2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning Location-Specific Latent Behavior Priors for Occupancy Prediction in Automated Driving

Julian Schmidt[1,†], Mario Ruoff[1], Christoph Rist[1,2], and Julian Jordan[1]

***Abstract*— Performance in automated driving tasks improves significantly with the incorporation of location-specific prior knowledge. This is because agent behavior usually strongly correlates with location features. A common example is the strong tendency of vehicles to follow their lane, but less obvious interactions exist as well. To this end, high definition (HD) map information is typically collected and made available during both training and inference to act as a location prior.**

In this paper, we propose to aggregate location-specific information in a data-driven way. Specifically, we learn a global latent grid that acts as a behavior prior to a learned occupancy prediction model. Since the prediction loss function is directly backpropagated into the latent grid, no additional labels are required beyond the already available future agent locations. We use the large real-world Lyft Level 5 motion prediction dataset to empirically demonstrate the merit of our learned location-specific latent behavior prior. Applied to two different prediction models, our approach achieves performance comparable to or exceeding baseline models that rely on HD maps, without requiring an HD map. Additional experiments reveal that the latent behavior prior is able to distill geometric and semantic information purely from agent behavior. These results indicate that directly learning location-specific priors is a promising direction towards automated driving without costly HD maps.

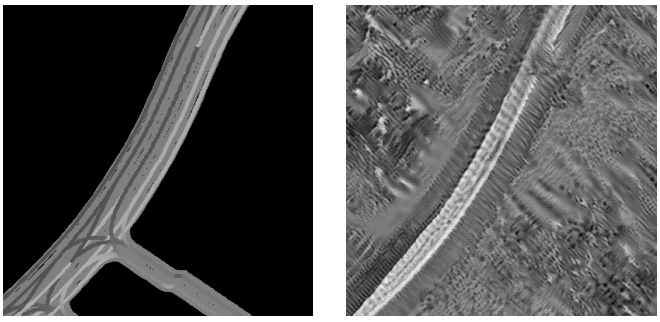
I. Introduction

Agent behavior is inherently location-specific. To account for this, many automated driving systems utilize high definition (HD) maps. The detailed geometric and semantic information provided by these maps enables the automated driving algorithms—such as those used for prediction and planning—to more effectively reason about agent behavior. For instance, knowing the topology and geometry of lanes allows to predict the motion of surrounding traffic agents more precisely.

Obtaining HD maps is a complex process [1]. A common approach leverages deep learning-based methods trained on raw sensor data to predict static infrastructure entities, as demonstrated in [2], [3]. This requires well-curated and manually labeled training datasets. However, to the best of our knowledge, the predicted HD maps may still contain errors, particularly in challenging scenarios [3].

*This work is a result of the joint research project "STADT:up" (19A22006O) and of the joint research project "NXT GEN AI METHODS – Generative Methoden für Perzeption, Prädiktion und Planung". The projects are supported by the German Federal Ministry for Economic Affairs and Energy (BMWE), based on decisions of the German Bundestag. The authors are solely responsible for the content of this publication.

[1]Mercedes-Benz AG, Research & Development, Stuttgart, Germany
[2]Intelligent Vehicles Group, TU Delft, The Netherlands
[†]julian.sj.schmidt@mercedes-benz.com



(a) HD map (simplified plot) (b) Ours: latent behavior prior

Fig. 1: While (a) current approaches use high definition (HD) maps as an input to occupancy prediction models, we propose to (b) directly learn location-specific latent behavior priors using the loss function of the final task, i.e., the occupancy prediction loss.

Rather than relying on HD maps as an intermediate representation, this paper investigates learning location-specific latent behavior priors directly from data. For this, we focus on the motion prediction task, more specifically the occupancy prediction task. In occupancy prediction, the goal is to predict a grid where the value of each grid cell represents the probability that the cell is occupied by any agent at a future timestep. Previous work related to learned behavior priors in motion prediction [4], [5] focuses on the prediction of pedestrian trajectories from static cameras and is therefore limited to a few selected locations only. Gilitschenksi et al. [5] introduce additional auxiliary tasks to supervise the training of the behavior prior, whereby one of the auxiliary task requires semantic labels of the selected locations. We argue that for approaches to be effective for automated driving, they should (a) scale to large areas and (b) be trainable without any additional labels. In this work we present a novel approach that covers both of these aspects and evaluate it on the occupancy prediction task using a large-scale dataset recorded from an egocentric perspective. For this, we consider not only pedestrians but also cyclists and vehicles of different types.

On a high level, the approach works as follows. As illustrated in Fig. 1, instead of using HD map information, a latent behavior prior is used. This behavior prior is represented as a local latent grid centered on the automated vehicle. Fig. 1b illustrates a single-channel latent grid, which our evaluations have found to be effective already, although the grid can have multiple channels. Each local latent grid is obtained

by cropping a patch from a large global latent grid. During the training of the occupancy prediction model, the local grid acts as a learnable parameter that gets updated using only the gradients propagated back from the prediction loss function. The global latent grid is updated with the local patches from the current training batch and the updated version is available for the next training batch. Thus, a prerequisite of our approach is that geographic locations need to be seen a few times until the training process captures their features in the latent behavior prior.

In summary, our main contributions are as follows:

- We propose a data-driven approach to learn location-specific behavior priors directly optimized for the final occupancy prediction task. The prior is represented as a latent grid and training it does not require any labels other than the labels for the final task.
- We show that our approach scales to a large real-world dataset recorded by a fleet of vehicles.
- We extensively evaluate our approach using two distinct occupancy prediction models, achieving performance similar to or better than HD map-reliant approaches. Our results indicate that the proposed approach can distill geometric and semantic information purely from agent behavior.

II. Related Work

This section discusses related work concerning latent priors in the context of automated driving in Section II-A and motion prediction without HD maps in Section II-B.

A. Latent Priors for Automated Driving

Related work in learning a location-specific memory, i.e., a prior, originates in robotics [6], [7], [8]. For automated driving itself, there is few related work. Xiong et al. [9] learn a location-specific latent prior for the task of online HD mapping. Cross-attention and a recurrent neural network are used to fuse information coming from the prior with the current sensor observations. The resulting HD maps are intended to assist during the downstream behavior-related tasks, i.e., prediction and planning. Building HD maps to assist downstream tasks is also done in offline map-learning, whereby computationally expensive algorithms can be used [10]. Online and offline HD mapping is different from our work, as we do not focus on generating HD maps as an intermediate representation. Instead, we aim to directly optimize the latent prior for the behavior-related occupancy prediction task. Yi et al. [4] and Gilitschenski et al. [5] align with this idea. Their work focuses on learned behavior priors for pedestrian trajectory prediction, using recordings from static cameras that are restricted to one or a few selected locations. Gilitschenski et al. additionally supervise the training of the behavior prior using auxiliary tasks, whereby one even requires labels. The restriction to pedestrian prediction in specific locations only and the use of auxiliary tasks limits the applicability of these approaches. As shown later in this paper, our approach works for a large-scale and in-vehicle recorded egocentric dataset with all types of agents.

B. Motion Prediction Without High Definition Maps

Motion prediction approaches [11], [12], [13], [14] and approaches to predict the intents of agents [15] most commonly rely on HD maps as one main source of information. For instance in VectorNet [12], entities from the HD map (e.g., lanes and crosswalks) are transformed into a vector representation that is then processed together with the detected agents by a neural network. In LaneGCN [13], the HD map is used to construct a lane graph that preserves the structure of the map. The lane graph is then processed by a graph neural network and fused with information about the detected agents. Due to the problematic scalability of HD map creation, there are also approaches that focus on prediction without HD map information. In [16], [17], completely map-free approaches using attention are proposed. However, their performance cannot match the performance of their counterparts that use HD maps. In [18], the use of globally available navigation maps is investigated. The results show that adding information from navigation maps substantially improves results relative to using no map. Nevertheless, a gap remains between these approaches and HD map-based approaches. Our approach does not use HD maps either. Instead, we learn a latent behavior prior by aggregating information from multiple data samples. In contrast to the aforementioned approaches, our approach consistently matches or outperforms the HD map-reliant approaches.

III. Problem Statement

We investigate directly learning location-specific latent behavior priors using the occupancy prediction task. Occupancy prediction is the task of predicting a spatial-temporal occupancy grid $\mathbf{O}_i \in \mathbb{R}^{\mathrm{T_{fut}} \times \mathrm{h_G} \times \mathrm{w_G}}$ that is centered on the automated vehicle. Here, i is the index of the current data sample. The scalar value $o \in \mathbb{R}$ of each cell is the probability that the cell is occupied by any agent at a future timestep $t \in \mathrm{T_{fut}}$. $\mathrm{h_G}$ and $\mathrm{w_G}$ are the height and width of the grid. They are determined by multiplying the desired field of view in Euclidean coordinates ($\mathrm{h_E}$ and $\mathrm{w_E}$) with the resolution $r \in \mathbb{R}$ $(\mathrm{cells/m})$ of each grid cell. During downstream planning, the probability per cell enables to determine whether specific areas in the surrounding of the automated vehicle will be navigable or occupied in the future. For the task of occupancy prediction, machine learning-based models have been proven to be effective, e.g., methods based on convolutional neural networks [19] or Transformers [20], [21].

In this work, we focus on occupancy prediction models that utilize a rasterized input representation. This means that the agents detected and tracked by perception components, more specifically their bounding boxes, are rendered as a bird's-eye view representation centered on the automated vehicle. Consequently, the input to the model is also a grid $\mathbf{A}_i \in \mathbb{R}^{\mathrm{T_{past}} \times \mathrm{h_G} \times \mathrm{w_G}}$. $\mathrm{h_G}$ and $\mathrm{w_G}$ correspond to the same height and width as in the predicted occupancy grid $\mathbf{O}_i$. To

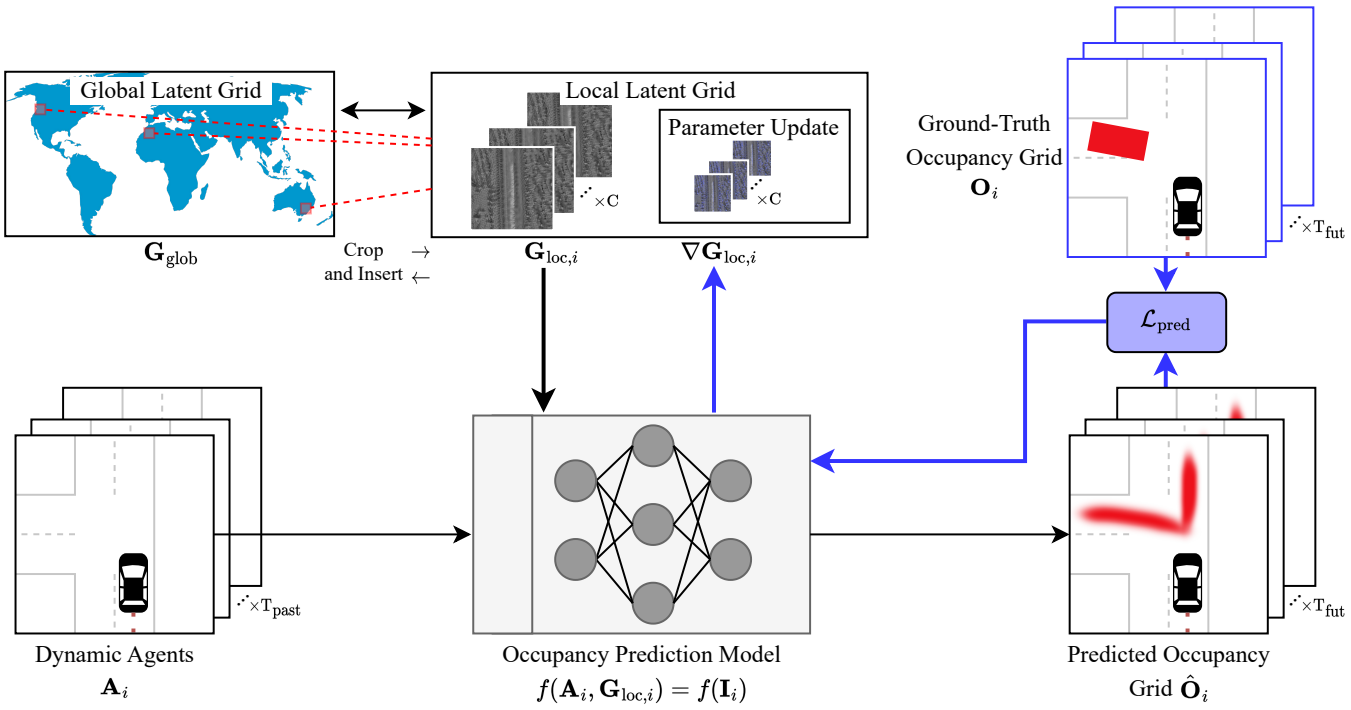


Fig. 2: Overview of our approach: In addition to information about dynamic agents $\mathbf{A}_i$, the occupancy prediction model $f(\mathbf{A}_i, \mathbf{G}_{\text{loc},i}) = f(\mathbf{I}_i)$ is provided with a local patch $\mathbf{G}_{\text{loc},i}$ from the global latent grid $\mathbf{G}_{\text{glob}}$. Blue indicates the update of the global latent grid during training directly using the prediction loss function $\mathcal{L}_{\text{pred}}$. The occupancy prediction model in this architecture is interchangeable.

retain the temporal aspect of the agents tracked by perception components, multiple past timesteps T_{past} are rendered. Our baseline models $f(\mathbf{A}_i, \mathbf{M}_i)$ additionally consume a rasterized representation of the HD map $\mathbf{M}_i \in \mathbb{R}^{4\times \mathrm{h_G}\times \mathrm{w_G}}$ centered on the automated vehicle.

IV. Approach

This section describes our approach of learning location-specific latent behavior priors for the occupancy prediction task. At its core, this involves forming a global latent grid that aggregates information from multiple geographically overlapping egocentric data samples. As later shown in the experiments section (Section V), our approach is applicable to different occupancy prediction models. The only prerequisite is the ability of the model to process grid-based input data, i.e., the latent grid.

Each data sample i is centered on the automated vehicle, i.e., egocentric, and only covers an area of size $\mathrm{h_G} \times \mathrm{w_G}$. In contrast, the global latent grid $\mathbf{G}_{\text{glob}} \in \mathbb{R}^{\mathrm{C}\times \mathrm{H_G}\times \mathrm{W_G}}$, which represents the latent behavior prior, is intended to cover the entire size of the dataset $\mathrm{H_G} \times \mathrm{W_G}$ and must aggregate information from multiple potentially overlapping data samples such that it can store information about location-specific behavioral patterns.

A. Learning Location-Specific Latent Behavior Priors From Egocentric Data Samples Using a Global Latent Grid

Fig. 2 illustrates the components of our approach along with the associated forward and backward dataflows among them. Before training, the global latent grid $\mathbf{G}_{\text{glob}}$ is initialized with zeros. As described, data sample i consists of a rendered grid $\mathbf{A}_i$ representing the agents tracked by perception components. The center of $\mathbf{A}_i$ is the global Euclidean coordinates $(y, x)_{\text{E,center},i}$. Data sample i now gets enhanced with additional information in the form of a location-specific latent behavior prior. This prior is represented as a patch $\mathbf{G}_{\text{loc},i}$, namely the local latent grid, extracted from the global latent grid. The patch is defined as

$$\begin{aligned}\mathbf{G}_{\text{loc},i} = \mathbf{G}_{\text{glob}}[&y_{\text{G,patch},i} : y_{\text{G,patch},i} + \mathrm{h_G},\\ &x_{\text{G,patch},i} : x_{\text{G,patch},i} + \mathrm{w_G}].\end{aligned} \tag{1}$$

The coordinates used for extracting the patch can be calculated as

$$\begin{pmatrix} y \\ x \end{pmatrix}_{\text{G,patch},i} = r \cdot \begin{pmatrix} y \\ x \end{pmatrix}_{\text{E,center},i} - \begin{pmatrix} \lfloor \frac{\mathrm{h_G}}{2} \rfloor \\ \lfloor \frac{\mathrm{w_G}}{2} \rfloor \end{pmatrix}. \tag{2}$$

To make this additional piece of information available to the corresponding occupancy prediction model, we concatenate $\mathbf{G}_{\text{loc},i}$ and $\mathbf{A}_i$ along the first dimension. This concatenation results in $\mathbf{I}_i \in \mathbb{R}^{(\mathrm{C}+\mathrm{T_{fut}})\times \mathrm{H}\times \mathrm{W}}$. The corresponding occupancy prediction model $f(\mathbf{A}_i, \mathbf{G}_{\text{loc},i}) = f(\mathbf{I}_i)$ is now provided $\mathbf{I}_i$ instead of only $\mathbf{A}_i$, allowing it to additionally reason about patterns in the latent grid.

Training the model requires an additional update and insertion step to feed back information into the global latent grid. During the update step, the local latent grid of the data sample is updated, i.e., $\mathbf{G}'_{\text{loc},i} \leftarrow \mathbf{G}_{\text{loc},i}$. This update

is carried out jointly with the parameters of the prediction model through backpropagation. This allows the existing prediction loss function, in our case the focal loss, to also update the latent behavior prior. Similar to the weights of the prediction model, the local latent grid in this case acts as a learnable parameter and gradients of the loss function with respect to each parameter in the local latent grid are used to update each value, i.e.,

$$\mathbf{G}'_{\mathrm{loc},i} \leftarrow \text{Optimizer}(\mathbf{G}_{\mathrm{loc},i}, \nabla_{\mathbf{G}_{\mathrm{loc},i}} \mathcal{L}_{\mathrm{pred}}) \tag{3}$$

Subsequently, each updated local latent grid $\mathbf{G}'_{\mathrm{loc},i}$ is inserted into the global latent grid $\mathbf{G}_{\mathrm{glob}}$. This is done by

$$\begin{aligned} \mathbf{G}_{\mathrm{glob}}[&y_{\mathrm{G,patch},i} : y_{\mathrm{G,patch},i} + \mathrm{h_G}, \\ &x_{\mathrm{G,patch},i} : x_{\mathrm{G,patch},i} + \mathrm{w_G}] = \mathbf{G}'_{\mathrm{loc},i}. \end{aligned} \tag{4}$$

which is the inverse operation to Equation 1.

B. Sparsifying the Learning Process Using Masking

The described approach will always update the entire local latent grid $\mathbf{G}_{\mathrm{loc},i}$. We argue that always updating the entire local latent grid makes the learning process difficult, as many updated cells might not even be relevant for the current scenario. To reduce this noise during learning, we employ a masking strategy that sparsifies the update of the local latent grid before inserting it into the global latent grid. The general idea of this strategy is that cells in the local latent grid that are not occupied by agents might contain a high amount of irrelevant information or noise, and by focusing only on the occupied cells this noise is reduced. Therefore, we only update the grid cells that are occupied during one of the past timesteps $\mathrm{T_{past}}$. This can be denoted as

$$\tilde{\mathbf{G}}'_{\mathrm{loc},i} \leftarrow \mathbf{B}_i \odot \mathbf{G}'_{\mathrm{loc},i} + (1 - \mathbf{B}_i) \odot \mathbf{G}_{\mathrm{loc},i}, \tag{5}$$

whereby $\odot$ denotes the element-wise product. The binary mask $\mathbf{B}_i$ is obtained by

$$\mathbf{B}_i = \bigvee_{t=1}^{\mathrm{T_{past}}} \mathbf{A}_i[t, :, :]. \tag{6}$$

The influence of this strategy is evaluated in Section V-D.

C. Implementation Details

The global latent grid must be accessible during training for both reading and writing local patches with random access. To ensure scalability to large-scale real-world datasets, we store the global latent grid on disk, as its size may grow significantly. The cost of disk space is relatively low compared to memory. We choose the zarr format [22] for our implementation. Zarr enables data storage in manually configurable chunks, allowing for efficient read and write access while also supporting sparsity to reduce storage usage. This sparsity is achievable by zero-initialization of the global latent grid.

We use $\mathrm{h_E} = \mathrm{w_E} = 75\,\mathrm{m}$ and $1/r = 0.1875\,\mathrm{m/cell}$. This results in $\mathrm{h_G} = \mathrm{w_G} = 400\,\mathrm{cells}$. The number of latent grid channels is set to $\mathrm{C} = 64$. An ablation study on C is given in Section V-F. Training is supervised with a pixel-wise focal loss ($\alpha = 0.5$, $\gamma = 2.0$) [23] that compares the predicted occupancy grid $\hat{\mathbf{O}}_i$ with the ground-truth $\mathbf{O}_i$. AdamW [24] with a global batch size of 48 (6 per GPU) serves as the optimizer. The learning rate decays from 10^{-3} to 0 within 50 epochs using cosine annealing scheduling. Weight decay of 10^{-2} is applied to all learnable parameters except norms. When rendering the input grid $\mathbf{A}_i$, different float values are used to encode different agent types.

V. Experiments

This section covers the experimental setup and the quantitative and qualitative results.

A. Dataset

Experiments are conducted using the Lyft Level 5 motion prediction dataset [25], which contains 145k scenes (training + validation) of 25 s each. An important requirement for our approach is a dataset that has many recorded agent movements at the same geographic locations so that the latent behavior prior can be learned and evaluated robustly. Because the Lyft dataset covers a relatively small geographic area, recorded scenes frequently overlap spatially, making it well-suited for this purpose. We use $\mathrm{T_{past}} = 30$ past and $\mathrm{T_{fut}} = 50$ future frames sampled at 10 Hz, each data sample therefore spans 8 s. From every 25 s scene we extract three consecutive data samples. A data sample is spatially centered on the automated recording vehicle and contains agents of type vehicle, pedestrian, and cyclist. Agents with an unknown type are removed. The Lyft dataset has a predefined training-validation split of approximately 90 % and 10 %. To obtain an unbiased testing set, we further divide the provided validation split equally into validation and testing subsplits, resulting in an approximate final split of 90 % training, 5 % validation, and 5 % testing.

B. Metrics

Performance of our approach on the occupancy prediction task is evaluated using three metrics for binary classification:

- Average precision (AP) calculated at 100 linearly spaced thresholds in the range from 0 to 1.
- Soft Intersection over Union (Soft IoU) [26], which is a continuous version of the IoU.
- Intersection over Union (IoU) with a threshold of 0.5.

All metrics are averaged across all timesteps of the testing split.

C. Baseline Models

This work focuses on proposing an alternative to costly HD maps: the learned latent behavior prior. Thus, our experiments focus on two aspects. (a) Showing that our approach can be applied to different occupancy prediction models and (b) comparing it to HD map-reliant variants of the same model architectures. To cover aspect (a), we choose two baseline models with fundamentally different architectures: a convolutional neural network (U-Net [27]) with $17.32 \cdot 10^6$ learnable parameters and a Transformer-based method (DAE-Former [28]) with $48.65 \cdot 10^6$ learnable parameters.

TABLE I: Performance comparison of different U-Net and DAE-Former variants (map-free, with HD map, and with our latent behavior prior) on the testing split. The **best** and second-best values are highlighted. HD map performance is consistently either exceeded or matched with our approach.

Name	U-Net			DAE-Former		
	AP [%] ↑	Soft IoU [%] ↑	IoU [%] ↑	AP [%] ↑	Soft IoU [%] ↑	IoU [%] ↑
Map-Free	70.26	11.04	51.65	78.54	16.00	59.21
W/ HD Map	74.54	14.60	55.43	**81.15**	**18.49**	**61.47**
Ours: W/ Latent Behavior Prior	75.19	14.76	55.88	80.37	17.53	60.85
Ours: W/ Latent Behavior Prior*	**75.45**	**15.09**	**56.16**	80.79	18.34	61.12

*With the masking strategy

The **U-Net** has originally been developed for biomedical image segmentation. It is a powerful, yet computational- and data-efficient architecture. The **DAE-Former** has also been initially developed for biomedical image segmentation. Its efficient design of self-attention enables to capture spatial and channel relations across the entire receptive field. To make our input grid of size $\mathrm{h_G} = \mathrm{w_G} = 400$ compatible with our adapted DAE-Former implementation, we upscale it to $\mathrm{h_G} = \mathrm{w_G} = 512$ when feeding it into the model and downscale it to $\mathrm{h_G} = \mathrm{w_G} = 400$ directly at the model output using bilinear interpolation.

To cover aspect (b), we implement three variants for each baseline model: (i) map-free $f(\mathbf{A}_i)$, (ii) with HD map $f(\mathbf{A}_i, \mathbf{M}_i)$, and (iii) with our learned latent behavior prior $f(\mathbf{A}_i, \mathbf{G}_{\mathrm{loc},i})$. The different inputs are concatenated along the channel dimension before being fed into the corresponding model. For the baseline variant with HD map information $\mathbf{M}_i$, the HD map is represented as a four-channel grid (normalized centerline direction x and y, lane border type, drivable space). Each variant is evaluated using the same protocol: We use early stopping after three epochs without an improvement in the AP metric on the validation split. The checkpoint with the highest AP on the validation split is then used for all further evaluations.

D. Quantitative Results

Table I shows the quantitative results of our approach applied to the two prediction models, namely the U-Net and the DAE-Former. The completely map-free variant and the HD map-reliant variant of the U-Net and the DAE-Former are used for comparison.

As shown in the third line of the table, even without employing the masking strategy described in Section IV-B, our approach using the latent behavior prior already shows significant improvements compared to the map-free variant. For the U-Net, our approach even outperforms the U-Net using the HD map in all metrics. This indicates that the behavior prior is able to capture information that goes beyond the geometry and semantics present in the HD map, e.g., information about common behavioral patterns. Compared to the U-Net, the DAE-Former achieves overall better results in all metrics. One obvious reason for this is the higher learning capacity due to the higher number of learnable model parameters. While the DAE-Former also significantly benefits from using the behavior prior, it does not outperform the variant with HD map. We hypothesize that the DAE-Former has more difficulties with noisy updates of the latent grid during learning, as it is an architecture that does not rely on convolutions. In contrast to that, the convolutional layers in the U-Net potentially have a filtering effect.

The results of our approach enhanced with the masking strategy, as shown in the last line of Table I, confirm this hypothesis: Using the masking strategy leads to significant improvements for the DAE-Former, especially in terms of Soft IoU ($17.53\,\%$ to $18.34\,\%$). The benefit is also visible for the U-Net. The U-Net variant that uses the masking strategy is the overall best U-Net in all metrics.

E. Qualitative Results

1) Prediction Results: Fig. 3 shows qualitative results of the U-Net (Sample 1) and the DAE-Former (Sample 2) map-free, with HD map, and with the latent behavior prior. The color codes are explained in the caption of the figure. Sample 1 shows a complex multi-lane intersection. The map-free U-Net variant predicts future occupancy in the border regions of the patch. This is because it is expected that some agents enter the field of view at a future timestep. However, due the lack of knowledge about the lanes and other static entities, the map-free variant predicts this occupancy not only in areas with lanes but also in areas that are not even drivable. The HD map-reliant variant and our approach with the latent behavior prior mainly predict future occupancy in areas that are actually drivable. Another observation is that the map-free variant predicts the topmost agent (indicated by the red box) to mainly go straight (indicated by the green occupancy cone below the red box). Similar to the HD map-reliant variant, our approach mainly predicts that this agent will turn right. In this case, turning right is not only the true future behavior of the agent, it is also the only legal behavior, as there is no lane going straight. Sample 2 confirms the observation that our method enhances prediction quality for the DAE-Former. While the map-free variant predicts blurry occupancy for the intersection in the top left of the field of view, the predictions of the HD map-reliant variant and of our approach mainly adhere to the lanes. This includes straight and turning lanes.

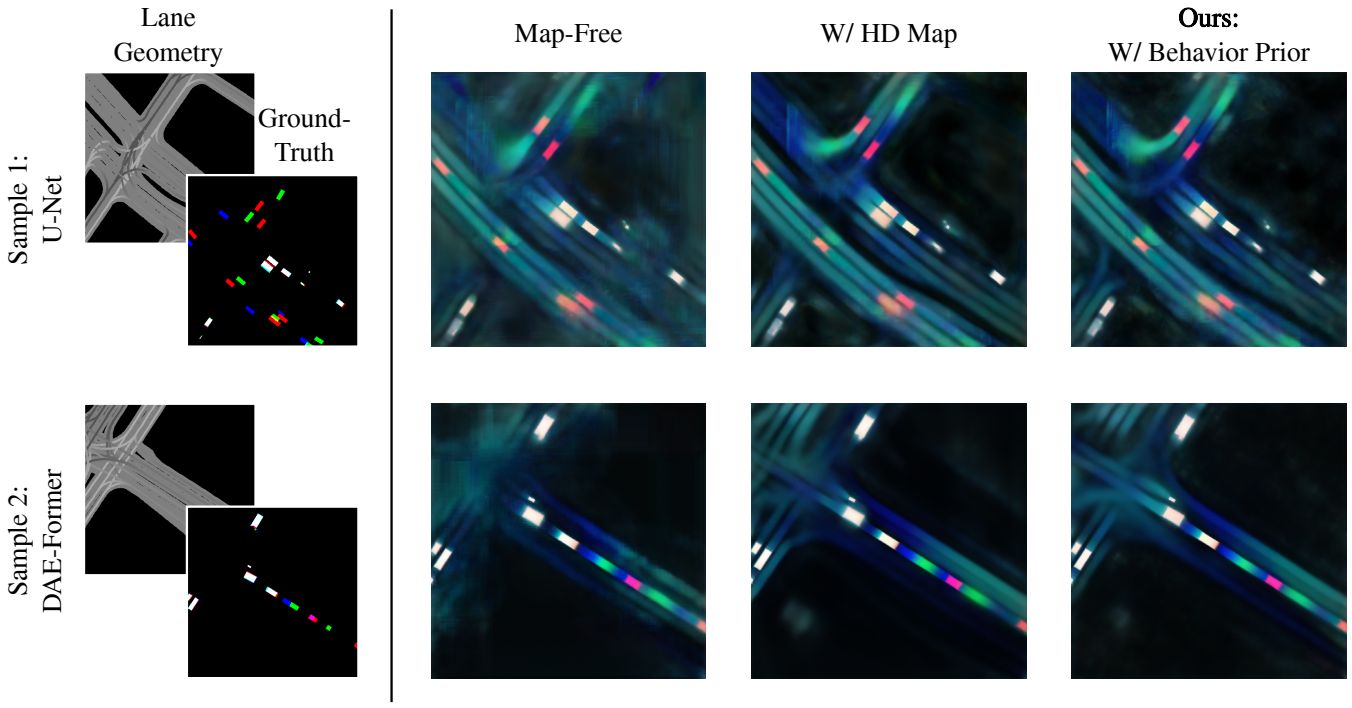


Fig. 3: Qualitative results of the map-free variant, the high definition (HD) map-reliant variant and our approach using the learned latent behavior prior. The colors red ($t = 1$), green ($t = 24$), and blue ($t = 49$) represent the prediction results for one of the $\mathrm{T}_{\mathrm{fut}}$ predicted timesteps. Probabilities are rescaled to a natural logarithmic scale, allowing to better highlight visual differences. To avoid negatively infinite values, an offset of 0.1 is added to all probabilities before rescaling. Areas that appear white have a similar probability o for each timestep t assigned. This is mainly caused by agents being predicted as static.

2) Visualizing the Latent Behavior Prior: Fig. 4 visualizes the rasterized HD map and the learned latent prior, i.e., the local latent grid, for two exemplary data samples. The $\mathrm{C} = 64$ channels are reduced to an RGB image using principal component analysis. In Sample 1, the curved road is recognizable in the latent prior learned by the U-Net and the DAE-Former. The side roads are not clearly visible, which is most likely caused by only few agents using these roads, leading to only few updates of the latent prior during training. Similar observations also apply to Sample 2. The main road from top right to bottom left is again recognizable, this time more obvious in the latent prior learned by the DAE-Former. In summary, these findings indicate that our approach is able to distill geometric information purely from agent behavior, among other less obvious information that might not be visible in this low-dimensional visualization. We further confirm this finding in the following Section V-E.3.

3) Reconstruction of the High Definition Map From the Latent Behavior Prior: We perform a reconstruction experiment to further analyze the type of information encoded within the latent behavior prior. Specifically, the previously learned prior is utilized as input to a convolutional neural network for predicting the HD map. For this, the dataset's total coverage area is partitioned into distinct training and validation regions. The reconstruction results on the previously unseen validation regions are presented in Fig. 5. The findings indicate that certain components of the HD map can be reconstructed solely from the latent prior, demonstrating that lane-related information is effectively encoded within the latent representation. The geometry of major roads is

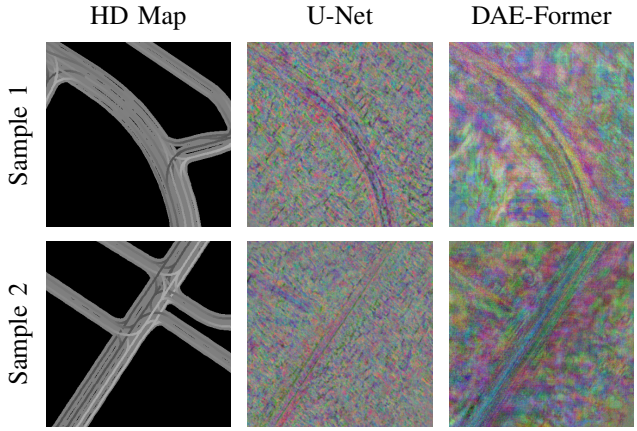


Fig. 4: High definition (HD) map and patches of our latent behavior prior learned by the U-Net and DAE-Former for two exemplary data samples. Principal component analysis is applied to the latent priors as dimensionality reduction.

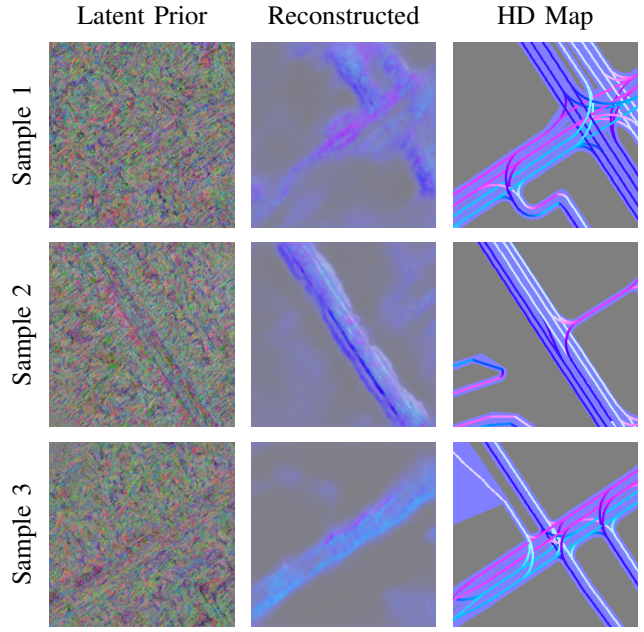


Fig. 5: A convolutional neural network is trained to reconstruct the high definition (HD) map (third column) using solely the learned latent behavior prior (first column) as input, demonstrating that it encodes similar geometric and semantic information as the HD map itself. The map coloring represents the lane direction. Major roads, which are observed by a larger number of agents can be reconstructed at their correct locations from the latent behavior prior.

predicted with high accuracy, whereas smaller side roads are often omitted. A similar observation can be made regarding the semantic information about lane directions, as indicated by the coloring. This phenomenon is expected, as the number of observed agents serves as a crucial factor in encoding lane information. Since fewer agents are typically present on smaller roads, the model receives less information to accurately reconstruct these regions.

F. Ablation Studies

Table II analyzes the influence of the latent prior's capacity, i.e., number of latent grid channels C. The results indicate that C is a model-specific hyperparameter. Depending on the model, even $\mathrm{C} = 1$ can already lead to significant improvements compared to using no map at all. Indeed, the U-Net with $\mathrm{C} = 1$ outperforms the $\mathrm{C} = 32$ and even the $\mathrm{C} = 64$ variant in Soft IoU and IoU.

Lastly, we want to further confirm that the superior performance of our models using the latent behavior prior is due to the storage of location-specific information, rather than to the presence of additional learnable parameters. For this, we evaluate training a U-Net that uses the same learnable grid $\overline{\mathbf{G}}$ for all locations, instead of location-specific grids $\mathbf{G}_{\mathrm{loc},i}$ that get extracted from a larger global grid $\mathbf{G}_{\mathrm{glob}}$. In other words, we add a single large learnable parameter to the input of the U-Net. $\mathrm{C} = 64$ is chosen for this experiment. The results

TABLE II: Ablation study on the latent behavior prior capacity C. U-Net and DAE-Former prediction models are used, results on the testing split.

C	U-Net			DAE-Former		
	AP [%] ↑	Soft IoU [%] ↑	IoU [%] ↑	AP [%] ↑	Soft IoU [%] ↑	IoU [%] ↑
64	75.19	14.76	55.88	80.37	17.53	60.85
32	75.22	14.99	55.88	78.92	15.35	58.90
1	75.02	15.15	56.26	79.66	16.44	59.68

TABLE III: Ablation study between a single location-independent learnable parameter $\overline{\mathbf{G}}$ and our location-specific latent behavior prior $\mathbf{G}$. U-Net prediction model is used, results on the testing split.

	AP [%] ↑	Soft IoU [%] ↑	IoU [%] ↑
$\mathbf{G}$	75.19	14.76	55.88
$\overline{\mathbf{G}}$	70.38	12.71	52.14

in Table III confirm that the additional learnable parameters alone are not the reason why our approach outperforms even the U-Net with HD map. What really matters is that the learnable parameters, i.e., the behavior priors, are location-specific, allowing to capture the nuances in behavior that are different for each location.

VI. Conclusion

We propose location-specific latent behavior priors, a novel data-driven way to aggregate georeferenced information, as an alternative to HD maps in occupancy prediction. Our learned approach achieves comparable or better final task performance than using expensive HD map information, while not requiring any additional labels beyond the already available future agent locations. We analyze the latent grid and find that it efficiently stores geometric and semantic information even with small capacities.

Based on the findings of this paper, future work should investigate how our approach transfers to tasks beyond prediction. Particularly interesting is the applicability to end-to-end automated driving systems, as their task also comes down to reasoning about agent behavior that is potentially location-specific.

References

[1] G. Elghazaly, R. Frank, S. Harvey, and S. Safko, "High-Definition Maps: Comprehensive Survey, Challenges and Future Perspectives," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 4, Jan. 2023.

[2] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "VectorMapNet: End-to-end Vectorized HD Map Learning," in *Proceedings of the 40th International Conference on Machine Learning*. PMLR, July 2023, pp. 22 352–22 369.

[3] T. Yuan, Y. Liu, Y. Wang, Y. Wang, and H. Zhao, "StreamMapNet: Streaming Mapping Network for Vectorized Online HD Map Construction," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2024, pp. 7341–7350.

[4] S. Yi, H. Li, and X. Wang, "Pedestrian Behavior Understanding and Prediction with Deep Neural Networks," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 263–279.

[5] I. Gilitschenski, G. Rosman, A. Gupta, S. Karaman, and D. Rus, "Deep Context Maps: Agent Trajectory Prediction Using Location-Specific Latent Maps," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5097–5104, Oct. 2020.

[6] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, Dec. 2016.

[7] E. Parisotto and R. Salakhutdinov, "Neural Map: Structured Memory for Deep Reinforcement Learning," in *International Conference on Learning Representations*, 2018.

[8] S. Gupta, V. Tolani, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive Mapping and Planning for Visual Navigation," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1311–1330, May 2020.

[9] X. Xiong, Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "Neural Map Prior for Autonomous Driving," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 17 535–17 544.

[10] Z. Xie, Z. Pang, and Y.-X. Wang, "MV-Map: Offboard HD Map Generation with Multi-view Consistency," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 8624–8634.

[11] J. Schmidt, P. Huissel, J. Wiederer, J. Jordan, V. Belagiannis, and K. Dietmayer, "RESET: Revisiting Trajectory Sets for Conditional Behavior Prediction," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, June 2023.

[12] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 11 522–11 530.

[13] 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*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Springer International Publishing, 2020, pp. 541–556.

[14] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal Motion Prediction with Stacked Transformers," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7573–7582.

[15] T. Monninger, J. Schmidt, J. Rupprecht, D. Raba, J. Jordan, D. Frank, S. Staab, and K. Dietmayer, "SCENE: Reasoning About Traffic Scenes Using Heterogeneous Graph Neural Networks," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1531–1538, Mar. 2023.

[16] J. Schmidt, J. Jordan, F. Gritschneder, and K. Dietmayer, "CRAT-Pred: Vehicle Trajectory Prediction with Crystal Graph Convolutional Neural Networks and Multi-Head Self-Attention," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 7799–7805.

[17] Y. Hou, X. Zhang, H. Zhang, X. Cao, Z. Lu, and X. Yuan, "Vehicle Trajectory Prediction Model for Map-Free Scenes Using the Spatio-temporal Attentional Mechanism," *IEEE Internet of Things Journal*, 2024.

[18] J. Schmidt, J. Jordan, F. Gritschneder, T. Monninger, and K. Dietmayer, "Exploring Navigation Maps for Learning-Based Motion Prediction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 3539–3545.

[19] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-Term Occupancy Grid Prediction Using Recurrent Neural Networks," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 9299–9305.

[20] H. Liu, Z. Huang, and C. Lv, "Multi-modal Hierarchical Transformer for Occupancy Flow Field Prediction in Autonomous Driving," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 1449–1455.

[21] B. Ferenczi, M. Burke, and T. Drummond, "MotionPerceiver: Real-Time Occupancy Forecasting for Embedded Systems," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2822–2829, Mar. 2024.

[22] A. Miles, J. Kirkham, M. Durant, J. Bourbeau, T. Onalan, J. Hamman, Z. Patel, shikharsg, M. Rocklin, r. dussin, V. Schut, E. S. d. Andrade, R. Abernathey, C. Noyes, sbalmer, p. i. bot, T. Tran, S. Saalfeld, J. Swaney, J. Moore, J. Jevnik, J. Kelleher, J. Funke, G. Sakkis, C. Barnes, and A. Banihirwe, "zarr-developers/zarr-python: v2.4.0," Jan. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3773450

[23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[24] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *International Conference on Learning Representations*, 2019.

[25] 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 *Proceedings of the 2020 Conference on Robot Learning*. PMLR, Oct. 2021, pp. 409–418.

[26] G. Máttyus, W. Luo, and R. Urtasun, "DeepRoadMapper: Extracting Road Topology From Aerial Images," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3438–3446.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Springer International Publishing, 2015, pp. 234–241.

[28] R. Azad, R. Arimond, E. K. Aghdam, A. Kazerouni, and D. Merhof, "DAE-Former: Dual Attention-Guided Efficient Transformer for Medical Image Segmentation," in *Predictive Intelligence in Medicine*, I. Rekik, E. Adeli, S. H. Park, C. Cintas, and G. Zamzmi, Eds. Springer Nature Switzerland, 2023, pp. 83–95.