreover, PGP-CA increases the prediction performance for $K = 10$ for each agent class, both measured in minADE_{10} and MR_{10}. This demonstrates the importance of class information for mixed-traffic prediction settings for longer prediction horizons and proves the effectiveness of our proposed change.

For the models that are pre-trained on nuScenes [13] and fine-tuned on VoD Prediction, we observe that PGP-CA outperforms PGP for predicting future trajectories for VRUs. This increase in prediction performance for VRUs comes at the cost of decreased prediction performance for vehicles. Overall, the PGP-CA outperforms the PGP model, with a 7.95% decrease in minADE_5.

However, for a shorter horizon of $T = 2$ s, the PGP-CA model shows a less clear difference in results, performing worse than the PGP model for $K = 10$ predictions, while performance for $K = 5$ predictions is slightly better.

VI. Discussion

To gain insight in the cause of the numerical domain shift between nuScenes and VoD Prediction and thus further stress the need for the VoD Prediction dataset, we study qualitative results of PGP trained and evaluated on nuScenes [13] and VoD Prediction, as shown in Figure 5 and 6 for prediction instances of nuScenes and Figure 7a, 8a and 9a, for prediction instances of VoD Prediction. Overall, we note that PGP excels in (multi-lane) straight prediction scenarios, where there are relatively few interactions with other agents, which is a frequently recurring scenario in the nuScenes dataset. Furthermore, we observe good predictions for turning cases, and good multimodality in general, observed on both datasets. Prediction instances with dense interactions between vehicles and VRUs are more sparse in the nuScenes dataset. In Figure 6 we observe that PGP trained and evaluated on nuScenes has difficulty accurately predicting pedestrian crossings, even at dedicated pedestrian crossing areas, which stresses the importance of evaluating these trajectory prediction methods in a dense urban setting. As shown in Figure 7a, we observe that the same model trained and evaluated on our dataset has an increased ability to account for interactions of vehicles and VRUs.

In addition, we observe that PGP pre-trained on nuScenes [13] and fine-tuned on VoD Prediction does not perform significantly better than PGP solely trained on VoD Prediction when evaluated on VoD Prediction. Qualitative analysis shows that the datasets are somewhat complementary, in the sense that models trained on nuScenes can learn to predict for cases where the target agents are in no proximity to other agents and are following the lane graph in VoD Prediction. However, trajectory prediction in the urban domain, with VRUs as additional target agents, comes with its own set of challenges. In urban traffic, there is more stochastic behaviour due to the high amount of VRUs, making the followed trajectories less bound to the lane graph, especially for prediction instances of VRUs. In Figure 14, we observe that the pre-trained models rely heavily on the lane graph, as learned during the pre-training phase. This is a relevant prior for the future trajectory of vehicles, but to a lesser extent for cyclists and pedestrians. The pre-trained models therefore overfit towards following the lane graph, leading to a lower policy loss, but no significant performance increase in terms of minimum average displacement error or miss rate. This is due to a lack of performance in cases where the agent is not following the lane graph. For the transfer learning setting with a six-second prediction horizon, the PGP-FT-CA model improves prediction performance for VRUs, as the model relies slightly less on the lane graph for all agent classes. Unfortunately, this leads to decreased prediction performance for vehicles in this setting, as the PGP-FT model is already able to predict very well for vehicles, due to the extensive pre-training for 'lane-bound' vehicles on nuScenes [13].

Next, we study the applicability of a vehicle-based trajectory prediction model (PGP [8]) for multi-class prediction in urban environments and study the performance of our proposed method: PGP-CA. Quantitively, both the PGP model and PGP-

TABLE V: Performance comparison of models trained and evaluated on the VoD dataset with $T = 6$ s and $T = 2$ s prediction horizons. The best performance on each metric is shown in **bold**. CA = Class-Aware, mADE = minimum Average Displacement Error, MR = Miss Rate.

	Method	Vehicle				Cyclist				Pedestrian			
		K=5		K=10		K=5		K=10		K=5		K=10	
		mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓
$T=6$	CVM	4.26	0.93	-	-	4.47	1.00	-	-	2.61	1.00	-	-
	PGP [8]	0.93	0.54	0.75	0.28	**1.34**	0.71	1.20	0.61	0.81	**0.31**	0.70	0.20
	PGP-CA (Ours)	**0.77**	**0.43**	**0.63**	**0.23**	1.40	**0.65**	**1.15**	**0.60**	**0.75**	0.39	**0.59**	**0.14**
$T=2$	CVM	1.68	0.94	-	-	2.09	0.98	-	-	0.91	0.92	-	-
	PGP [8]	0.25	0.25	**0.18**	**0.05**	**0.53**	**0.56**	**0.45**	**0.40**	**0.28**	0.27	**0.23**	**0.14**
	PGP-CA (Ours)	**0.22**	**0.18**	**0.18**	0.10	0.57	0.58	0.51	0.49	**0.28**	**0.26**	0.25	0.20

TABLE VI: Performance comparison of models pre-trained on nuScenes [13], fine-tuned and evaluated on the VoD Prediction dataset with $T = 6$ s prediction horizon. Best performance on each metric is shown in **bold**. FT = Fine Tuning (Transfer Learning), CA = Class-Aware, mADE = minimum Average Displacement Error, MR = Miss Rate.

	Method	Vehicle				Cyclist				Pedestrian			
		K=5		K=10		K=5		K=10		K=5		K=10	
		mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓	mADE ↓	MR ↓
FT	PGP-FT	**0.97**	0.47	**0.67**	**0.26**	1.54	**0.61**	1.11	0.53	0.85	0.37	0.69	0.26
	PGP-FT-CA (Ours)	1.03	**0.45**	0.75	0.28	**1.13**	**0.61**	**0.97**	**0.45**	**0.77**	**0.33**	**0.65**	**0.25**

CA model lead to reasonable results, taking into account the lower speeds that are attained in urban traffic. Especially for vehicles, the metrics indicate that PGP translates relatively well to urban environments, given a MR_{10} of 0.23 and $\min \mathrm{ADE}_{10}$ of 0.63 for our PGP-CA model in a six-second prediction horizon. PGP-CA outperforms PGP for all agent classes, leading to an overall performance increase on VoD Prediction.

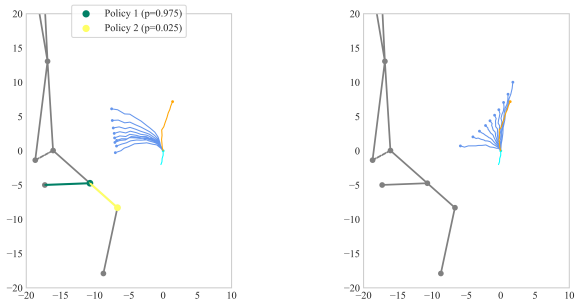
Analysis of the models trained for a two-second prediction horizon shows that there is no significant difference between the PGP and PGP-CA architectures. We argue that the PGP model, which relies heavily on the lane graph to account for longer prediction horizons, in its unadjusted form is not suited for short prediction horizons. We reason that this is due to the sampling density of nodes on the lane graph (once per twenty meters), which is too sparse for the limited amount of distance that target agents tend to move over such a short prediction span, as further studied in Appendix E. For these cases, the inductive prior of the lane graph may thus not be as informative, leading to a large number of noisy features in the model, which limits the learning ability of the model.

Qualitative analysis of the results for a six-second prediction horizon, depicted in Figure 7 till Figure 13, shows that for both PGP and PGP-CA, the policy is informative and serves as a good prior for the future trajectories of all vehicles and some VRU instances, given they are reasonably close to the lane graph. We note that the models in these cases accurately forecast future trajectories while exhibiting multi-modal prediction behaviour. For vehicles, the model is thus able to account for more dense interactions present in the VoD Prediction dataset, as shown in Figure 7. We argue that the prediction performance for cyclists, which numerically leads to the highest errors, rises from the fact that their dynamics are highly non-linear and less bound to the lane graph, while their attained speeds are relatively close to that of vehicles in urban environments.

For pedestrians, we observe another trend, as shown in Figure 10. If a pedestrian is located sufficiently close to the lane graph (e.g. adjacent pavement), it aids the prediction for that instance. However, we note that for instances where the pedestrian is located on a pedestrian domain that is not close to the lane graph, the predictions are being drawn towards the lane graph, leading to inaccurate predictions. This especially holds for the PGP model, as the PGP-CA model relies more on the motion dynamics of the target agent for predicting its future trajectory and thus suffers from this issue to a lesser extent.

To further investigate this phenomenon, we train the PGP baseline model and PGP class-aware model on the six-second variant of the VoD Prediction dataset, where we disable the policy header for pedestrians. Quantitatively, this does not lead to a performance improvement, as the pedestrian instances close to the lane graph suffer from the lack of policy information. Qualitatively, we observe in Figure 4 that more feasible predictions are made for the instances where a pedestrian is located far away from the lane graph, as the predictions tend to rely more on the agent's motion dynamics than the (non-informative) lane graph.

In a further comparison of the PGP and PGP-CA architectures, we observe that PGP-CA relies more on the dynamics of the agent to predict future trajectories, as opposed to PGP, which relies more on the policy sampled over the lane graph. This is in line with our hypothesis in section IV. This increased influence of the target agent dynamics leads to more accurate and feasible predictions, which is supported by the lower miss rate and minimum average displacement errors. This often leads to increased prediction behaviour in challenging and turning cases, as shown in Figure 8, Figure 9 and Figure 11, and proves the effectiveness of our proposed PGP-CA model.



(a) Baseline model (b) Pedestrian policies disabled.

Fig. 4: Policy visualization showing a pedestrian instance located far away from the lane graph for the baseline model and the baseline model trained without policy information for pedestrians. Both the left and right images show the target agent (+ motion history) in cyan, the ground truth in orange and predictions in blue. The lane graph is visualized in grey, with policies overlaid as per the legend.

Finally, we study the shortcomings of PGP (in urban traffic). We note that a general drawback of PGP is that when a mode (e.g. left turn) is not accounted for as sampled traversal in the policy header, the model fails to predict trajectories for that mode, as illustrated in Figure 12. The multimodality of the predictions thus strongly relies on the quality of the sampled traversals, as given by the policy header, which limits the robustness of the model. For the nuScenes dataset, this is shown in Figure 5c, where PGP fails to capture the mode (left turn) followed by the ground truth trajectory. It, however, captures the possibility of a right turn. The predictions are divided between the feasible modes given by the policy header, based on the likelihood of each sampled traversal. A similar case for the VoD dataset is shown in Figure 12a. Specific to the urban domain, we note that VRUs tend to switch from road elements (e.g. from lane to sidewalk) easily, which makes the agent deviate from the lane graph. Both PGP and PGP-CA are unable to account for these cases, as shown in Figure 13. Lastly, we note that the PGP model encodes agent interactions via the lane graph. This also means that in cases when the lane graph is located far away from the target agent, social interaction is not accounted for properly. We recommend encoding the remaining static road elements using the allowed agents per element, in addition to the lane graph, to increase scene understanding. This will result in a different traversable space per agent type. This is enabled by the VoD Prediction dataset, to resolve the shortcomings for prediction in dense urban traffic.

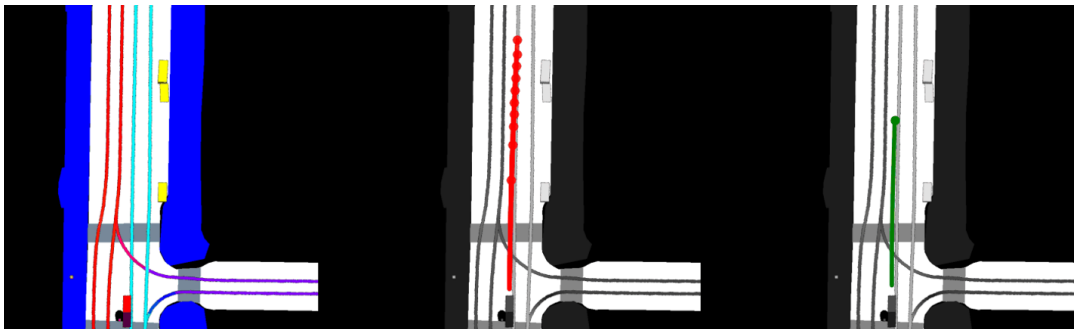
VII. Conclusions

We introduced the View-of-Delft Prediction dataset, an extension of the View-of-Delft (VoD) dataset, enriching the available sensor information with vectorized map information. We show that there is a significant domain shift between the urban View-of-Delft Prediction dataset and the widely used nuScenes [13] dataset, highlighting the need for urban prediction datasets, which are rich in number of Vulnerable Road Users (VRUs) and dense interactions. Our dataset is a step towards bridging the gap, enabling future research on trajectory prediction in complex urban traffic. Further, we modify the vector-based PGP model [8] by presenting the class-aware PGP-CA, which leads to increased prediction performance on VoD Prediction.

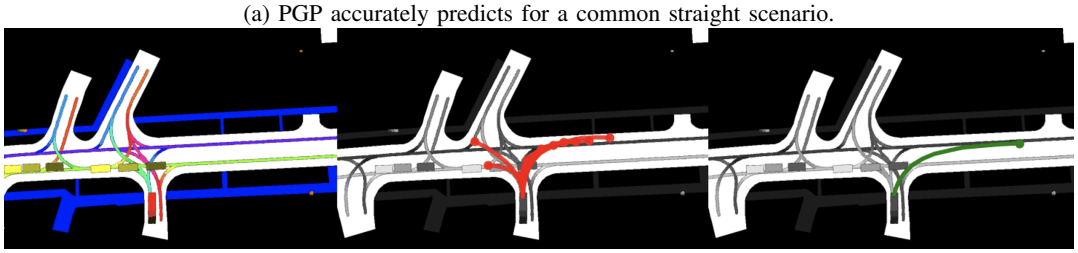
In experiments, we analysed the domain gap between the vehicle-heavy nuScenes dataset and our VRU-rich View-of-Delft Prediction dataset using a modified PGP [8] model. The results show that there is a significant domain shift between the datasets, as training only on nuScenes results in a 107.79% higher $\min \mathrm{ADE}_{10}$ on the VoD Prediction test set than training the model on our dataset.

We proposed a class-aware version of the PGP model, which outperforms the original model for each single agent class, leading to an average performance improvement of 13.92% in $\min \mathrm{ADE}_{10}$ for a six-second prediction horizon. Qualitatively, we observe that PGP-CA has an increased ability to differentiate between class dynamics.

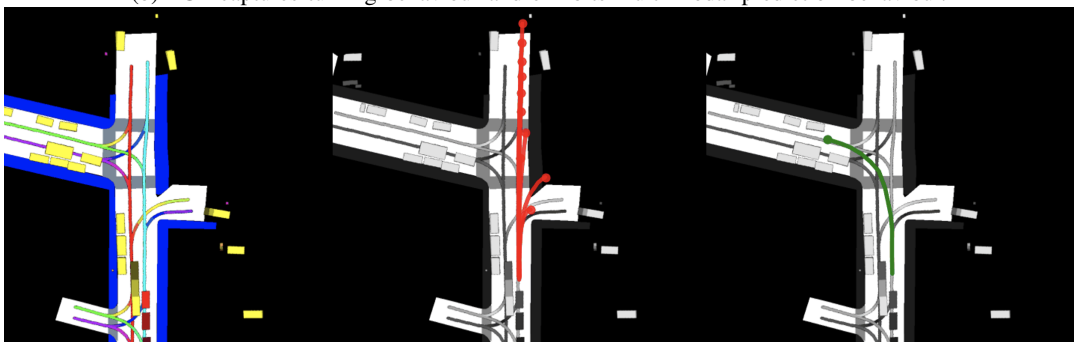
Future work includes encoding map context differently for each agent class to account for their unique interactions with the static environment and ensuring that predictions are scene-compliant.



(a) PGP accurately predicts for a common straight scenario.

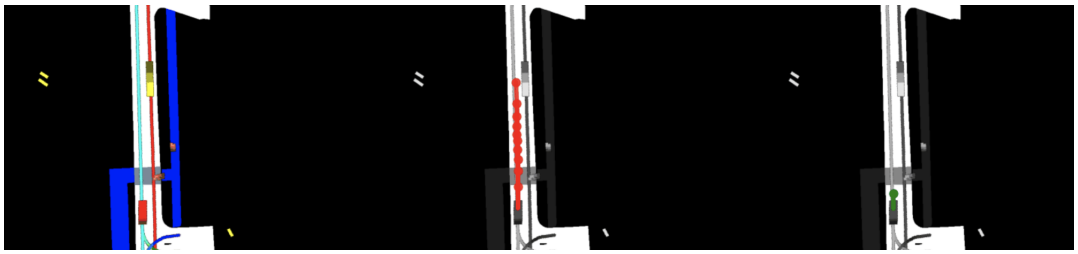


(b) PGP captures turning behaviour and exhibits multi-modal prediction behaviour.

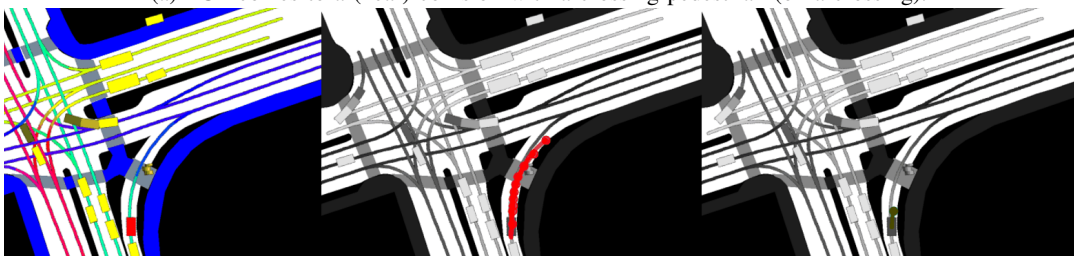


(c) Failure Case: PGP fails to capture the mode (left turn) followed by the ground truth trajectory.

Fig. 5: Common prediction scenarios in the nuScenes dataset. The left image shows the prediction scenario, the middle image the predictions and the right image the ground truth trajectory.

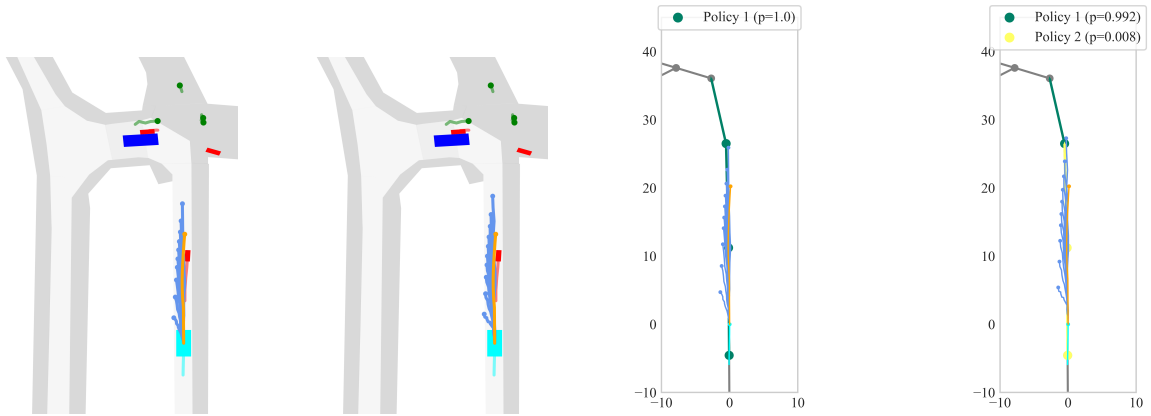


(a) PGP comes to a (near) collision with a crossing pedestrian (on a crossing).



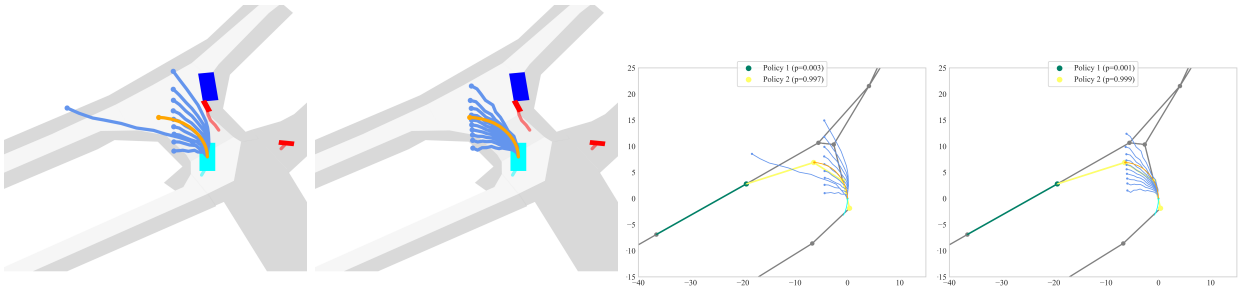
(b) PGP predicts some feasible and some infeasible predictions for a crossing pedestrian scenario (on a crossing).

Fig. 6: Prediction scenarios with vehicle - VRU interactions in the nuScenes dataset.



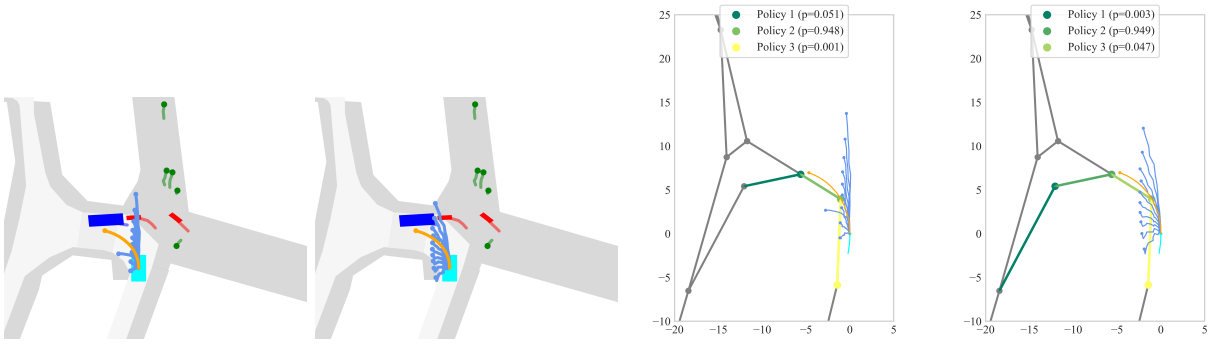
(a) PGP (b) PGP-CA (c) PGP (d) PGP-CA

Fig. 7: Prediction instance of a vehicle successfully dealing with an interaction with a cyclist. Both the left and right images show the target agent (+ motion history) in cyan, the ground truth in orange and predictions in blue. The left images show the prediction scene, with road topology and surrounding agents, where dark blue denotes vehicles, red is used for cyclists and green for pedestrians. The right images show the lane graph with the policies visualized.



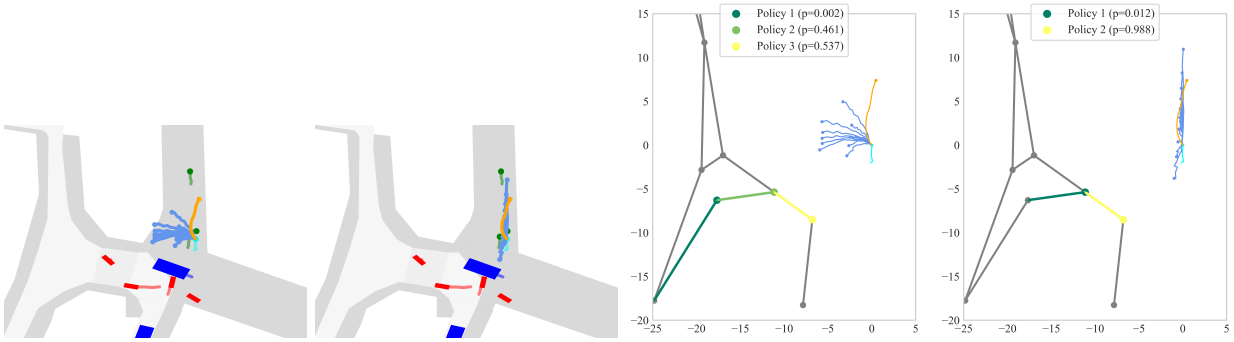
(a) PGP (b) PGP-CA (c) PGP (d) PGP-CA

Fig. 8: Prediction scenario for a vehicle making a turn. Both models capture the turning dynamics. The PGP-CA model better predicts the dynamics, leading to a more accurate prediction of the turning case.



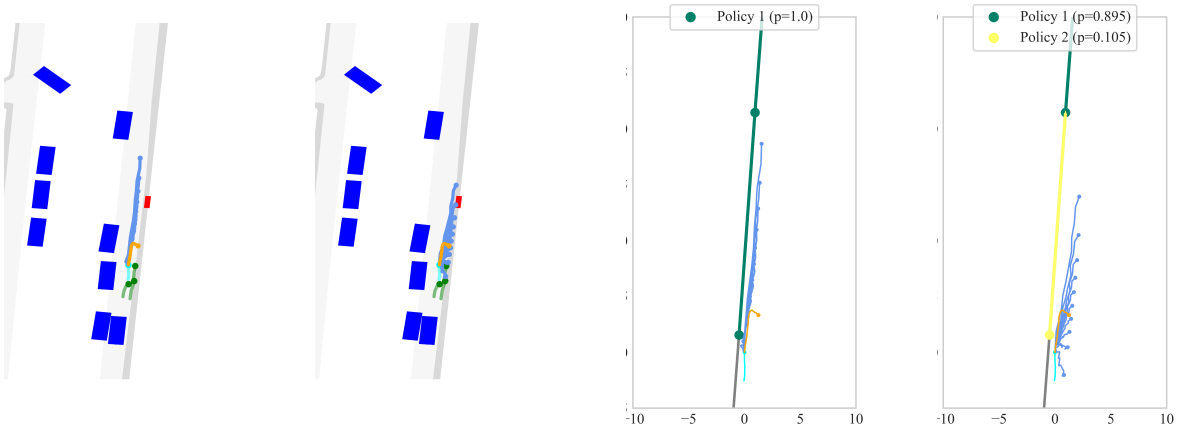
(a) PGP (b) PGP-CA (c) PGP (d) PGP-CA

Fig. 9: Prediction scenario for a vehicle making a turn. Both models capture the turning dynamics. The PGP-CA model better predicts the dynamics, leading to a more accurate prediction of the turning case.



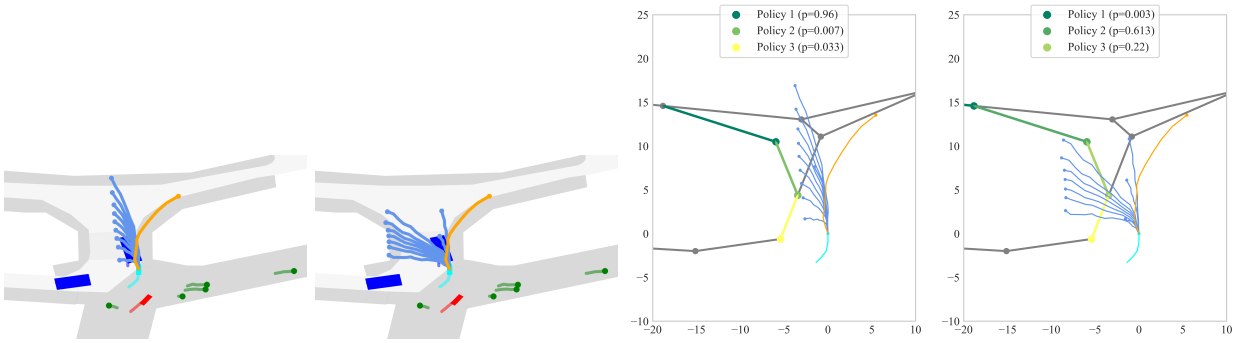
(a) PGP (b) PGP-CA (c) PGP (d) PGP-CA

Fig. 10: Prediction scenario for a pedestrian located far away from the lane graph. PGP-CA is better able to account for these cases, as it is not being drawn towards the lane graph as much as the PGP model.



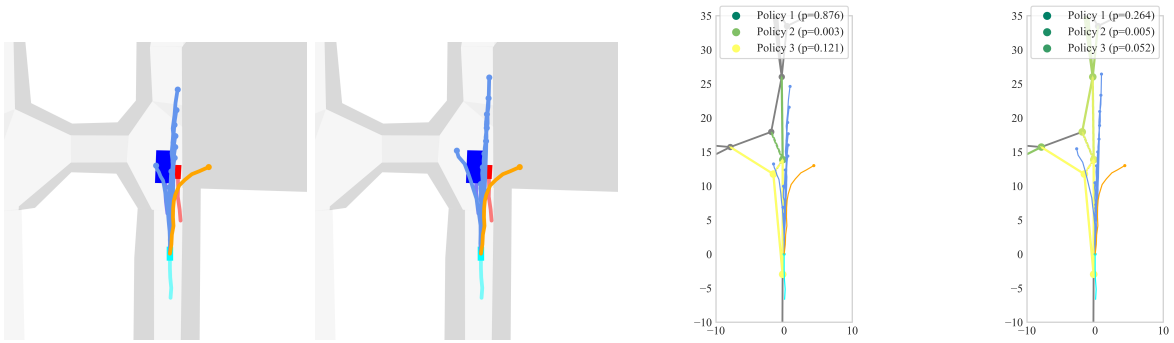
(a) PGP (b) PGP-CA (c) PGP (d) PGP-CA

Fig. 11: Prediction scenario for a pedestrian making a turn during the ground truth trajectory. Only PGP-CA predicts the challenging dynamics, leading to a more accurate prediction for this instance.



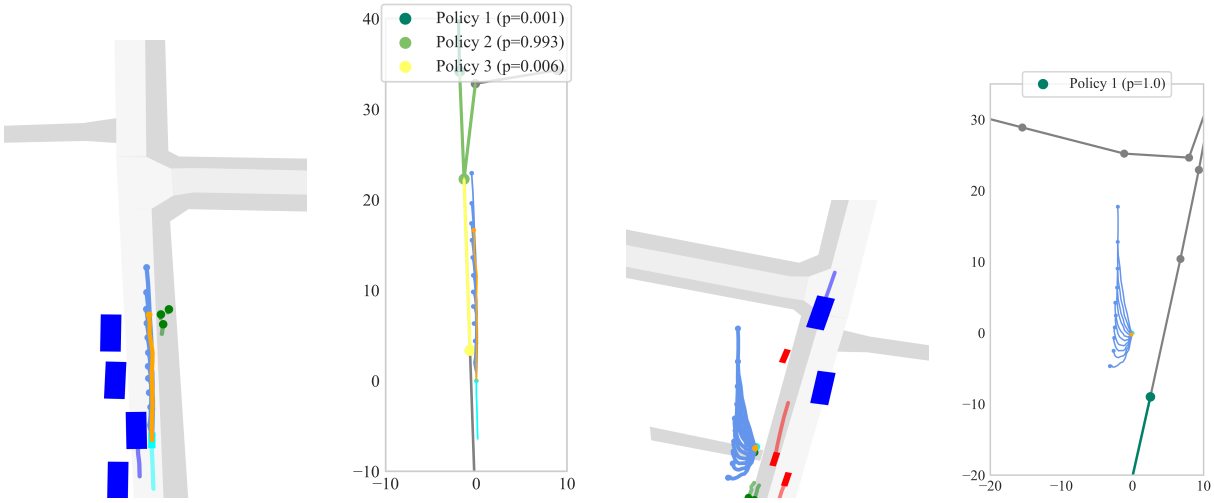
(a) PGP (b) PGP-CA (c) PGP (d) PGP-CA

Fig. 12: Failure Case: Prediction scenario for a cyclist. Here, an incomplete policy outcome leads to bad predictions for both models.



(a) PGP (b) PGP-CA (c) PGP (d) PGP-CA

Fig. 13: Failure Case: Prediction scenario for a cyclist. Both models are unable to handle agents that follow the lane graph and suddenly deviate from it during their ground truth trajectory by switching to another static map element.



(a) PGP-FT (b) PGP-FT (c) PGP-FT (d) PGP-FT

Fig. 14: Prediction scenarios for VRUs with a cyclist clearly following the lane graph and a pedestrian who does not. The PGP-FT model can accurately predict the first case, where it performs less for cases similar to the latter case.

APPENDIX A
EURO-PVI ANALYSIS

In this section, we study the accuracy of the available information in the Euro-PVI [11] dataset. Euro-PVI shares segmented maps from OpenStreetMap as static map information in their dataset. We argue that these maps from OpenStreetMap are insufficiently accurate for autonomous driving tasks, as their positional accuracy was found to be within 1.57 meters in the following case study that compares accurate land surveying reference data to the OpenStreetMap of that same region: [34].

We assert that the minimum accuracy of static map information should be more accurate than this, ideally within a couple of centimetres, as such an error margin could lead to a sidewalk located on a lane and vice versa. To investigate the accuracy of OpenStreetMap in The Netherlands, we compare the accuracy of our annotations and aerial images to the OpenStreetMap map from Delft in Figure 15. Here, we again observe the limitations of using the map information of OpenStreetMap directly. As shown in Appendix B, the VoD dataset is consistent with our aerial images and annotations. Overlaying these images and annotations over the OpenStreetMap, we see that the OpenStreetMap map information deviates at many points from the topology shown in the aerial image and that the provided map information is less accurate than the provided map information in the VoD Prediction dataset. The most striking case is the mislocated lane on the right, where a sidewalk is placed on a road and vice versa. This could lead to safety-critical situations, where a self-driving car would predict its future trajectory on a sidewalk, increasing the chance of severe collisions. Therefore, we reason that OpenStreetMap is not sufficiently accurate for autonomous driving-related tasks.

Additionally, we study the agent tracks in Euro-PVI to compare the type of traffic to our dataset. We observe that the agent tracks of the recording vehicle are noisy, leading to yaw deviations greater than 60 degrees between consecutive frames recorded at 10 Hz, which is physically impossible given the non-holonomic constraints of vehicles. When we plot the surrounding agent motion around the recording vehicle, as shown in Figure 16 for surrounding cyclists, we observe that ego-motion correction using these noisy tracks of the recording vehicle leads to an inaccurate overview of the surrounding cyclist motion as there tend to be multiple collisions with the recording vehicle located at the origin. In addition, we observe many infeasible trajectories following a zig-zag pattern. This complicates a feasible analysis of the type of interactions in Euro-PVI compared to our dataset.

APPENDIX B
ANNOTATION PIPELINE

We present our developed procedure to generate accurate map information using open-source tools, to enable researchers to follow this pipeline for their use case. To the best of our knowledge, we are the first to release this procedure in the field of autonomous driving datasets, which often relies on in-house tools. First, we explain the georeferencing procedure, used to match the annotations to the correct location, followed by the annotation and labelling process and the preprocessing of the road elements, such that a vectorised representation results.

A. Georeferencing

The goal of the georeferencing phase is to match the coordinates of the map annotations to the original VoD dataset, such that the map annotations are consistent with the data in the VoD dataset. For accurate and detailed annotations, we rely on aerial images from PDOK[2], an open-source dataset with accurate geo-information about the Netherlands. We retrieve these aerial images based on the locations of the recording vehicle in the VoD dataset and georeference them in the same manner.

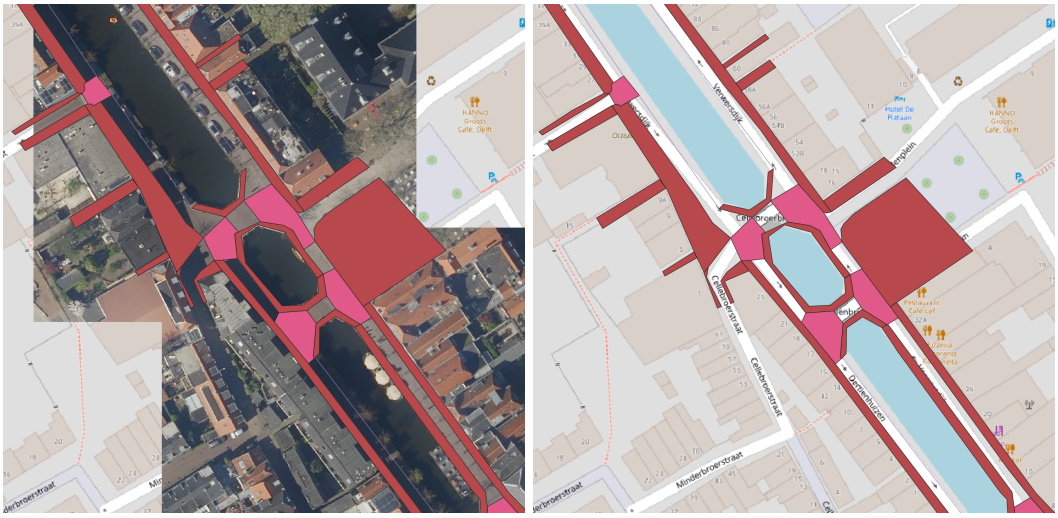
To end up with a georeferenced aerial image, which serves as our annotation region, we undertake the following steps:

- First, select the annotation region by scanning the camera frames of the VoD dataset. We utilize the GPS location of the recording vehicle in the VoD dataset, to obtain the recording locations to generate annotations for. If a frame is roughly in the region of interest (we annotate a region of 100 m by 100m, with the recording vehicle as centre), select that frame number as [FRAME NUMBER].
- If not done yet, clone the prediction branch from the VoD git repository first at a desired location on your machine.
- Open the terminal on your local machine and navigate towards the following directory: `view-of-delft-dataset`.
- Install the required Python packages as a virtual environment using the following command: `conda env create -f environment.yml`.
- Navigate towards the folder: `pdok-wms-request`.
- To activate the right environment, run the following command: `source 01-activate.sh`.
- Run `python aerial_maps.py -f [FRAME NUMBER]`. [FRAME NUMBER] follows a five-digit format (e.g. frame 15, thus becomes 00015). This script will save both a [FRAME NUMBER].png and [FRAME NUMBER].tiff file in the current folder. The latter is the aerial image with an equally spaced grid of 100 georeferenced points overlaid, where each coordinate on the grid is matched to a pixel in the aerial image.
- Repeat the above such that you have tiles of aerial images that cover the locations that need to be annotated, in this case where the recording vehicle has driven. We will use this as the starting point for the next step.

Some practicalities to take into account: to retrieve the right region to annotate, the script makes use of the BBOX finder[3] and PDOK server. Requesting bounding boxes based on their centre location is preferred, as the bounding boxes of BBOX finder are inaccurate by a couple of centimetres, leading to slight inconsistencies between different georeferenced aerial images. Using the centre coordinates, this slight inaccuracy is omitted. For the annotation and georeferencing process, we adhere to a static coordinate frame (EPSG:28992) to

[2]https://www.pdok.nl/

[3]http://bboxfinder.com/



(a) Aerial images with our static map information overlaid.

(b) OpenStreetMap with our static map information overlaid.

Fig. 15: Comparison of our map annotations with the OpenStreetMap map. We observe inconsistencies in where lanes and sidewalks are located. Near the large square pedestrian area, we observe a mislocated lane on the pedestrian area.

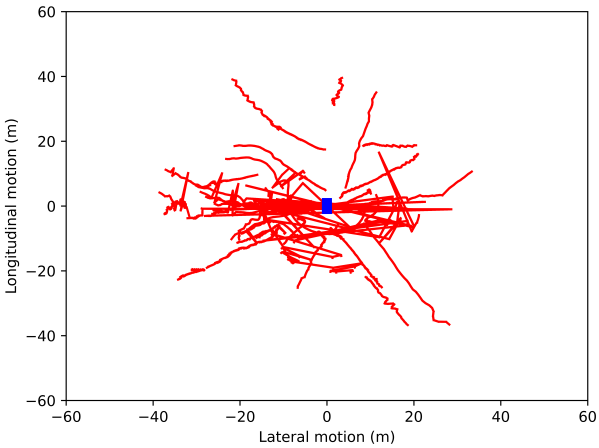


Fig. 16: Surrounding cyclist tracks from the first thirty scenes as observed by the recording vehicle, depicted by the blue rectangle, in the Euro-PVI dataset [11].

omit slight deviations in location, due to dynamic coordinate systems being handled differently between programs (QGIS[4] (annotation tool) and the python packages: osgeo, gdal and geopandas). Therefore, we convert the centre coordinate and bounding boxes to the static coordinate frame before sampling the grid over the aerial image.

B. Annotation & Labelling

Now we have a set of georeferenced aerial images, we are ready to proceed with the annotation and labelling phase. First, we load the georeferenced images as rasters into QGIS, the annotation and labelling tool, such that tiles of aerial images result over the recording locations, on which the annotations can be drawn. Then we start the annotation and labelling process, which is set out in more detail below.

As mentioned in section III, we annotate lanes by marking their right and left boundaries, as they are more clearly visible compared to the lane centreline, and as interpolation of the lane centreline from the lane boundaries will lead to more accurate results. The directionality of the lane centreline is determined by the direction of the right lane boundary, which should thus be drawn in the correct direction. Bidirectional lanes may be annotated in one direction and labelled as bidirectional. The preprocessing code will take care of creating the lane copy in the other direction. Furthermore, we denote intersections, pedestrian crossings and the pedestrian domain by enclosing the area as a polygon. We export the annotations and labels per category of road elements as geopackage, which is a widely used format to store and handle geospatial data. The process looks as follows:

- In QGIS, open a new project.
- In the top menu bar, select `Web > QuickMapServices > OSM > OSM standard`. This will load an OpenStreetMap of the world, which can serve as a rough reference check to see whether the georeferencing is somewhat correct.
- In the bottom right, the current coordinate system is shown. Click the coordinate system to select the right coordinate system. Select ‘Amersfoort/ RD New EPSG:28992’. The OpenStreetMap is now shown as seen from the top of the North Pole.
- From the `map_annotation` repository, load the template geopackage layers (.gpkg): Layers, Crosswalk, Off-Road and Intersections. These already contain the proper labelling fields, geometry type and coordinate system for each road element.
- In QGIS, select `Layer > Add layer > Add`

[4]https://qgis.org/en/site/

raster layer. Open the [FRAME NUMBER].tiff file. Repeat this step for all the generated georeferenced aerial images.
- In the layer menu on the bottom left, right-click on the [FRAME NUMBER].tiff file and select Zoom to layer. You are now ready to annotate.
- In the layer menu on the bottom left, select the right geopackage layer (lanes, crosswalks, intersections or off-road).
- Click the pencil icon to toggle editing.
- Based on the element you are drawing, select either the vertex tool (for lanes) or the polygon tool (for the other elements).
- Start drawing the road element by adding points on desirable locations, using a left mouse click.
- To finish a drawn element, use right mouse click. In the pop-up window, label the element following the required labels per element as set out in Table II and the following taxonomy:
  - For element and lane identifiers, we use a numerical numbering system.
  - For predecessors and successors, enter the lane identifier of the predecessor and successor lanes.
  - To indicate the left or right boundary, we use a boolean.
  - To indicate the road type, we use the following system: 1 = urban road, 2 = car road, 3 = bike lane, 4 = tram/bus lane.
  - To indicate the boundary type, we use the following system: 1 = solid marking, 2 = dashed marking.
  - To indicate the allowed agents on a specific element, we use the following system: 1 = pedestrian, 2 = cyclist, 3 = car, 4 = bus, 5 = tram, 6 = other. Multiple agents can be entered per road element.
- This step only holds for lanes that share boundaries with other lanes:
  - Right-click the layer in the layer menu, select Open attribute table.
  - Next, toggle editing by pressing the pencil icon (same icon as in step 2).
  - Select the lane element you would like to duplicate.
  - Duplicate the layer.
  - Adjust the element_id of the copied element accordingly and save the changes.
- Repeat the above steps for all road elements. Save the file. The .gpkg layers now contain the right information to start preprocessing the static map information.

C. Verification

To verify the georeferencing process, we again use the aerial_maps.py file, where we convert the centre coordinate and corners of the sampled grid back to the dynamic coordinate frame (UTM/WGS:84) and compare it to the initial recording vehicle coordinate location from the corresponding frame [FRAME NUMBER]. Additionally, we perform visual checks from an onboard and BEV perspective by plotting the annotations and agents in the VoD dataset to see whether their location (on the road) corresponds with the generated annotations, in test_maps.py. This indicates consistency between the georeferenced map information and the original VoD dataset. An example reference check is shown in Figure 17 and 18.

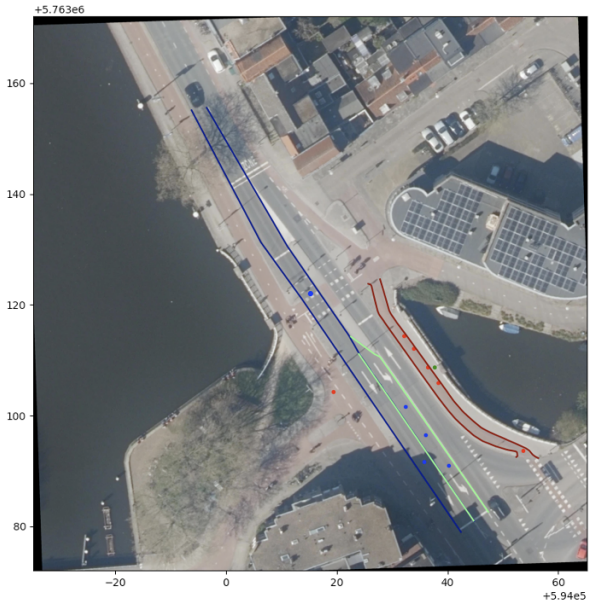


Fig. 17: Annotation verification from a birds-eye view. Drawn lanes are both consistent with the aerial image and the surrounding agents denoted as dots.

Fig. 18: Reference frame for agent location and type for Figure 17 from the VoD dataset.

D. Map Information Preprocessing

To generate vectorised representations from the map information, we require preprocessing of the raw annotated road elements. The git repository map_annotation contains the relevant code to perform the preprocessing steps. The following steps are required:

- If not done yet, clone the git repository map_annotation first at a desired location on your machine.

- Open the terminal on you local machine and navigate towards the following directory: `map_annotation`.
- Install the required python packages as a virtual environment using the following command: `conda env create -f environment.yml`.
- To activate the right environment, run the following command: `source 01-activate.sh`.
- Navigate towards the folder: `data`.
- Copy the final annotation geopackages to the data folder in the repository.
- Next we run the file `preprocess.py`.

This file will take care of the following operations. First, we revert the coordinates of all road elements back to UTM (WGS:84), to be consistent with the VoD dataset. To save annotation time/cost, we automate the handling of bidirectional lanes, requiring them to be drawn once in an arbitrary direction. We duplicate lanes labelled as bidirectional and reverse the duplicate lane. Given the directionality and possible traversals of surrounding traffic elements, such as adjacent intersections, we determine the feasible predecessor-successor combinations and assign them as labels to the resulting pair of uni-directional lanes. We also update the remainder of the lane labels such as element identifier and lane identifier accordingly, before interpolating the lane centreline, as described earlier. Similar to lane centrelines, it is difficult to annotate lane connections on intersections directly, as their path is not directly visible on aerial images. Therefore, we match the predecessor and successor lanes to an intersection that connects both lanes so that we can generate a connection line. As most prediction models rely on the lane centreline instead of lane boundaries, we retrieve the lane centreline by sampling a thousand equally spaced points over both lane boundaries and interpolating the lane centreline. The directionality of the lane centreline is determined by the direction of the right lane boundary, a convention which was adhered to during annotation. Between the interpolated lane centrelines we interpolate a natural connection line between the lanes enclosed by their matched intersection. This way we retrieve organic lane connections, to which we automatically assign labels based on the labels of their connecting lanes.

APPENDIX C
DATASET SOFTWARE KIT

Accompanying the preprocessed map information, we also release a modular software kit for the View-of-Delft Prediction dataset. This modular software kit leverages the agent information and their corresponding tracking identifier in the VoD dataset and the preprocessed map information to generate an object-centric representation of each dataset scene with all required features.

First of all, it allows for setting the required scene parameters, such as observation length, prediction horizon, the distance around the target agent and frame rate in a config file, for which the accompanied dataset splits will be calculated. For larger prediction horizons, we incorporate some overlap in the frames, to increase the size of the dataset. For a six-second prediction horizon, we start a new scene halfway. Scenes with overlap are assigned to the same split. For each scene, the target agents will be determined based on their presence in every frame of the scenes.

Based on the resulting scenes and accompanying target agents from the scene selection process, the software kit handles all required operations to generate a vectorized representation for each scene. For both the target and surrounding agents, it calculates the kinematic states, such as velocity, acceleration and rotational velocity, during the scenes based on the positional information in every frame. Additionally, it also generates the road topology based on the target agent's location and orientation. We use homogeneous transformations to represent the scene in an object-centric frame of reference, resulting in a unique scene for each target agent. Following, we vectorise the road elements by discretizing the lane elements and labelling nodes whether they are located on an intersection or crosswalk. Furthermore, we determine whether lanes are neighbouring lanes to determine lane-switching possibilities. Lastly, we vectorise the kinematic vectors of both the surrounding agents and target agents, such that a vectorised representation in an object-centric reference frame is served as input to the model.

A. Obtaining Accurate Agent Trajectories

To minimize the noise in the trajectories in the presented VoD Prediction dataset, we retrieve more accurate and smooth agent trajectories by fusing GPS and IMU localization data.

The GPS locations in the VoD dataset do not possess accuracy up to centimetres, which results in noisy trajectories for the recording vehicle and similarly for all the agents, as they are observed in an onboard setting. The odometry data on the other hand is more smooth and accurate but requires to be calibrated to a global coordinate in the dataset scene.

For each scene, we calculate the closest GPS location of the recording vehicle to the lane centreline it is positioned on. Under the assumption that the recording vehicle is located on the lane centreline, we update the GPS location of the closest point of the recording vehicle accordingly. Next, we fuse this global coordinate with the locally accurate and smooth odometry data to retrieve a more accurate and smooth agent trajectory. As other agents are observed from the recording vehicle, all agent trajectories are corrected by updating the trajectory of the recording vehicle. Using the updated agent trajectories results in a significant boost in prediction performance for all models.

APPENDIX D
NUMERICAL RESULTS

This Appendix consists of Table VII, VIII and IX and lists the average results from the experiments including standard deviation based on five evaluation runs, both split out per agent class and measured over the total dataset.

APPENDIX E
MISCELLANEOUS EXPERIMENTS

Over the course of this work, we have performed numerous additional experiments that did not end up in the main body of

TABLE VII: Numerical results of models trained and evaluated on the VoD dataset with a $T = 6$ s prediction horizon. The best performance on each metric is shown in **bold**. CA = Class-Aware, mADE = minimum Average Displacement Error, MR = Miss Rate.

		CVM	PGP	PGP-CA
Vehicle	$\text{mADE}_5 \downarrow$	4.26 ± 0.0	0.93 ± 0.04	**0.77 ± 0.03**
	$\text{MR}_5 \downarrow$	0.93 ± 0.0	0.54 ± 0.01	**0.43 ± 0.05**
	$\text{mADE}_{10} \downarrow$	-	0.75 ± 0.01	**0.63 ± 0.01**
	$\text{MR}_{10} \downarrow$	-	0.28 ± 0.02	**0.23 ± 0.02**
Cyclist	$\text{mADE}_5 \downarrow$	4.47 ± 0.0	**1.34 ± 0.03**	1.40 ± 0.09
	$\text{MR}_5 \downarrow$	1.00 ± 0.0	0.71 ± 0.05	**0.65 ± 0.08**
	$\text{mADE}_{10} \downarrow$	-	1.20 ± 0.02	**1.15 ± 0.02**
	$\text{MR}_{10} \downarrow$	-	0.61 ± 0.03	**0.60 ± 0.00**
Pedestrian	$\text{mADE}_5 \downarrow$	2.61 ± 0.0	0.81 ± 0.01	**0.75 ± 0.03**
	$\text{MR}_5 \downarrow$	1.00 ± 0.0	**0.31 ± 0.04**	0.39 ± 0.05
	$\text{mADE}_{10} \downarrow$	-	0.70 ± 0.00	**0.59 ± 0.01**
	$\text{MR}_{10} \downarrow$	-	0.20 ± 0.01	**0.14 ± 0.01**
Total	$\text{mADE}_5 \downarrow$	3.36 ± 0.00	0.91 ± 0.01	**0.83 ± 0.02**
	$\text{MR}_5 \downarrow$	0.98 ± 0.00	**0.43 ± 0.03**	**0.43 ± 0.04**
	$\text{mADE}_{10} \downarrow$	-	0.77 ± 0.01	**0.67 ± 0.01**
	$\text{MR}_{10} \downarrow$	-	0.27 ± 0.01	**0.22 ± 0.01**

TABLE VIII: Numerical results of models trained and evaluated on the VoD dataset with a $T = 2$ s prediction horizon.

		CVM	PGP	PGP-CA
Vehicle	$\text{mADE}_5 \downarrow$	1.68 ± 0.0	0.25 ± 0.01	**0.22 ± 0.00**
	$\text{MR}_5 \downarrow$	0.94 ± 0.0	0.25 ± 0.01	**0.18 ± 0.02**
	$\text{mADE}_{10} \downarrow$	-	**0.18 ± 0.00**	**0.18 ± 0.00**
	$\text{MR}_{10} \downarrow$	-	**0.05 ± 0.01**	0.10 ± 0.00
Cyclist	$\text{mADE}_5 \downarrow$	2.09 ± 0.0	**0.53 ± 0.00**	0.57 ± 0.01
	$\text{MR}_5 \downarrow$	0.98 ± 0.0	**0.56 ± 0.06**	0.58 ± 0.01
	$\text{mADE}_{10} \downarrow$	-	**0.45 ± 0.01**	0.51 ± 0.00
	$\text{MR}_{10} \downarrow$	-	**0.40 ± 0.02**	0.49 ± 0.01
Pedestrian	$\text{mADE}_5 \downarrow$	0.91 ± 0.0	**0.28 ± 0.00**	**0.28 ± 0.00**
	$\text{MR}_5 \downarrow$	0.92 ± 0.0	0.27 ± 0.02	**0.26 ± 0.01**
	$\text{mADE}_{10} \downarrow$	-	**0.23 ± 0.00**	0.25 ± 0.00
	$\text{MR}_{10} \downarrow$	-	**0.14 ± 0.01**	0.20 ± 0.00
Total	$\text{mADE}_5 \downarrow$	1.27 ± 0.00	**0.31 ± 0.00**	**0.31 ± 0.00**
	$\text{MR}_5 \downarrow$	0.93 ± 0.00	0.31 ± 0.02	**0.29 ± 0.01**
	$\text{mADE}_{10} \downarrow$	-	**0.25 ± 0.00**	0.28 ± 0.00
	$\text{MR}_{10} \downarrow$	-	**0.16 ± 0.01**	0.22 ± 0.00

the paper but could provide additional insights to practitioners continuing on this work. Therefore, we list the experiments and our observations below. Please note that at the time of experimentation, we did not find the optimal hyperparameters for PGP on VoD Prediction yet, which might have influenced the results. Conclusions, based upon this list, should therefore be drawn with consideration.

- **Adjusting the sampling resolution of the road graph for dense traffic:** As mentioned in section VI, the original node sampling resolution and traversal horizon threshold of 20 and 15 meters respectively are too sparse for dense traffic. Relatively few agents move more than 20 meters in dense traffic scenarios with short prediction horizons. Thus, not many agents would traverse an edge, limiting the effect of actively using the lane graph to predict their future motion. This is due to the loss function, which is used to optimize the model [38]:

$$\mathcal{L} = \alpha \mathcal{L}_{BC} + \mathcal{L}_{reg}, \tag{3}$$

where $\alpha = 0.5$, $\mathcal{L}_{reg}$ denotes the minimum average displacement loss and $\mathcal{L}_{BC}$ is the behaviour cloning loss. $\mathcal{L}_{BC}$ is defined as the negative log-likelihood that a policy (π_{route}) traverses an edge (E) between two nodes (u, v), given that the ground truth trajectory traverses that edge:

$$\mathcal{L}_{BC} = \sum_{(u,v) \in E_{gt}} -\log\left(\pi_{\text{route}}\left(v \mid u\right)\right) \tag{4}$$

For this loss term to be effective, a target agent would need to traverse an edge. Relatively few agents move more than 20 meters in dense traffic scenarios with short prediction horizons. Thus, not many agents would traverse an edge, limiting the effect of actively using the lane graph to predict their future motion. Therefore, we experiment with higher resolution node sampling of the lane graph, such that the model actively uses the lane graph as a prior for predicting the future trajectory of all agent classes and accounts for social interactions at a finer resolution. For a two-meter sampling resolution, we observed an increase in policy loss, which is expected, as an increased amount of potential policies come available. We suggest combining this experiment with a lower weight for the policy loss term in the loss function and experimenting with a slightly increased sampling resolution (e.g. five meters).
- **Custom encoder:** Apart from adding the target agent class information to the decoder, we also experimented with a variant of the PGP model, where we added this information to the encoder. This allows for direct input of class information to the encoder and policy header, instead of relying on backpropagation to influence these stages of the model as in PGP-CA. No definite conclusions can be drawn from this experiment. We

TABLE IX: Numerical results of models pre-trained on nuScenes [13], fine-tuned and evaluated on the VoD Prediction dataset with $T = 6$ s prediction horizon. FT = Fine Tuning (Transfer Learning).

		PGP-FT	PGP-FT-CA
Vehicle	$\text{mADE}_5 \downarrow$	**0.97 ± 0.03**	1.03 ± 0.04
	$\text{MR}_5 \downarrow$	0.47 ± 0.05	**0.45 ± 0.03**
	$\text{mADE}_{10} \downarrow$	**0.67 ± 0.01**	0.75 ± 0.02
	$\text{MR}_{10} \downarrow$	**0.26 ± 0.02**	0.28 ± 0.01
Cyclist	$\text{mADE}_5 \downarrow$	1.54 ± 0.09	**1.13 ± 0.05**
	$\text{MR}_5 \downarrow$	**0.61 ± 0.05**	**0.61 ± 0.05**
	$\text{mADE}_{10} \downarrow$	1.11 ± 0.04	**0.97 ± 0.02**
	$\text{MR}_{10} \downarrow$	0.53 ± 0.07	**0.45 ± 0.05**
Pedestrian	$\text{mADE}_5 \downarrow$	0.85 ± 0.02	**0.77 ± 0.02**
	$\text{MR}_5 \downarrow$	0.37 ± 0.03	**0.33 ± 0.02**
	$\text{mADE}_{10} \downarrow$	0.69 ± 0.00	**0.65 ± 0.01**
	$\text{MR}_{10} \downarrow$	0.26 ± 0.01	**0.25 ± 0.01**
Total	$\text{mADE}_5 \downarrow$	0.97 ± 0.01	**0.89 ± 0.01**
	$\text{MR}_5 \downarrow$	0.43 ± 0.03	**0.40 ± 0.02**
	$\text{mADE}_{10} \downarrow$	0.73 ± 0.00	**0.72 ± 0.01**
	$\text{MR}_{10} \downarrow$	0.29 ± 0.01	**0.28 ± 0.01**

recommend redoing the experiment with the reported hyperparameters.

- **Custom encoder + decoder:** In this experiment, we evaluated the addition of class information both in the encoder and the decoder. This is thus a combination of the above and PGP-CA. No definite conclusions can be drawn from this experiment. We recommend redoing the experiment with the reported hyperparameters.
- **Loss function terms:** given the lower minimum average displacement errors on the VoD Prediction dataset compared to nuScenes, we tested whether a lower weight for the policy loss improves the results, given that it has a bigger influence on the total loss in the absolute sense. This experiment showed an increase in overall prediction performance for pedestrians, with the model relying less on the policy header. However, we observed a slight decrease in performance for vehicles and cyclists. This again shows that pedestrians rely to a lesser extent on the lane graph compared to vehicles and cyclists.
- **Decreased model complexity:** We also studied the network complexity by experimenting with decreased hidden layer sizes of the policy header and trajectory decoder. Given the results in the main body of the paper, we conclude that the model has sufficient data to learn meaningful patterns from data and advise sticking with the current model, given that related methods employ similar or more extensive network architectures.

REFERENCES

[1] World Health Organization. (2021) Road traffic injuries. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[2] J. Liu, X. Mao, Y. Fang, D. Zhu, and M. Q.-H. Meng, "A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2021, pp. 978–985.

[3] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[4] P. Karle, M. Geisslinger, J. Betz, and M. Lienkamp, "Scenario understanding and motion prediction for autonomous vehicles-review and comparison," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[5] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2020.

[6] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," *arXiv preprint arXiv:2207.05844*, 2022.

[7] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[8] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *Conference on Robot Learning*. PMLR, 2022, pp. 203–212.

[9] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 541–556.

[10] B. Kim, S. H. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi, "Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 636–14 645.

[11] A. Bhattacharyya, D. O. Reino, M. Fritz, and B. Schiele, "Euro-pvi: Pedestrian vehicle interactions in dense urban centers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6408–6417.

[12] A. Palffy, E. Pool, S. Baratam, J. F. Kooij, and D. M. Gavrila, "Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4961–4968, 2022.

[13] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[14] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.

[15] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning*. PMLR, 2021, pp. 895–904.

[16] E. A. Pool, J. F. Kooij, and D. M. Gavrila, "Crafted vs learned representations in predictive models—a case study on cyclist path prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 747–759, 2021.

[17] Y. Liu, Q. Yan, and A. Alahi, "Social nce: Contrastive learning of socially-aware motion representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 118–15 129.

[18] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 233–15 242.

[19] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[20] X. Mo, Y. Xing, and C. Lv, "Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction," *arXiv preprint arXiv:2106.07161*, 2021.

[21] J. F. Kooij, F. Flohr, E. A. Pool, and D. M. Gavrila, "Context-based path prediction for targets with switching dynamics," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 239–262, 2019.

[22] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Home: Heatmap output for future motion estimation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 500–507.

[23] N. Djuric, H. Cui, Z. Su, S. Wu, H. Wang, F.-C. Chou, L. San Martin, S. Feng, R. Hu, Y. Xu *et al.*, "Multixnet: Multiclass multistage multimodal motion prediction," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 435–442.

[24] F.-C. Chou, T.-H. Lin, H. Cui, V. Radosavljevic, T. Nguyen, T.-K. Huang, M. Niedoba, J. Schneider, and N. Djuric, "Predicting motion of vulnerable road users using high-definition maps and efficient convnets," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1655–1662.

[25] S. Casas, C. Gulino, S. Suo, K. Luo, R. Liao, and R. Urtasun, "Implicit latent variable model for scene-consistent motion forecasting," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 2020, pp. 624–641.

[26] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 532–539.

[27] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 2020, pp. 683–700.

[28] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Thomas: Trajectory heatmap output with learned multi-agent sampling," in *International Conference on Learning Representations*.

[29] ——, "Gohome: Graph-oriented heatmap output for future motion estimation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9107–9114.

[30] X. Gao, X. Jia, Y. Li, and H. Xiong, "Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2946–2953, 2023.

[31] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.

[32] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[33] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Conference on Robot Learning*. PMLR, 2021, pp. 409–418.

[34] K. L. El-Ashmawy, "Testing the positional accuracy of openstreetmap data for mapping applications," *Geodesy and Cartography*, vol. 42, no. 1, pp. 25–30, 2016.

[35] K. Cho, B. v. M. C. Gulcehre, D. Bahdanau, F. B. H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation."

[36] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.

[37] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.

[38] N. Deo, E. Wolff, and O. Beijbom, "Multimodal trajectory prediction conditioned on lane-graph traversals," in *Conference on Robot Learning*. PMLR, 2022, pp. 203–212.