Learning to Predict Mixed-Traffic Trajectories in Urban Scenarios from Little Training Data with Refined Environment Modeling

Alexander Prutsch[1] and Horst Possegger[1]

***Abstract*— Trajectory prediction for autonomous driving has been extensively studied using large-scale datasets from the US and Asia. These datasets typically have a strong bias toward predicting vehicle motion. Recently, the View-of-Delft Prediction (VoD-P) dataset introduced a collection of European urban mixed-traffic scenarios, posing unique challenges due to its diversity and relatively small dataset size.**

In this work, we conduct a detailed study on trajectory prediction on the VoD-P dataset. We show that state-of-the-art trajectory prediction models, which perform well on large-scale vehicle-biased datasets, struggle to generalize to the scenarios. To address this limitation, we propose a simple yet effective transformer-based trajectory prediction model, specifically designed to handle the challenges posed in diverse urban scenarios. Combining a strong baseline with refined environment modeling, our approach significantly outperforms all existing methods on the VoD-P dataset.

I. INTRODUCTION

Trajectory prediction is an integral component of an autonomous driving control stack. The goal is to predict the future movements of other traffic agents based on past observations and scene context. The predicted trajectories help an autonomous vehicle to plan its motion by anticipating the movement of other traffic agents.

Trajectory prediction for autonomous vehicles is a well-studied field. This led to numerous large-scale autonomous driving datasets, like the Argoverse 1 (AV1) [1], Argoverse 2 (AV2) [2], Waymo Open Motion Dataset (WOMD) [3] and nuScenes [4]. The majority of these datasets are recorded in North America; only nuScenes also includes data from Singapore. In addition, these datasets primarily feature vehicles, and the motion patterns of pedestrians are limited, *e.g.,* only following the walkway [5]. Thus, the traffic scenarios from these datasets significantly differ from European urban road traffic. Especially, traffic in cities with many cyclists and pedestrians poses very challenging scenarios, which are not reflected in datasets that are widely used for trajectory prediction research.

Recently, a new trajectory prediction dataset featuring European urban traffic was released to overcome this gap. The View-of-Delft Prediction (VoD-P) dataset [5] is recorded in the city of Delft in the Netherlands. It is an extension of the View-of-Delt dataset [6], which is a multi-modal (LiDAR, camera, and radar) dataset originally proposed for tasks like 3D object detection. Compared to other trajectory prediction datasets, VoD-P is rather small, as it only includes 1850 scenarios. However, it features a high number of vulnerable road users (VRUs), namely pedestrians and cyclists. Additionally, it captures complex interactions between the different types of road users.

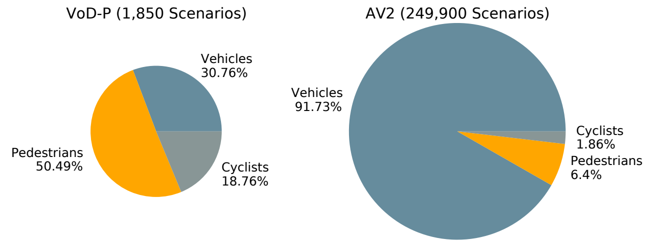

Fig. 1. Comparison of focal agent distribution across the VoD-P [5] and AV2 [2] benchmarks.

State-of-the-art trajectory prediction models [7]–[9] commonly use complex (mostly transformer-based) architectures and are designed to perform well on large-scale datasets like AV2 [2] and WOMD [3]. While achieving accurate performance on these datasets, the distinct characteristics of VoD-P mean that the approaches are not well-suited for transferring. First, VoD-P has a low number of scenarios, which puts a limit on the reasonable model size for training on the dataset. Also, transfer learning/finetuning has only a limited advantage [5] due to the large domain gap to other datasets like nuScenes [4]. Second, VoD-P features highly diverse trajectories for pedestrians and cyclists, whereas common trajectory datasets have a strong bias towards vehicle trajectories. Figure 1 compares the distribution of focal agent types in VoD-P and AV2. The large share of vehicles in AV2 leads to a strong bias towards vehicle prediction in the benchmark evaluation. For a more detailed comparison, Figure 2 shows the trajectory shapes for pedestrians and cyclists from the AV2 and VoD-P datasets. Both categories show more complex movement patterns in the VoD-P dataset.

A small, but well-capable trajectory prediction model is required for accurate results on the VoD-P dataset. Building upon strong prior work on efficient motion prediction [10], we propose a new model architecture which significantly outperforms state-of-the-art methods on VoD-P. Our approach mitigates the limitation of existing models in capturing the diverse and complex motion patterns of vulnerable road users, and leads to significant performance improvement on the challenging VoD-P dataset.

To the best of our knowledge, we are also the first to conduct a detailed study for trajectory prediction on the VoD-P dataset. Previously, only two standard models (P2T [11] and PGP [12]) were evaluated as baselines.

[1] Alexander Prutsch and Horst Possegger are with the Institute of Visual Computing, Graz University of Technology. Corresponding author: alexander.prutsch@tugraz.at

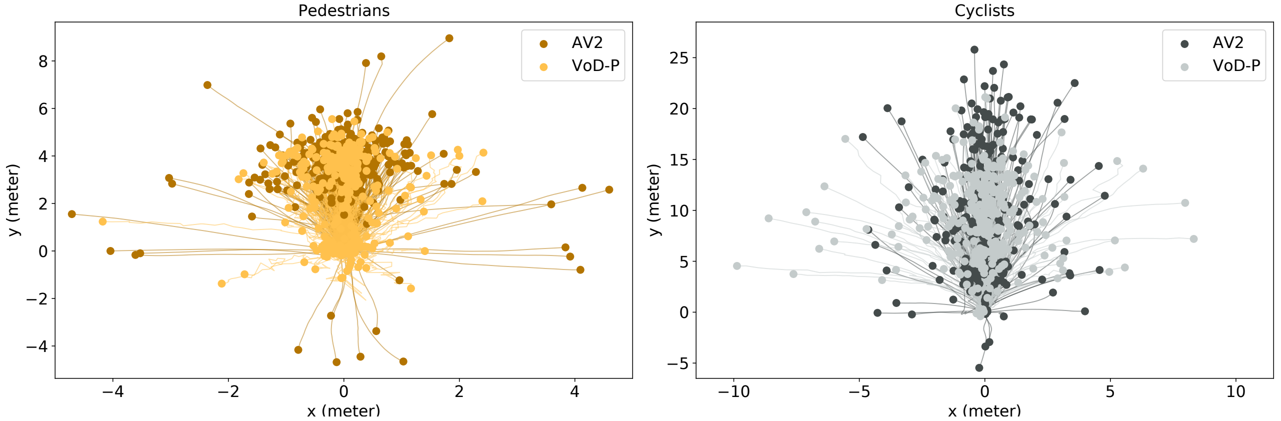

Fig. 2. Comparison of 300 randomly sampled focal agent future trajectories from each of the VoD-P [5] and AV2 [2] datasets over a 3-second time horizon. The agent's position at the current time step is set at the origin, with its heading oriented toward the positive y-direction. *Left:* Pedestrians in VoD-P exhibit highly diverse movement patterns, often wandering, whereas most pedestrians in AV2 move straight. While some trajectory endpoints in AV2 are also widely spread, the trajectories are mostly linear. This suggests they primarily result in abrupt direction changes early in the prediction horizon rather than complex walking behavior. *Right:* Cyclists in VoD-P show a more diverse trajectory distribution, whereas in AV2, most cyclists travel only in a straight line forward.

Our approach (REM) uses a simple but effective transformer-based encoder architecture and combines it with a **r**efined **e**nvironment **m**odeling. We show that our model outperforms the VoD-P baselines and related work which was designed for large-scale datasets and thus suffer from a strong vehicle bias [10], [13]. We also show that models which perform highly accurate on large-scale datasets such as Argoverse, do not necessarily perform well on small, but complex datasets like VoD-P.

In summary, our main contributions include:

- We introduce an efficient trajectory prediction model that leverages environment information to set a new state-of-the-art on the challenging VoD-P dataset, significantly surpassing all baselines.
- We conduct a comprehensive study of model components, emphasizing the critical role of input representation and environment modeling in achieving highly accurate trajectory predictions on the VoD-P dataset.
- To the best of our knowledge, we are the first to perform an in-depth analysis of trajectory prediction on the new VoD-P dataset. Additionally, we provide benchmarking results for related works [10], [13] on the VoD-P dataset.

II. RELATED WORK

A. *VoD-P Baseline Results*

The authors of the View-of-Delft Prediction (VoD-P) [5] dataset also provide trajectory prediction baseline results. They use two methods, Plans-to-Trajectories (P2T) [11] and Prediction via Graph-based Policy (PGP) [12], originally proposed for the nuScenes [4] dataset, as baselines.

Following early works in trajectory prediction [15], [16], P2T [11] employs a rasterized grid representation to model the scene context, including map elements and neighboring agents. This information is encoded using a convolutional neural network (CNN). Subsequently, maximum entropy inverse reinforcement learning (MaxENT IRL) is applied to infer possible agent paths and goals. The paths and goals are sampled to 2D plans, and an attention-based generator predicts the final trajectories using these plans and the encoded motion agent motion vector.

PGP [12] adopts a vectorized map representation, which is now also standard in state-of-the-art methods [7], [8], [10], [13], [17], [18]. Following, they process the relationship between different agents and map elements using a Graph Neural Network (GNN)-based encoder. A policy header, trained on the encoded lane graph, learns a discrete policy. Final trajectory outputs are generated based on these graph traversals. PGP [12] builds on the ideas of Plans-to-Trajectories (P2T) [11] and demonstrates superior performance on the VoD-P dataset.

B. *Efficient State-of-the-Art Trajectory Prediction*

Extensive research on trajectory prediction has led to the development of highly sophisticated trajectory prediction models [7]–[9], [17]–[20] with accurate benchmark results. All these leverage advanced model blocks, such as transformers [7], [8], [17]–[20] or state-space models [9], and incorporate well-designed processing methodologies like multi-stage refinement [7], [8] or streaming architectures [7], [20], use of raw-sensor data [18], [21] or utilize self-supervised pretraining [13].

Although these models perform well on large-scale benchmark datasets, they often encounter significant challenges when applied to custom datasets. The limitations include large model sizes that require an extensive amount training data and computational resources. Additionally, Streaming-processing methods [9], [20] rely on longer historical data, which is not available in datasets like VoD-P. Multi-stage refinement techniques [8] improve performance, but demand scene re-encoding, making them significantly harder to use in resource-constrained scenarios.

Recent research has also explored more efficient approaches to trajectory prediction. Adapt [22] introduces a simple network architecture with a resource-efficient two-stage refinement that avoids scene re-encoding. However,

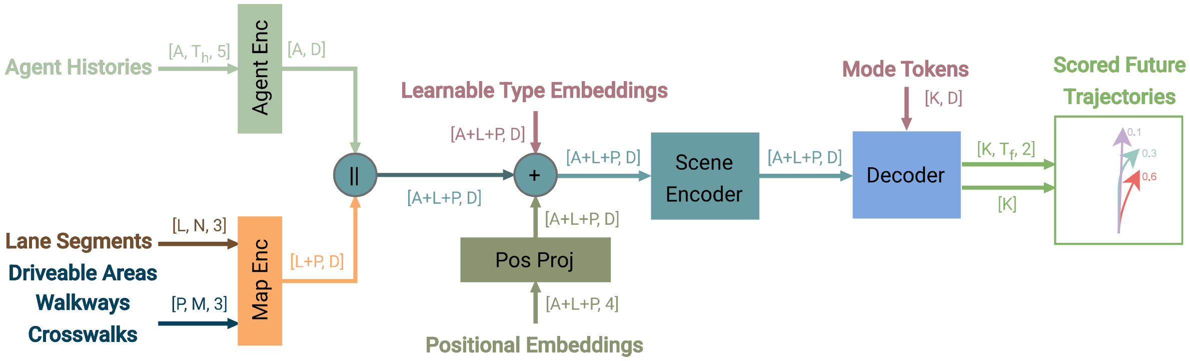

Fig. 3. Overview of our REM architecture. We process the historical agent information using temporal self-attention. Fine-grained map information (lane segments and map areas) is processed by a dedicated map encoder, which includes two Mini-PointNet-like [14] blocks. Next, we apply self-attention across all agent and map tokens to capture the relationships between the map elements. Finally, we use a MLP-based decoder to generate the trajectory output.

its evaluation is limited to the relatively simple Argoverse-1 [1] dataset. SIMPL [23] is an efficient motion prediction baseline delivering accurate results on the more complex Argoverse 2 [2] benchmark. SEPT [24] and Forecast-MAE [13] adopt simple model architectures combined with self-supervised pretraining to achieve strong trajectory prediction performance. RealMotion [20] leverages the compact Forecast-MAE model architecture while using a streaming-processing approach instead of pretraining.

Recently, EMP [10] was proposed as an efficient, yet highly accurate trajectory prediction model. It surpasses other efficient methods like SIMPL and Forecast-MAE, even without pretraining, on the Argoverse 2 [2] benchmark. The combination of a small model size with limited resource demands and the strong performance on benchmark data establishes EMP as a robust baseline for efficient trajectory prediction in diverse scenarios. However, the standard EMP model is tailored for the vehicle-biased AV2 dataset where lane information alone gives an excellent prior. As we show in our evaluations, however, it requires our proposed refined environment modeling to handle the complexity of urban scenarios to achieve strong performance on VoD-P.

III. TRAJECTORY PREDICTION WITH REFINED ENVIRONMENT MODELING

To achieve highly precise trajectory prediction on the View-of-Delft Prediction dataset (VoD-P) [5], we propose an extended version of EMP [10]. While maintaining the same simplistic design, our model features **r**efined **e**nvironment **m**odeling (REM), which is particularly important for trajectory prediction in complex urban scenarios: Our refined map encoder provides a significant performance gain with only a slight increase in model size, an important advantage given the limited training dataset size.

A. Input Data Representation

The VoD-P [5] dataset adopts the same structure as nuScenes [4]. It includes map annotations for the city of Delft, Netherlands, containing polygon annotations (*e.g.,* walkways, crosswalks, and drivable areas) and polylines representing lanes. Additionally, agent annotations are provided as pose trajectories with timestamps. To prepare the raw dataset for our model, we employ a custom preprocessing pipeline to generate structured tensors for map and agent data.

EMP [10] uses lane and map information as input. The input representation follows the design of [13], which is also adapted by state-of-the-art methods [9], [20] on the Argoverse 2 [2] dataset. All lane elements are represented in a tensor $\mathbb{R}^{L\times N\times 3}$, where L is the number of lane segments and N is the number of sampled points per segment. Each point is represented by a 3D feature vector containing the (x, y) coordinates and a padding flag. To limit input to nearby context, we define a circular region of interest and mask all points outside this radius using the padding flag.

We propose the addition of a new map input tensor containing polygon-shaped map elements $\mathbb{R}^{P\times M\times 3}$, where P denotes the number of polygon area elements and M is the number of sampled points. Each feature vector contains the (x, y) coordinates and a padding flag.

As noted in previous works [18], [24], higher sampling frequencies N and M allow models to capture fine-grained map details but also increase the computational complexity of the map encoder. Thus, N and M are hyperparameters that balance model accuracy and efficiency.

The agent history is captured in a state tensor $\mathbb{R}^{A\times T_h\times 5}$, where A is the number of agents in the scene and T_h is the history length (number of historic time steps). We represent each agent state using the relative (x, y) displacement between consecutive time steps and the velocity changes in x and y direction. Additionally, we use a padding flag to indicate whether an agent is observed at a given time step.

B. Model Architecture

We use EMP [10] as baseline for our trajectory prediction model. EMP features a simple yet effective transformer-based architecture, comprising agent and lane encoders, a scene encoder and a trajectory decoder. To adapt the model to the challenges of the VoD-P dataset, we enhance the map information processing by an additional map encoder. Figure 3 shows an overview of our REM architecture.

1) Agent Encoding: The agent histories, represented as $\mathbb{R}^{A\times T_h\times 5}$, are encoded using self-attention across the temporal dimension [24] using standard multi-head transformer blocks. To enhance computational efficiency, we apply a max-pooling operation along the temporal dimension [24] to reduce the dimensionality. This yields feature tokens $\mathbb{R}^{A\times D}$, where each token captures the movement pattern of an individual agent.

2) Map Encoding: Lane information $\mathbb{R}^{L\times N\times 3}$ is processed using a Mini-PointNet-like [14] encoder. Before encoding, the lanes are normalized to a local coordinate system by subtracting the polyline center from their global coordinates.

Lanes provide a strong prior for vehicle trajectories in well-structured scenarios like following a (multi-lane) road in an American city. Thus, many state-of-the-art methods [9], [10], [13], [20], [24], which target large-scale datasets like Argoverse-2 [2] with a particular focus on vehicle trajectories, rely solely on lane information as map input. Other map elements, such as drivable or walkable areas, are omitted in these approaches.

While lanes are often sufficient for predicting vehicle trajectories, the trajectories of cyclists and pedestrians are highly dynamic and do not strictly follow lane structures. They are also significantly influenced by environmental elements such as walkway boundaries and crosswalks. Additionally, these elements also have an impact on vehicle movements in complex scenarios, *i.e.,* where numerous other traffic agents are in a crossing.

VoD-P includes polygon annotations for structural elements such as crosswalks and walkways. To leverage this additional information, we extend the map encoder in our model to process polygon-shapes map elements alongside lane segments. We introduce a second PointNet-like encoder block to process polygon information $\mathbb{R}^{P\times M\times 3}$ into area tokens $\mathbb{R}^{P\times D}$. As with lanes, we transform all polygon elements to local coordinates by subtracting the mean coordinate from each shape. The additional encoder ensures our model captures diverse environmental cues.

3) Scene Encoding: We combine agent and map information by stacking agents, line and area tokens from the scene representation $\mathbb{R}^{A+L+P\times D}$ [10], [13]. Then, we add learnable type embeddings that encode categorical information such as agent type (vehicle, cyclist, or pedestrian) and lane type. They also allow different scene token types to be integrated into a unified space [13].

Additionally, we incorporate positional encodings to model the global relationships between scene elements. Both agent and map encoding are performed in local coordinates to facilitate more efficient learning. The global positional encodings capture the relationship between these local coordinate systems with the origin of the global coordinate system set at the position of the focal agent at the current time step ($t = 0$).

To represent the transformations, we use the (x, y) position and the rotation of each local coordinate system encoded using sine and cosine functions. The position features are encoded using a shallow MLP-based network to obtain our position embeddings $\mathbb{R}^{A+L+P\times D}$.

To model the interactions and spatial relationships between all scene elements, we apply self-attention using multi-head transformer blocks. Our scene encoder yields a final encoded scene representation $\mathbb{R}^{A+L+P\times D}$.

4) Decoding: EMP [10] uses two different trajectory decoder: a simple multi-layer perceptron (MLP)-based decoder (EMP-M) and a transformer-based decoder (EMP-D). EMP-D significantly outperforms EMP-M on the large-scale dataset Argoverse 2 [2]. However, our experiments show that on the smaller VoD-P dataset, the simpler MLP-based decoder actually yields better performance. Based on this observation, we also use a MLP-based decoder [10] in our model designed for the VoD-P dataset.

To obtain K hypothesis for the future trajectory of a focal agent, we broadcast the encoded focal agent token $\mathbb{R}^{1\times D}$ to $\mathbb{R}^{K\times D}$, and add learnable agent mode tokens $\mathbb{R}^{K\times D}$ to generate a multi-modal output. Next, we apply two shallow MLPs to map the focal agent tokens $\mathbb{R}^{K\times D}$ to future trajectories $\mathbb{R}^{K\times T_f\times 2}$ and their associated scores $\mathbb{R}^{K}$.

IV. EXPERIMENTAL SETUP

A. Data Preprocessing

To generate input scenarios, we sample map elements and neighboring agents within a minimum distance of 30 meters (L^2 norm) from the origin of our global coordinate system (focal agent's position at time $t = 0$). In our main results, we sample $N = 40$ points for each lane segment, and $M = 80$ points for each polygon area. Additionally, we provide an ablation study for different input resolutions. We distinguish between two lane types: *standard lane segments* and *intersections*. For the polygon areas, we differentiate between three types: *drivable areas*, *walkways* and *crosswalks*.

Following the evaluation protocol [5], we use a historic horizon of $T_h = 0.5$s and predict trajectories for $T_f = 3$s into the future. The frame rate for the agent data is 10 Hz. We use the data split proposed by the official VoD-P benchmark [5] (1396 train sequences 454 sequences for evaluation). We do not incorporate data from other sources for training, and neither perform pre-training nor data augmentation.

B. Implementation Details

In our main results, we adopt an embedding dimension of 128 for our models, with 4 transformer blocks in both the agent and scene encoders. These settings align with those of the comparison models, EMP [10] and FMAE [13]. Additionally, we provide results for smaller models in our ablation study using an embedding size of 64 and shallower encoders. These reductions lower inference latency, making the models even more suitable for practical applications.

We train our model for 25 to 35 epochs, varying depending on the model's size. We use AdamW as optimizer with a learning rate schedule that includes warm-up. During the first 10 epochs, the learning rate is linearly increased from 0.0001 to 0.001. Then the learning rate is reduced to 0.0001 using a cosine decay schedule. For better optimization, we apply gradient clipping and apply weight decay regularization.

Following the standard state-of-the-art approaches [9], [10], [13], [20], we apply a winner-takes-all strategy for

TABLE I

TRAJECTORY PREDICTION RESULTS ON THE VOD-P TEST SET. WE SET THE NUMBER OF MODEL OUTPUTS TO $K = 10$ IN OUR EXPERIMENTS. FOR ALL METRICS LOWER VALUES INDICATE BETTER RESULTS. SORTED BY MINFDE$_{10}$ IN DESCENDING ORDER. **BOLD** MARKS BEST VALUE, UNDERLINE MARKS SECOND BEST VALUE.

	All Agents								
Method	MR$_1$	minADE$_1$	minFDE$_1$	MR$_5$	minADE$_5$	minFDE$_5$	MR$_{10}$	minADE$_{10}$	minFDE$_{10}$
EMP-D [10]	**0.833**	1.461	2.904	0.786	1.096	2.185	0.747	0.694	1.362
Forecast-MAE [13]	0.852	1.199	2.345	0.758	0.548	1.007	0.758	0.476	0.833
PGP [12]	0.86	0.89	1.81	0.66	0.57	1.06	0.56	0.46	0.79
EMP-M [10]	0.868	**0.710**	**1.466**	0.661	0.498	0.989	0.586	0.404	0.764
REM (Ours)	0.921	0.951	1.892	**0.654**	**0.423**	**0.792**	**0.564**	**0.356**	**0.643**

TABLE II

RESULTS ON THE VOD-P DATASET BROKEN DOWN BY FOCAL AGENT TYPES. VALUES ARE REPORTED FOR THE SAME MODELS AS IN TABLE I. **BOLD** MARKS BEST VALUE, UNDERLINE MARKS SECOND BEST VALUE.

	Pedestrian			Cyclist			Vehicles		
Method	MR$_{10}$	minADE$_{10}$	minFDE$_{10}$	MR$_{10}$	minADE$_{10}$	minFDE$_{10}$	MR$_{10}$	minADE$_{10}$	minFDE$_{10}$
EMP-D [10]	0.638	0.342	0.571	0.930	0.724	1.489	0.820	1.298	2.680
Forecast-MAE [13]	0.613	0.340	0.569	0.977	0.746	1.459	0.872	0.540	0.896
PGP [12]	0.43	0.27	**0.38**	0.90	**0.53**	**1.05**	**0.57**	0.76	1.35
EMP-M [10]	0.460	0.256	0.425	**0.779**	0.564	1.132	0.684	0.563	1.125
REM (Ours)	**0.404**	**0.251**	0.401	0.791	0.564	1.153	0.699	**0.407**	**0.741**

TABLE III

TRAJECTORY PREDICTION RESULTS ON THE VOD-P TEST SET. WE SET THE NUMBER OF MODEL OUTPUTS TO $K = 5$. SORTED BY MINFDE$_5$ IN DESCENDING ORDER.

	All Agents		
Method	MR$_5$	minADE$_5$	minFDE$_5$
Forecast-MAE [13]	0.797	0.625	1.119
EMP-D [10]	0.714	0.563	1.063
EMP-M [10]	**0.659**	0.488	0.929
REM (Ours)	0.663	**0.452**	**0.862**

model optimization. First, we select the best-fitting trajectory from the K predicted modes based on the L^2-norm deviation from the ground truth. Then, we compute a regression loss (Huber loss [25]) for the selected trajectory and a classification loss (cross-entropy loss) for predicting the highest probability score for the best-fitting mode.

Our final loss also includes an auxiliary regression loss for predicting a single future trajectory for neighboring agents in the scene [9], [10], [13], [20]. To achieve this, we process the agent tokens $\mathbb{R}^{A-1\times D}$ for all neighboring agents using a two-layer MLP to generate single mode trajectories $\mathbb{R}^{A-1\times T_f\times 2}$.

C. Evaluation Metrics

Following the Vod-P [5] and nuScenes [4] evaluation protocols, we evaluate the minimum average displacement error (minADE$_k$), minimum final displacement error (minFDE$_k$) and the miss rate (MR$_k$) for $k = \{1, 5, 10\}$. The k highest-scoring trajectory predictions from the model are considered for each respective metric. We report both overall results and results for individual agent classes (*pedestrians*, *cyclists* and *vehicles*).

V. RESULTS AND DISCUSSION

We present our main results for trajectory prediction on the View-of-Delft Prediction (VoD-P) [5] dataset in Table I. Our approach (REM) is compared to PGP [12], which is the strongest baseline provided by the dataset team[1]. Additionally, we evaluate two recent trajectory prediction models: Forecast-MAE [13] and EMP [10], both of which have demonstrated strong performance on large-scale vehicle-focused datasets such as AV1 [1] and AV2 [2]. To ensure a fair comparison, we employ the official implementations of these methods and adhere to their data preprocessing procedures. Furthermore, we use a uniform lane segment sampling rate of $N = 40$ across all methods and omit the self-supervised pretraining step for Forecast-MAE due to the small dataset size.

Overall, REM achieves the best prediction results across the evaluated metrics. It demonstrates a significant improvement evaluating the top-5 and top-10 trajectory hypotheses. Specifically, compared to the PGP baseline [5], our model improves the minADE$_{10}$ by approximately 23% and the minFDE$_{10}$ by around 19%. Against the second-best performing model, EMP-M [10], our approach achieves a 12% improvement in minADE$_{10}$ and 16% in minFDE$_{10}$. In Figure 4, we provide qualitative results of our approach for scenarios from the VoD-P dataset.

While EMP-M produces the most accurate top-1 predictions, this is likely due to the small dataset size relative to the high number of potential output modes. In contrast, our model is significantly more effective at leveraging multiple hypotheses, providing superior performance for broader trajectory evaluation scenarios. Notably, EMP-D, the model demonstrating the best performance on the large-scale Argoverse 2 dataset, yields the poorest results in our evaluation on VoD-P. This emphasizes the importance of adapting model designs to account for differing dataset characteristics.

Table II presents a breakdown of the results from Table I by focal agent class. REM outperforms the other models for predicting pedestrians and vehicle trajectories. For cyclists, our results are slightly worse than PGP and EMP-M, but the

[1] Results obtained from the official leaderboard: `https://eval.ai/web/challenges/challenge-page/2410/leaderboard`

TABLE IV

ABLATION STUDY ON DIFFERENT NUMBER OF SAMPLING POINTS PER MAP ELEMENT. N AND M DENOTE THE NUMBER OF POINTS SAMPLED FROM EACH LANE SEGMENT/MAP POLYGON. FOR ALL EXPERIMENTS, WE USE REM WITH $K = 10$ MODE OUTPUTS.

Input Sampling		All Agents			
N	M	minADE$_5$	minFDE$_5$	minADE$_{10}$	minFDE$_{10}$
20	20	0.467	0.933	0.373	0.704
20	40	0.462	0.916	0.388	0.715
20	80	0.464	0.908	0.367	0.677
40	40	0.474	0.879	0.397	0.696
40	80	**0.423**	**0.792**	**0.356**	**0.643**
80	80	0.425	0.826	0.361	0.663

TABLE V

ABLATION STUDY ON DIFFERENT MODEL SIZES FOR REM ($K = 10$ MODES). WE VARY THE NUMBER OF TRANSFORMER BLOCKS IN OUR ENCODER AND TEST A SMALLER MODEL DIMENSIONALITY $D = 64$.

	Enc. Blocks		All Agents				Params
D	Agent	Scene	minADE$_5$	minFDE$_5$	minADE$_{10}$	minFDE$_{10}$	#
64	2	2	0.424	0.837	0.371	0.687	673K
128	2	2	0.455	0.863	0.381	0.676	1.4M
128	4	4	**0.423**	**0.792**	**0.356**	**0.643**	2.2M

difference is minimal within just a few centimeters.

For the results in Table I we set the number of output trajectory modes in our experiments to $K = 10$. Although the VoD-P benchmark allows up to 25 trajectories for evaluation, only the top-5 or top-10 scoring trajectories are considered in the official evaluation protocol. Increasing the number of hypotheses does not necessarily improve performance, as accurately ranking the probability scores becomes increasingly challenging with more trajectories. Additionally, increasing the number of output trajectories makes training more challenging, as commonly only the best fitting trajectory is optimized for each scenario.

To illustrate this, we provide additional results in Table III, where K is set to 5. Again, our approach yields the best results overall. Interestingly, EMP-D performs significantly better with $K = 5$ output modes compared to $K = 10$. This can be attributed to its larger model size, which requires more training data to effectively optimize the individual mode queries. The limited training data in VoD-P is insufficient to fully leverage EMP-D's capacity when using $K = 10$.

A. Ablation Studies

In Table IV we present an ablation study examining different sampling steps for the map input. The results demonstrate that the environment sampling step plays a crucial role for achieving highly accurate trajectory predictions. We achieve the best results using $N = 40$ for sampling lane segments and $M = 80$ for sampling map polygons. Notably, even with lower sampling rates of $N = 20$ and $M = 20$, our model outperforms competing approaches (see Table I). Furthermore, increasing the input sampling beyond these values does not yield further improvements, as the additional points do not lead to an information gain.

In Table V, we evaluate different encoder depths and network sizes. The results indicate that using shallower encoder modules while maintaining an embedding size of $D = 128$ leads to only a marginal performance loss, with our reduced model still outperforming all competitors. Even when lowering the embedding size from $D = 128$ to $D = 64$, the shallower encoder achieves excellent results. The minFDE$_{10}$ is only 7% worse than our full model while requiring less than a third of the parameters. All tested configurations outperform the current state-of-the-art baseline on VoD-P (PGP [11]).

TABLE VI

COMPARISON OF REM TO OUR BASELINE (EMP-M [10]) ON THE ARGOVERSE 2 [2] VALIDATION SET.

	All Agents		
Method	MR$_6$	minADE$_6$	minFDE$_6$
EMP-M	**0.191**	0.730	1.457
REM (Ours)	**0.191**	**0.724**	**1.448**

We also compare our approach to EMP-M [10] on the large-scale AV2 [2] dataset. We follow the parameter settings of EMP-M for training, and use the default AV2 evaluation protocol. In addition to polyline annotations (lanes and crosswalks), AV2 provides only the driveable area as map polygons. Since these polygons are significantly larger than those in VoD-P, we use a higher sampling rate ($M = 160$). REM achieves better results than EMP-M. However, due to the vehicle-centric nature of the dataset, where lanes already provide a strong prior, the performance gain from incorporating the driveable area is limited.

VI. CONCLUSIONS

We propose a transformer-based trajectory prediction approach achieving state-of-the-art results on the View-of-Delft prediction dataset. Our model design is specifically tailored to the challenges of having a rather small dataset with highly diverse agent movements. Our experiments show that state-of-the-art methods for large-scale datasets like AV2 do not naturally perform well on specialized dataset. However, by adapting the environment processing while keeping a compact architecture, we achieve excellent results, outperforming all baselines and other methods.

For future work, [18], [21] indicates advantages for predicting pedestrian and cyclists behavior by incorporating raw sensor data in addition to static map data. However, this leads to an increased model size which makes integration difficult to due the small training dataset size.

REFERENCES

[1] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3D Tracking and Forecasting with Rich Maps," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting," in *Proc. of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[3] 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 *Proc. of the IEEE/CVF Conference on Computer Vision (ICCV)*, 2021.

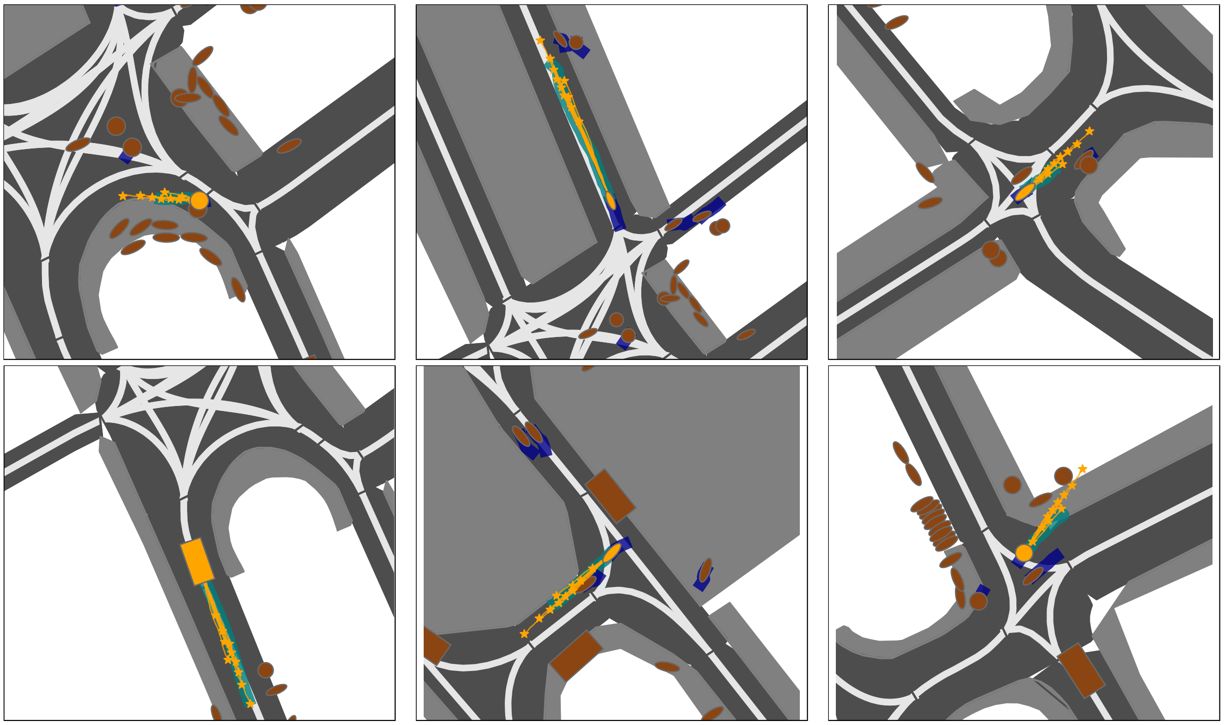

Fig. 4. Exemplary results for our method on the VoD-P dataset. Visualizations show **focal agent**, **predictions**, **ground truth**, **agent histories**, **neighboring agents** and map elements. Vehicles are shown as squares, cyclists as ellipses and pedestrian as circles. *Top row (from left to right):* pedestrian enters walkway, cyclist follows a driving lane, and cyclist passing through an intersection. *Bottom row (from left to right)*: vehicle exits an intersection, cyclist crosses the street continuing into a driving lane, and pedestrian crosses the streets and enters the walkway.

[4] 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 *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] H. J. Boekema, B. K. Martens, J. F. Kooij, and D. M. Gavrila, "Multi-Class Trajectory Prediction in Urban Traffic Using the View-of-Delft Prediction Dataset," *IEEE Robotics and Automation Letters (RAL)*, 2024.

[6] Palffy, Andras and Pool, Ewoud and Baratam, Srimannarayana and Kooij, Julian F. P. and Gavrila, Dariu M., "Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset," *IEEE Robotics and Automation Letters (RAL)*, 2022.

[7] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-Centric Trajectory Prediction," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] Y. Zhou, H. Shao, L. Wang, S. L. Waslander, H. Li, and Y. Liu, "SmartRefine: A Scenario-Adaptive Refinement Framework for Efficient Motion Prediction," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[9] B. Zhang, N. Song, and L. Zhang, "DeMo: Decoupling Motion Forecasting into Directional Intentions and Dynamic States," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[10] A. Prutsch, H. Bischof, and H. Possegger, "Efficient Motion Prediction: A Lightweight & Accurate Trajectory Prediction Model With Fast Training and Inference Speed," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[11] N. Deo and M. M. Trivedi, "Trajectory Forecasts in Unknown Environments Conditioned on Grid-Based Plans," *arXiv preprint arXiv:2001.00735*, 2020.

[12] N. Deo, E. Wolff, and O. Beijbom, "Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals," in *Proc. of the Conference on Robot Learning (CoRL)*, 2022.

[13] J. Cheng, X. Mei, and M. Liu, "Forecast-MAE: Self-supervised Pre-training for Motion Forecasting with Masked Autoencoders," in *Proc. of the IEEE/CVF Conference on Computer Vision (ICCV)*, 2023.

[14] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "CoverNet: Multimodal Behavior Prediction Using Trajectory Sets," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction," in *Proc. of the Conference on Robot Learning (CoRL)*, 2020.

[17] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion Transformer with Global Intention Localization and Local Movement Refinement," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[18] Y. Gan, H. Xiao, Y. Zhao, E. Zhang, Z. Huang, X. Ye, and L. Ge, "MGTR: Multi-Granular Transformer for Motion Prediction with LiDAR," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2024.

[19] X. Wang, T. Su, F. Da, and X. Yang, "ProphNet: Efficient Agent-Centric Motion Forecasting with Anchor-Informed Proposals," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[20] N. Song, B. Zhang, X. Zhu, and L. Zhang, "Motion Forecasting in Continuous Driving," in *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[21] K. Chen, R. Ge, H. Qiu, R. Ai-Rfou, C. R. Qi, X. Zhou, Z. Yang, S. Ettinger, P. Sun, Z. Leng, *et al.*, "WOMD-LiDAR: Raw Sensor Dataset Benchmark for Motion Forecasting," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2024.

[22] G. Aydemir, A. K. Akan, and F. Güney, "ADAPT: Efficient Multi-Agent Trajectory Prediction with Adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[23] L. Zhang, P. Li, S. Liu, and S. Shen, "SIMPL: A Simple and Efficient Multi-agent Motion Prediction Baseline for Autonomous Driving," *IEEE Robotics and Automation Letters (RAL)*, 2024.

[24] Z. Lan, Y. Jiang, Y. Mu, C. Chen, S. E. Li, H. Zhao, and K. Li, "SEPT: Towards Efficient Scene Representation Learning for Motion Prediction," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2023.

[25] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.