Lanes Are Not Enough: Enhancing Trajectory Prediction in Intralogistics Through Detailed Environmental Context

Alexander Prutsch[1], Matthias Wess[2,3] and Horst Possegger[1,3]

***Abstract*— Trajectory prediction is an essential component of the perception stack in autonomous mobile robots (AMRs). AMRs operate in complex environments where their movements are influenced by various environment elements, such as racks and storage locations. Therefore, accurate and efficient trajectory prediction for intralogistics requires detailed environment modeling that goes beyond the lane-based context mainly used in road traffic methods. We propose the addition of a new environment context encoder module that can be seamlessly integrated into state-of-the-art autonomous driving systems. Our approach, tailored to the specific challenges of intralogistics, achieves highly accurate predictions using compact and efficient baseline networks.**

I. INTRODUCTION

Autonomous vehicles play a major role in modern intralogistics. Autonomous mobile robots (AMRs) are commonly used to transport cargo in complex environments like warehouses and production facilities. To achieve efficient behavior by AMRs in crowded traffic scenarios, they must comprehend the behavior of other traffic participants.

Trajectory prediction is an important task to provide autonomous vehicles an understanding of traffic scenes. Predicting the future movements of other agents helps in planning driving maneuvers, since potential collisions and dangerous situations can be mitigated. For example, vehicle deadlocks can cause operational delays, leading to significant costs. Additionally, a lack of proactive driving in situations involving manual traffic can pose risks to warehouse workers. Compared to other technologies, *e.g.,* intra-vehicle communication, visual perception-based approaches have the advantage that they do not rely on the technology and equipment of other vehicles. Hence, manual vehicles can be easily taken into account, and there are no limitations due to vendor restrictions.

For road traffic, trajectory prediction is a well-studied field, *e.g.,* [1], [2], [3], [4], [5], [6], [7]. However, the characteristics of intralogistics vehicles and their operating environments present special challenges. Compared to road users, intralogistics vehicles can execute more diverse and complex maneuvers due to their wheel geometry [8]. Additionally, lane graphs, which are commonly used to model environmental information in trajectory prediction models, offer only weak guidance in the intralogistics domain. In warehouse traffic, vehicles less strictly follow driving lanes, as they might choose to use shortcuts. Moreover, warehouse environment elements generally have a strong influence on potential driving maneuvers. As an example, if a vehicle moves inside a rack aisle, there is a high chance of an abrupt change of driving speed in case the vehicle initiates a load handling operation. Hence, it is essential to model these map elements for accurate trajectory prediction in intralogistics.

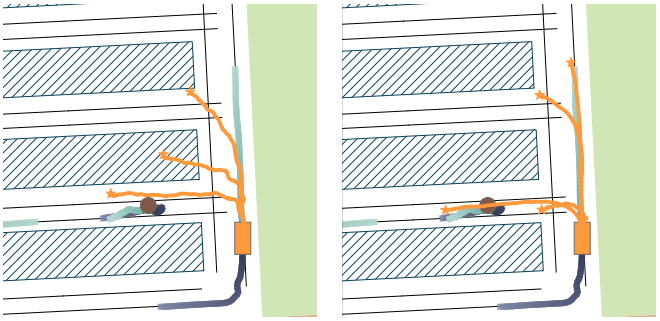

Fig. 1. Trajectory prediction scenario from our intralogistics scenario. On the left side we show the **output** from a trajectory prediction model [9] designed for autonomous driving which is trained on our data. The model produces multi-modal outputs, but the predictions are unfeasible since they cut trough **racks**. On the right side we show the predictions using our new ECTX encoder. The predictions perfectly align with the given rack aisles.

Due to different environment types, vehicle characteristics, and available onboard resources, trajectory prediction models designed for autonomous cars cannot be directly transferred to the AMR domain. We propose a new environment encoder module to enable accurate trajectory prediction in intralogistics. Our **e**nvironment **c**on**t**e**x**t (ECTX) encoder processes information on diverse map elements like rack positions, charging stations and gates. It enables highly accurate predictions in scenarios where lanes only offer a weak prior. Figure 1 shows an exemplary scenario from an intralogistics dataset highlighting that adding ECTX (right image) yields significantly improved trajectory predictions since it also takes the rack locations (blue areas) into account. As mentioned above, intralogistics vehicles may take shortcuts and do not strictly follow designated lanes. Therefore, relying solely on a lane encoder for environmental context, as in [9], prevents the model from learning which corners can be cut and which ones would lead to collisions.

We choose two strong baselines with compact transformer-based architectures, Forecast-MAE [9] and EMP [10], and integrate our ECTX module to these architectures. State-of-the-art multi-stage models like QCNet [3] and SmartRefine [7], or large models like SEPT [11], are unfavorable for our problem domain since we require resource-efficient

[1] Affiliated with the Institute of Visual Computing, Graz University of Technology. Corresponding author: alexander.prutsch@tugraz.at

[2] Affiliated with the Institute of Computer Technology, Vienna University of Technology.

[3] Affiliated with with the Christian Doppler Laboratory for Embedded Machine Learning.

methods. We show that for both baselines, the integration of our ECTX encoder leads to improved results for trajectory prediction in complex intralogistics driving scenarios. The additional environment information significantly improves the trajectory prediction accuracy while adding only a little overhead to the networks.

Our ECTX encoder directly uses semantic warehouse maps as input, which are generally available in practice for AMR deployment scenarios. Compared to recent autonomous driving trajectory prediction models [6], [5], which utilize LiDAR data for detailed environmental context, our model is more efficient. Another advantage is that it does not require LiDAR data for training. For our use case, the collection of LiDAR data is challenging due to data privacy restrictions. Hence, our trajectory model is well suited for intralogistics solutions, since finetuning on application-specific sensor data is not required.

In our experiments, we demonstrate the effectiveness of our ECTX. We evaluate it on a custom large-scale dataset for motion prediction of different intralogistics vehicles, *e.g., reach trucks* and *order pickers*. In our evaluations we shot that our approach performs well on long-term prediction horizons. This gives AMRs sufficient time to react to the prediction and to use the output to perform smooth, proactive driving maneuvers.

In summary, our main contributions include:

- We study the problem of trajectory prediction in intralogistics and highlight that lane topology context is insufficient for these use cases.
- We propose a new environment context (ECTX) encoder for trajectory prediction models to obtain a detailed environment understanding beyond lane topology.
- We show that adding ECTX to different baseline models significantly improves the performance for prediction of vehicle trajectories in intralogistics scenarios and that our adapted architecture enables small models to outperform large-scale models designed for autonomous driving in public road scenarios.

II. RELATED WORK

A. Trajectory Prediction Models for Autonomous Vehicles

State-of-the-art trajectory prediction methods are primarily transformer-based and combine multiple encoder modules with a trajectory decoder. Separate encoders are commonly utilized to learn movement information from past agent data and environment information from static map data. Following, relationships between these cues are learned using either type-specific attention operations [12], [13], [3], *e.g.,* agent-to-map and agent-to-agent, or general self-attention on all scene tokens [2], [9], [11], [5], [10], [14], [15]. Shi *et al*. [2] proposed a Motion Transformer (MTR) utilizing a local context encoder to learn scene information. Following, a DETR-like [16] decoder is used to generate trajectory predictions. MGTR [5] advances the MTR framework by introducing a multi-granular scene input encoding and adds an additional LiDAR encoder. Currently, MGTR yields state-of-the-art performance on the Waymo Open Motion Dataset [17].

QCNet [3] proposes a query-centric approach and learns spatiotemporal-invariant scene encodings. Additionally, they propose a two-stage trajectory decoding: anchor free proposals are generated at first, and then they are used as anchors to generate refined trajectories. SmartRefine [7] advances this idea by using anchor-centric coordinate systems to better capture local information about the target regions.

Our focus on AMRs requires a particular emphasis on the computational complexity of trajectory prediction models, since only limited onboard resources are available. Forecast-MAE [9] has a compact architecture that achieves good accuracy with good inference speed by using a pretraining scheme. Also, SEPT [11] combines a simple model architecture with a sophisticated pretraining. SEPT is highly accurate on the Argoverse 2 (AV2) [18] benchmark and outperforms, *e.g.,* [3], [7] despite both having a more complex architecture. EMP [10] uses a simple transformer-based model that achieves high precision and fast inference speed. The paper highlights that a compact model can achieve competitive results on autonomous driving datasets like AV2.

Given our use case, we consider modified versions of Forecast-MAE and EMP as well-suited baselines for our approach. They offer a good performance/resource trade-off and do not utilize type-specific attention in the scene encoder, which leads to easier integration of our environment tokens. Further examples of recent resource-efficient trajectory prediction models would be, *e.g.,* SIMPL [19] and ADAPT [13]. We do not consider these methods, since both yield inferior results on benchmark datasets compared to Forecast-MAE and EMP.

B. Environment Modeling for Trajectory Prediction

The past focal agent movement, the movement of other adjacent traffic participants, and the scene environment are essential for predicting the future trajectory of traffic agents. Using only the agent movement, *e.g.,* [20], yields inferior results as, *e.g.,* future turn movements that follow a road bend cannot be modeled accurately.

Generally, the environmental context, including elements like driving lanes and non-driveable areas, can be effectively modeled using static maps. To this end, the usage of rasterized maps was initially proposed, *i.e.,* [21], [22]. In this representation, each map cell corresponds to a semantic context class. The map resolution introduces a significant trade-off between model efficiency and accuracy.

To mitigate this trade-off and to achieve lower latency, the use of vectorized map data has become state-of-the-art in recent years. Using vectorized maps, each map element is represented by polylines. Each polyline is associated with a specific map element type, *i.e.,* a one-way lane segment. Utilizing vectorized map data, models like [2], [3], [23], [24] achieve highly accurate results on autonomous driving datasets like Argoverse 1 [18], Argoverse 2 [25] and the Waymo Open Motion dataset [17].

Lanes provide clear guidance for the movement of road users, *i.e.,* cars, motorcyclists, trucks, and buses, but other influencing environmental factors, which are not captured

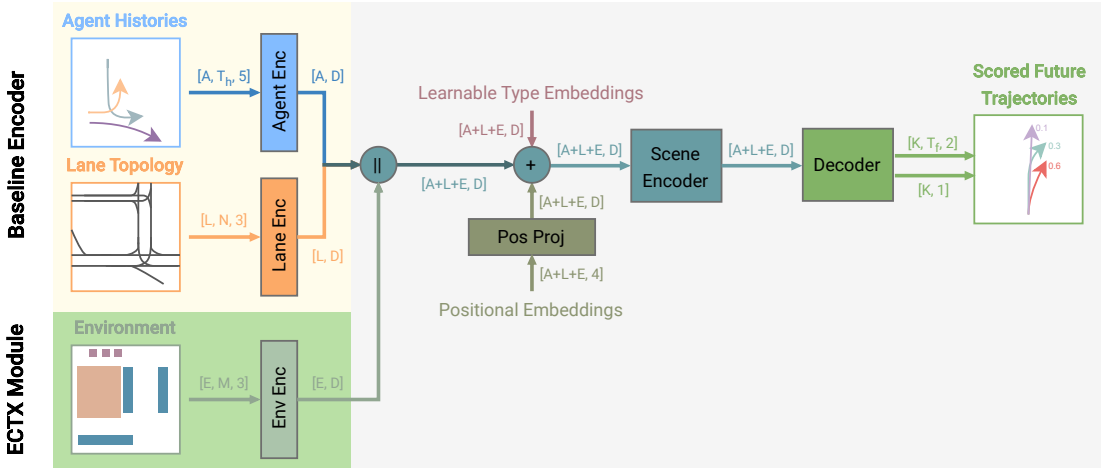

Fig. 2. Commonly, trajectory prediction models utilize an agent and a lane encoder (top left). Tokens from both encoder are then combined to a scene matrix which is further encoded and then used in the trajectory decoder. Both baselines, Forecast-MAE [9] and EMP [10], follow this architecture. Hence, our ECTX encoder can be used as a simple plug-in to these baselines by adding the environment tokens to the scene matrix. || denotes concatenation.

in static map data, also play a significant role. To increase prediction accuracy, Wayformer [24] proposes the integration of traffic light states as an additional input to the scene encoding. Static map data also commonly lacks information on temporary scene contexts, *e.g.,* road cones, or fine-grained information on structures like vegetation. Recently, MGTR [5] and Wayformer+LiDAR [6] propose integrating LiDAR data directly into trajectory prediction models using an additional LiDAR encoder. LiDAR scans give a detailed description of the current environment. They demonstrate that the additional LiDAR data is especially beneficial for predicting pedestrian and cyclist movement in road scenes, as their movement is heavily influenced by temporary objects.

C. Autonomous Driving in Warehouses

Autonomous mobile robots (AMRs) are commonly used in modern warehouses and production facilities. They often operate in dense traffic environments with both autonomous and manually operated vehicles. AMRs are equipped with various sensors, like 2D/3D LiDAR scanners and cameras, which enable them to capture their surroundings. In order to enable efficient driving in such scenarios, a detailed understanding of the current traffic scene is essential. Detecting the actions of other traffic agents [8] and the prediction of future vehicle movements are essential for smooth AMR traffic. Understanding the behavior of other traffic agents helps AMRs dynamically react to their environment and execute proactive driving.

Compared to autonomous cars, AMRs have significantly fewer resources. Additionally, traffic in warehouses differs substantially from road traffic. Driving lanes provide only a weak guide, as shorter paths may be preferred. Furthermore, traffic rules are less strictly followed than on roads [8]. Hence, predicting the trajectories of other intralogistics vehicles offers significant challenges compared to road users. The high versatility of intralogistics vehicles adds to the difficulty, since they can execute complex driving maneuvers such as rotating on the spot.

III. TRAJECTORY PREDICTION WITH ENHANCED ENVIRONMENT ENCODING

To enhance trajectory prediction in intralogistics, we propose the addition of a new encoder to model environmental objects. The standard structure of state-of-the-art trajectory prediction models uses different encoder modules to learn agent information and map data. The features from all encoders are then combined and used for trajectory prediction. Standard approaches, including our two baselines, Forecast-MAE [9] and EMP [10], commonly use two encoders (agents and lane data) to extract scene information. To implement our approach, we extend the models by adding a third encoder. This design choice also aligns with models using LiDAR data as additional input [5], [6].

Our two baselines follow the same high-level architecture, but differ in the design of the individual components. Figure 2 shows the network structure and how to integrate our ECTX encoder. Analogously, ECTX could also be integrated into other state-of-the-art models like [5], [11].

A. Agent and Lane Encoding

Both baseline models [9], [10] use a PointNet-like [26] architecture to encode lane data. The lane data must be provided as a list of L lane segments. A fixed number of N 2D points from each segment is used. An additional flag is added to each point to implement padding, *i.e.,* if a segment crosses the observation radius. The lane encoder yields lane tokens with shape $\mathbb{R}^{L\times D}$.

Forecast-MAE [9] uses a neighborhood attention-based [27] agent encoder, whereas EMP [10] utilizes simple self-attention blocks. The comparison of both models shows that the simpler version used in EMP achieves better inference speed and accuracy on the Argoverse 2 benchmark. Both encoder types generate for each agent in the scene a token, resulting in an agent matrix $\mathbb{R}^{A\times D}$.

B. Map Modeling

We generate the map input for our model based on floor plans, which are commonly available for intralogistics sce-

narios. To extract lane information, we sample lane segments from the floor plan. Following recent research [11], [5] we choose to use rather short lane segments. We split all lanes into smaller segments with a fixed size. Then, we sample N points from each segment to prepare our model input.

Floor plans are also the source for generating the input for our environment encoder. Based on a use-case analysis, we identified a set of six environmental elements that have a strong influence on the driving behavior of intralogistics vehicles. The most important influence is the position of racks. They are static obstacles but also indicate that a vehicle might perform a load handling action there, which would mean a change of driving speed and direction. Furthermore, we extract non-driveable areas, which can be used to model static obstacles like walls, machines, pillars, and other warehouse equipment. We also model charging stations, where vehicles might park for longer periods of time. Additionally, we model storage locations, where vehicles might drop off cargoes. Lastly, we also include free areas where it is likely that vehicles will cut corners or move freely even if there are no lane segments.

C. Environment Encoding

The environment elements are defined as a list of polygons $\mathbb{R}^{E\times M\times 3}$. Similar to the lane encoder, we encode the polygons using a small PointNet [26]-like network. The network takes M points sampled from the polygon as input. Compared to the lane encoder, which learns line information, our environment encoder learns the shape of arbitrary polygons. Our environment encoder gives as output an environment matrix with tokens $\mathbb{R}^{E\times D}$.

D. Scene Encoding and Trajectory Decoding

Both baselines apply token type independent self-attention. This allows convenient integration of the additional environment tokens compared to models [12], [13], [3] with explicit type-specific modeling, *i.e.,* agent-to-map, map-to-map, map-to-agent, and agent-to-agent attention blocks. The global position of each scene element is added to the tokens to learn spatial relations. To this end, we use the center pose (x, y, θ) of each element and encode it following [9]. For our environment elements, we use the mean of all points as center and set the angle to 0 since no common orientation can be computed for arbitrary polygon shapes. Type embeddings are used to preserve category-specific data. We use different type tokens for each intralogistics vehicle class and each environment class.

Forecast-MAE [9] uses a simple MLP-based trajectory decoder. EMP uses two different encoder types, a refined MLP-based decoder (EMP-M) and a sophisticated DETR-like decoder (EMP-D). EMP-D uses learnable mode embeddings and then applies cross-attention to the scene tokens.

IV. EXPERIMENTAL SETUP

A. Dataset

We conduct our evaluations on a large-scale dataset generated using NVIDIA Omniverse™. The dataset is recorded in virtual warehouse environments, which are designed based on real-world warehouse layouts and traffic situations. We use CAD models of different real-world intralogistics vehicles and apply custom vehicle controllers to obtain motion patterns that are close to reality.

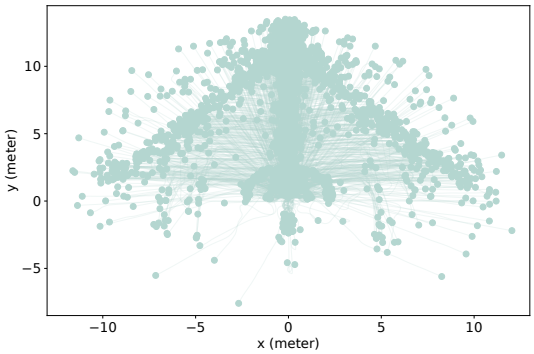

Fig. 3. Distribution of future trajectory in our dataset. We randomly sample 2% of data from out validation set for visualization. The current focal agent position is at the origin with heading in positive y-direction.

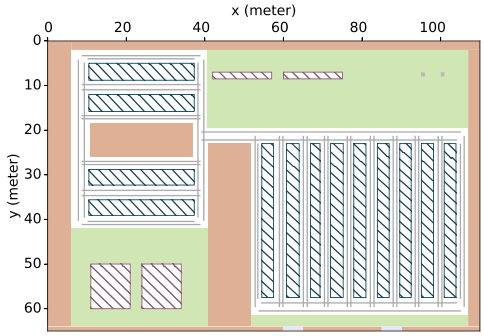

Fig. 4. Map of our warehouse environment including driving lanes and environment elements. Semantic map classes: **dark gray: lanes, blue: racks, green: open areas, orange: non-driveable areas, violet: storage locations, brown: charging stations,** and **light gray: doors**.

Trajectory prediction models commonly use static map data and vehicle tracks as inputs. Since we use real-world vehicle geometry and controller, the tracks accurately match the movements of real vehicles. Thus, our simulated trajectory prediction dataset accurately resembles real warehouse data.

The maps in our dataset contain lane topology implemented as directed graphs and detailed information on intralogistics environment elements. These include charging stations, free areas, rack locations, gates, non-driveable areas (static obstacles), and storage locations. Overall, our dataset features 267,146 scenarios recorded in two different environments. We use data from one environment for training/validation and from the second environment for our test set. We record two separate simulation sequences for training (94,621 scenarios) and for validation (63,605 scenarios). For testing, we use 108,920 scenarios. The rather small ratio for training is a deliberate design decision to closely reflect practical scenarios where only limited data is available.

In our dataset, we utilize a total scenario length of 11 seconds, where $5s$ serve as historic context and the goal is

to predict the following $6s$. Due to the slower speeds of intralogistics vehicles, the maximum trajectory length in our dataset is $13.60m$, and the median length is $10.45m$. The dataset contains difficult intralogistics-specific movements like load handling operations. For our dataset, we use various vehicle types, *i.e.,* reach trucks, forklifts, and order pickers, to capture the different driving dynamics of each type.

Figure 3 shows a subsample of trajectories from our dataset. Compared to autonomous driving datasets, the trajectories with rotations on the spot (arches close to origin) and maneuvers of driving direction change (trajectory endpoints with negative y values) are very distinctive.

Figure 4 shows the environment map we use for our test set. For each scenario, we sample all neighboring map elements and agents within a radius of $25m$ as model input. Given the speed of intralogistics vehicles and our scenario length, this sampling radius includes all relevant context. Based on our real observed warehouse traffic, our dataset scenarios have up to 8 agents at the same time. We split all lanes into segments of approximately $10m$ and sample $N = 20$ points from each segment. Using a point distance of approximately $0.5m$ yields a robust yet efficient representation. Analogously, we sample data from our environment elements polygons. Due to large environment elements, we limit the sampling points to a maximum of $M = 160$ points per polygon to achieve a good accuracy/complexity trade-off.

B. Implementation Details

We integrate our ECTX module into two state-of-the art trajectory prediction models, *i.e.,* Forecast-MAE [9] and EMP [10]. We do not utilize a pretrain scheme for Forecast-MAE to keep our training efficient. This is essential considering our intralogistics use case, where finetuning on special warehouse or factory layouts might be necessary. As a post-processing step, we remove low scoring trajectories (probability lower than 0.01) from our model output. For both baselines, we evaluate the original network configurations, as well as a smaller version with reduced feature dimension to $D = 64$. The baseline models are designed for autonomous driving applications, where large-scale datasets are available. For robotics applications like intralogistics, dataset sizes are often limited. This can lead to issues due to insufficient training data. Reducing the model size also aligns well with our intended use case of deployment on embedded hardware.

We use a batch size of 128 and train our models for 60 epochs on one NVIDIA V100 GPU. The first 10 epochs are used as warm-up, where we increase the learning rate linearly from 1e-6 to 8e-5. We then gradually decrease the learning rate to 1e-6 again using a cosine-schedule. As an optimizer, we use AdamW [28] with gradient clipping. We apply weight decay and do not use any data augmentations.

Following our baselines [9], [10], we combine three loss terms: We apply hard-assignment and only optimize the focal agent trajectory with the lowest deviation (smallest L^2-norm) from the ground truth trajectory. We compute a regression loss (Huber loss [29]) between the best-fitting trajectory and the ground truth trajectory. Furthermore, we also compute a classification loss (cross-entropy loss) for predicting the highest probability for the best fitting trajectory. In addition, we compute a regression loss on a single future trajectory for each neighboring agent in the scene. This helps to facilitate better learning behavior in the agent encoder.

V. RESULTS AND DISCUSSION

We present detailed evaluations on our custom intralogistics dataset by comparing three baselines models with and without our ECTX encoder. In the following, we study the influence of the lane and environment encoder modules on trajectory prediction accuracy. Subsequently, we evaluate the model size and the inference speed of our baselines compared to the version with the additional ECTX encoder.

We evaluate our models using standard trajectory prediction metrics [18], [30], [25]: minADE_K, minFDE_K, brier-minFDE_K and, MR_K. Each metric is computed using a set of K trajectory hypotheses. We set the output of our model to 4 trajectory hypotheses, as this allows us to model diverse future movements while still providing a compact representation for integration into the control system of an AMR. We compute our metrics on the most probable prediction ($K = 1$) and for all predictions ($K = 4$).

A. Evaluation on Intralogistics Dataset

Table I compares our baselines, Forecast-MAE [9] and EMP-M/D [10], with and without our new ECTX module on our custom intralogistics dataset. For all baselines we test the original configurations and our reduced model configurations (marked by a *). For all three models, the addition of ECTX significantly increases the trajectory prediction accuracy on our test set. Additionally, our updated model configurations outperform the original configurations on our test set, since their model capacity aligns better with the dataset size. Using the original model configurations, the models easily overfit (see their validation set performance), leading to bad generalization performance (as indicated by the test set performance). This algins well with our intended use-case, where only limited computation resources are available on the robot.

The performance gap between our validation set (the same environment as training data) and our test set (different environment) highlights the influence of layout-specific map context. As intralogistics often features repetitive task execution (*i.e.,* similar motion patterns) within one warehouse, the performance of all models is significantly better on the validation data compared to the test set. Comparing the train-to-validation gap of the baseline models and our approach, we observe better generalization capabilities. Our approach generalizes well to other environments, but finetuning on target-specific map data can further increase its performance. Since our model only requires little training resources and floor plans are generally available for production facilities and warehouses, finetuning our method is easily feasible.

Aligned with the results from [10] on AV2, EMP-D outperforms EMP-M since it uses a more sophisticated decoder

TABLE I

RESULTS ON OUR INTRALOGISTICS DATASET GROUPED BY BASELINE MODEL AND SORTED IN DESCENDING ORDER BY TEST BRIER-MINFDE$_4$. DENOTES THE PERFORMANCE OF THE REDUCED MODEL CONFIGURATION, WHICH HAS A MORE SUITABLE CAPACITY FOR AMR DEPLOYMENT WITHIN INTRALOGISTICS SCENARIOS. THE BEST SCORES ARE SET IN BOLD, SECOND BEST ARE UNDERLINED.

	Validation Set				Test Set					
Method	MR_4	$minADE_4$	$minFDE_4$	brier-$minFDE_4$	MR_4	$minADE_1$	$minFDE_1$	$minADE_4$	$minFDE_4$	brier-$minFDE_4$
EMP-M [10]	0.055	0.347	0.679	0.928	0.156	1.124	2.834	0.571	1.206	1.557
EMP-M*	0.090	0.398	0.809	1.070	0.182	1.090	2.689	0.572	1.221	1.517
EMP-M*+ECTX	0.090	0.398	0.806	1.074	0.155	1.081	**2.671**	0.546	1.169	1.455
EMP-D [10]	**0.035**	**0.313**	**0.593**	**0.804**	0.140	1.121	2.836	0.513	1.067	1.410
EMP-D*	0.056	0.350	0.682	0.912	0.129	1.108	2.761	0.512	1.061	1.353
EMP-D*+ECTX	0.046	0.341	0.653	0.879	0.129	1.145	2.834	0.501	1.032	1.325
Forecast-MAE [9]	0.037	0.321	0.604	0.814	**0.117**	1.164	2.967	0.479	**0.979**	1.336
Forecast-MAE*	0.065	0.367	0.718	0.952	0.126	**1.060**	**2.672**	0.487	1.033	1.318
Forecast-MAE*+ECTX	0.058	0.355	0.690	0.913	**0.117**	1.063	2.695	**0.473**	0.982	**1.266**

TABLE II

ABLATION STUDY ON THE INFLUENCE OF DIFFERENT ENCODER MODULE USING FORECAST-MAE*. THE EXPERIMENT MARKED WITH ✓* USES ONLY A SINGLE ENCODER FOR ENVIRONMENT AND LANE DATA.

Context Encoder		Test Set			
Lanes	ECTX	MR_4	$minADE_4$	$minFDE_4$	brier-$minFDE_4$
✓*		0.323	0.837	1.721	2.026
✗	✗	0.251	0.671	1.489	1.827
✗	✓	0.188	0.637	1.380	1.703
✓	✗	0.126	0.487	1.033	1.318
✓	✓	0.117	0.473	0.982	1.266

TABLE III

COMPARISON OF MODEL SIZE AND INFERENCE SPEED (LATENCY FOR BATCH PREDICTIONS OF 32 SCENARIOS ON A SINGLE NVIDIA V100 GPU). ALL MEASUREMENTS ARE DONE BY US.

Method	Model Parameters	Latency
Forecast-MAE*+ECTX	860K	38 ms
EMP-D*+ECTX	1.2M	36 ms
Forecast-MAE* [9]	678K	32 ms
EMP-M*+ECTX	791K	31 ms
EMP-D* [10]	1.0M	29 ms
EMP-M* [10]	609K	24 ms

TABLE IV

LATENCY ANALYSIS FOR PRACTICAL USE OF OUR APPROACH ON ROBOTICS HARDWARE. WE MEASURE THE LATENCY FOR PREDICTING ALL AGENTS FROM THE LARGEST SCENE IN OUR TEST SET (8 VEHICLES AND 110 SCENE CONTEXT ELEMENTS). WE ALSO PROVIDE COMPARISON FOR A DESKTOP GPU.

	NVIDIA Jetson		NVIDIA
Method	Orin Nano	AGX Orin	V100
EMP-D*+ECTX	38 ms	15 ms	28 ms
EMP-M*+ECTX	37 ms	15 ms	22 ms

architecture. In contrast to the autonomous driving benchmarks, we achieve better results using Forecast-MAE than EMP. The intralogistics scenarios feature significantly fewer agents per scene. Hence, using the neighborhood attention-based [27] agent encoder from Forecast-MAE brings an advantage over the pure transformer version from EMP-M/D.

Overall, we achieve highly accurate results on our validation and test sets. Comparing the evaluation results for $K = 1$ and $K = 4$, we can see that the introduction of ECTX has a beneficial influence for multiple hypotheses. This indicates that the additional environmental information has a strong impact on modeling multi-modality.

Figure 5 shows the predictions of EMP-D* without and with ECTX on two example scenarios from our test dataset. The first scenario (top row) shows a vehicle driving in a *rack pre-zone*. Given its movement, there is the possibility that it enters a rack aisle or continues in the open area. Without using the rack positions as additional input, EMP-D* outputs a potential left turn, which would lead to a collision with a rack and hence is invalid. Adding the rack positions using our ECTX encoder leads to well-suited trajectory predictions. In the second scenario (bottom row), a vehicle is leaving an aisle, where it can either go left or right. Using only lane data as context, the model outputs that the vehicle might cut the corner. Again, this would lead to a collision with a static obstacle, and using our additional encoder yields plausible results.

B. Ablation Study

We conduct an ablation study, where we evaluate the influence of lane and environmental context on trajectory prediction. To this end, we conduct experiments for map-free trajectory prediction on our dataset and for using only our new ECTX encoder as context without lane data.

Table II shows the results for our ablation study. We use the Forecast-MAE models with/without ECTX (see Table I) as baseline comparisons. As expected, the map-free model using only agent data yields the worst results. Without additional map context, predictions are limited to different driving speed variations but, *e.g.,* indicators for future turn maneuvers cannot be considered. Using the position of environment elements in ECTX as context, we see that the trajectory prediction accuracy is increased. We also conduct an experiment where we utilize a single encoder module to encode both lane polylines and environment polygons simultaneously. The experiment shows modeling of the environment elements requires a separate encoder module. Using a single encoder for both modalities leads to worse results since the model does not learn the context guidance. This confirms that using the map elements gives a viable input for trajectory prediction. As expected, the addition of lane data yields the best results overall.

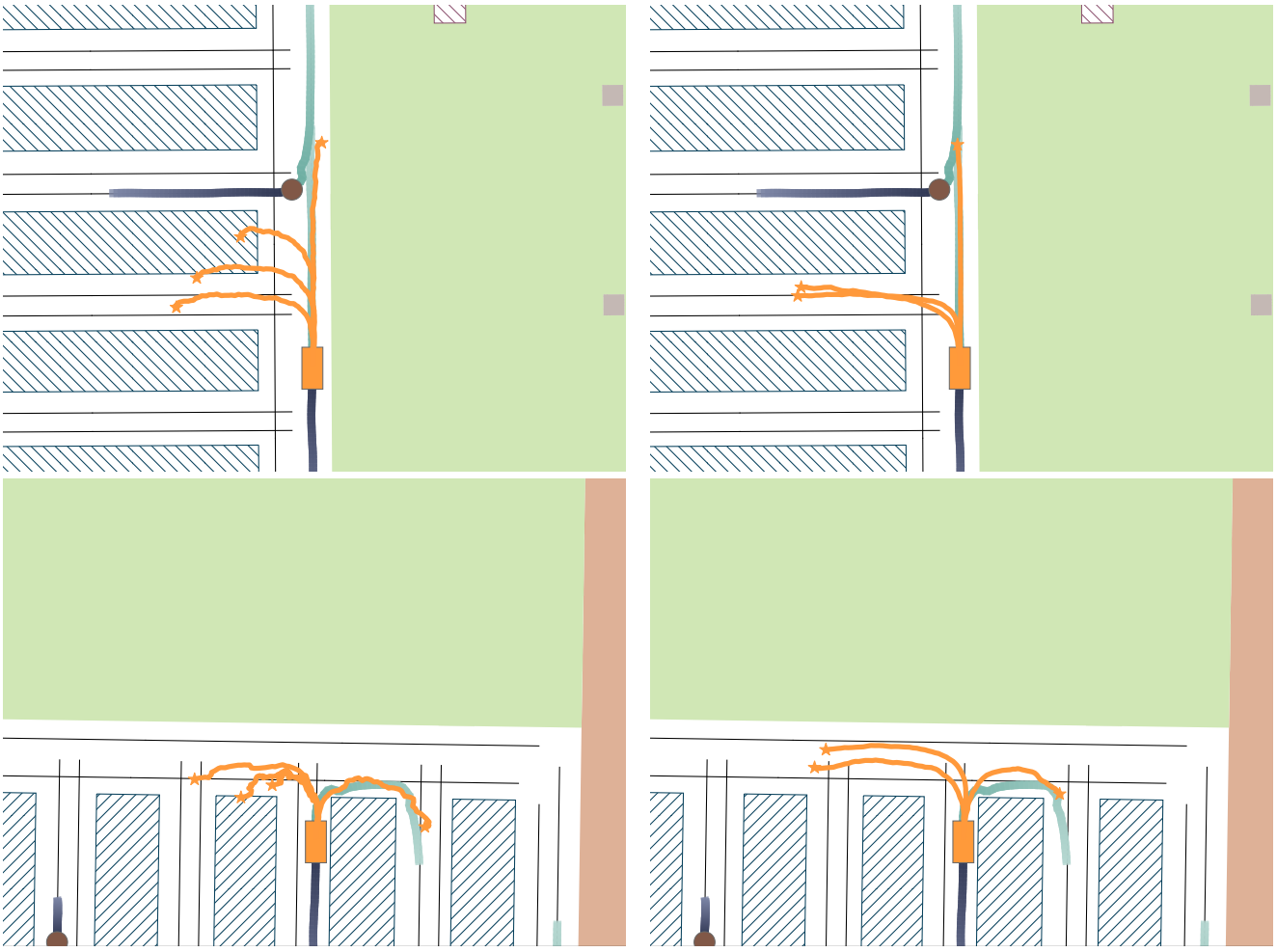

Fig. 5. Both image pairs show scenarios from our custom intralogistics dataset (test set). For each scenario, the left image shows **predictions** from our baseline EMP-D* model and the right image from a EMP-D*+ECTX. In all images also the **ground truth trajectory** is shown. Using ECTX the possible turn inside the rack aisles much better aligns with the given environment structures. For the second example using ECTX the predicted trajectories do not interfere with the racks as the corners are not cut.

C. Resource Analysis

In Table III we compare the number of model parameters and the inference latency for all models presented in Table I. To measure the inference latency, we make predictions on a batch of 32 scenarios using a single NVIDIA V100 GPU. Overall, EMP-M* has the smallest model size and the fastest inference latency, whereas Forecast-MAE is the slowest. This also correlates with the comparisons of both models on autonomous driving datasets [10]. Adding our ECTX encoder slightly increases the inference latency due to the additional model component and increased scene context size. However, our approach still easily achieves real-time trajectory forecasting and provides a notably improved prediction accuracy with better generalization capabilities (*cf.* Table I) which is a desirable runtime-accuracy tradeoff in the intralogistics domain.

In Table IV, we present the real-world inference latency of our EMP-based models when predicting all agents in the most complex scene from our test set. We evaluate performance on two types of NVIDIA Jetson devices, which are designed for robotics applications, and include a comparison with a standard NVIDIA GPU. The results highlight that our approach is very well suited for real-time processing on AMR hardware. Forecast-MAE could not be tested in this evaluation, because it uses neighborhood attention blocks [27], which are not supported for export to ONNX[1], a requirement to port models to embedded hardware platforms for efficient inference.

TABLE V

RESULTS FOR INTEGRATING OUR ECTX MODULE TO EMP-M [10] FOR TRAJECTORY PREDICTION ON THE ARGOVERSE 2 [25] VALIDATION SET.

	Validation Set			
Method	MR_6	$minADE_6$	$minFDE_6$	b-$minFDE_6$
EMP-M [10]	0.191	0.730	1.457	2.098
EMP-M+ECTX	0.191	0.724	1.448	2.086

D. Results on Argoverse 2

We also present experimental results on the AV2 validation set in Table V. In this standard road traffic dataset, the performance gain is only marginal. This, however, is expected

[1] `https://onnx.ai/`.

because AV2 primarily features car trajectories (where the lane information already provides a strong prior), and furthermore, includes only driveable-area annotations (limiting the available scene context for ECTX). Our ECTX module still slightly enhances the prediction performance, which demonstrates its potential for improvements in other domains besides our targeted intralogistics scenarios.

VI. CONCLUSIONS

To solve trajectory prediction in complex intralogistics environments, we propose a new **e**nvironment **c**on**t**e**x**t (ECTX) encoder, which can be used as an extension to a wide range of state-of-the-art trajectory prediction models. Our work emphasizes that for custom use cases, such as commonly observed within intralogistics, specialized compact model designs outperform standard large-scale models which are designed for autonomous driving in road traffic scenarios. In our evaluations, we demonstrate that the integration of the proposed ECTX encoder yields notably improved results using a wide range of different baselines.

ACKNOWLEDGMENTS

This work was partially supported by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association.

REFERENCES

[1] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal Motion Prediction with Stacked Transformers," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7577–7586, 2021.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] A. Prutsch, H. Possegger, and H. Bischof, "Action-By-Detection: Efficient Forklift Action Detection for Autonomous Mobile Robots in Warehouses," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2024.

[9] 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.

[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] 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)*, 2024.

[12] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] 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.

[14] 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.

[15] 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.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020.

[17] 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.

[18] 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.

[19] L. Zhang, P. Li, S. Liu, and S. Shen, "SIMPL: A Simple and Efficient Multi-agent Motion Prediction Baseline for Autonomous Driving," *arXiv preprint arXiv:2402.02519*, 2024.

[20] J. Xiang, J. Zhang, and Z. Nan, "A Fast and Map-Free Model for Trajectory Prediction in Traffics," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023.

[21] 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.

[22] 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.

[23] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, *et al.*, "MultiPath++: Efficient Information Fusion and Trajectory Aggregation for Behavior Prediction," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2022.

[24] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion Forecasting via Simple & Efficient Attention Networks," in *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2023.

[25] 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.

[26] 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, pp. 652–660.

[27] A. Hassani, S. Walton, J. Li, S. Li, and H. Shi, "Neighborhood Attention Transformer," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[28] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv preprint arXiv:1711.05101*, 2017.

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

[30] 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.