Deep Occupancy-Predictive Representations for Autonomous Driving

Eivind Meyer, Lars Frederik Peiss, and Matthias Althoff

***Abstract*— Manually specifying feature spaces that capture the diversity in traffic environments is impractical. Consequently, learning-based agents cannot realize their full potential as neural motion planners for autonomous vehicles. Instead, this work proposes to *learn* which features are task-relevant. Given its immediate relevance to motion planning, our proposed representation architecture encodes ego-centered lane occupancy as a proxy for actionable representations. By leveraging a map-aware graph formulation of the environment, the learned state representations generalize to arbitrary road networks and traffic situations. We show that our approach significantly improves the downstream performance of a reinforcement learning agent operating in urban environments.**

I. Introduction

Human drivers inherently possess an ability to react to new situations. This is in stark contrast to the narrow operational domains of current reinforcement learning (RL) approaches to autonomous driving, given the prevalence of ad hoc feature spaces associated with poor generalization [1], [2]. In particular, two domain-specific characteristics of autonomous driving render the systematic design of relevant and comprehensive state representations difficult: First, the set of other traffic participants, given its variable size and lack of a canonical ordering, is incompatible with fixed-sized feature spaces. Second, the diversity in road networks in terms of geospatial topology complicates specifying a universal map representation [3].

By adopting graph neural networks (GNNs) as RL policy architectures, recent works have outperformed traditional approaches relying on fixed-sized feature vectors. However, these were confined to homogeneous road network geometries such as highways [4], [5] or roundabouts [6], streamlining the learning environment. On the other hand, GNN architectures that unify traffic and infrastructure have been proposed for the related task of vehicular motion prediction [7]–[15]. However, directly adopting heterogeneous GNNs as policy networks is challenging, as current state-of-the-art RL algorithms cannot be reliably trained in complex environments [16]–[18].

To mitigate the challenging nature of the learning task, our proposed approach is anchored in the core principle of state representation learning (SRL), namely the formulation of a representation objective and the design of a corresponding representation model detached from the RL agent [19]. As opposed to letting the agent directly infer control signals from a multi-modal graph representation, we formalize occupancy prediction as a proxy for environment understanding. Specifically, we develop a GNN-based encoder-decoder model whose intermediate latent states serve as low-dimensional, pre-trained state representations for the RL agent. Notably, the flexible nature of our graph-based encoding architecture allows arbitrary road network topologies and traffic environments to be captured by the learned representations. To alleviate the lossy nature of compressive graph encoding, we employ a novel recurrent decoder architecture that constrains the representation space in accordance with a priori known physical priors.

Department of Informatics, Technical University of Munich, Garching, Germany. {eivind.meyer,lf.peiss,althoff}@tum.de

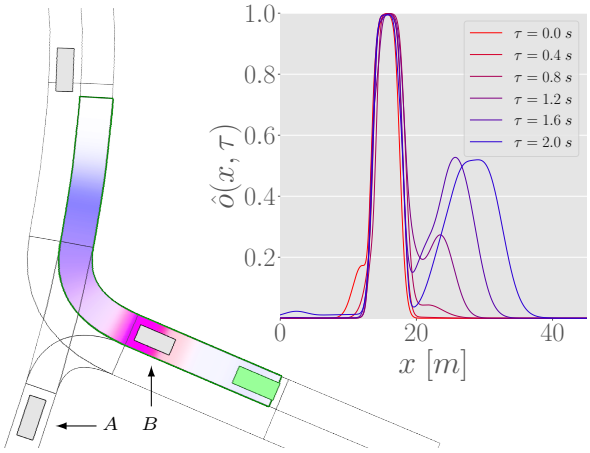


Fig. 1: Our model leverages social patterns to encode lane occupancy $o^{(t)}(x,\tau)$ in an ego-centered coordinate frame (green). Here, the decoded vehicle trajectories (top right) reflect that vehicle B is yielding for A.

II. Related work

We first introduce the learning frameworks used by our approach alongside related applications to motion planning.

A. State representation learning (SRL)

SRL methods have been shown to enhance the performance of RL agents operating in high-dimensional, complex environments [19], [20]. An *agent-centric* approach is generally preferred, so that the learned representations are aligned with the planning context [21]. Further common requirements for representations are to be *predictive* of the future world state [22]–[26] (as opposed to merely *reconstructive* of the present) and *low-dimensional* [27]–[29] (i.e., non-redundant). To mitigate the trade-off between dimensionality reduction and expressiveness, SRL can be supported by incorporating knowledge about the world as *representation priors* [19], [30], [31]. It has been shown that imposing structural constraints on the representations, e.g. by enforcing correspondance to physically plausible world states, improves their generalization performance and downstream effectiveness [32]–[34].

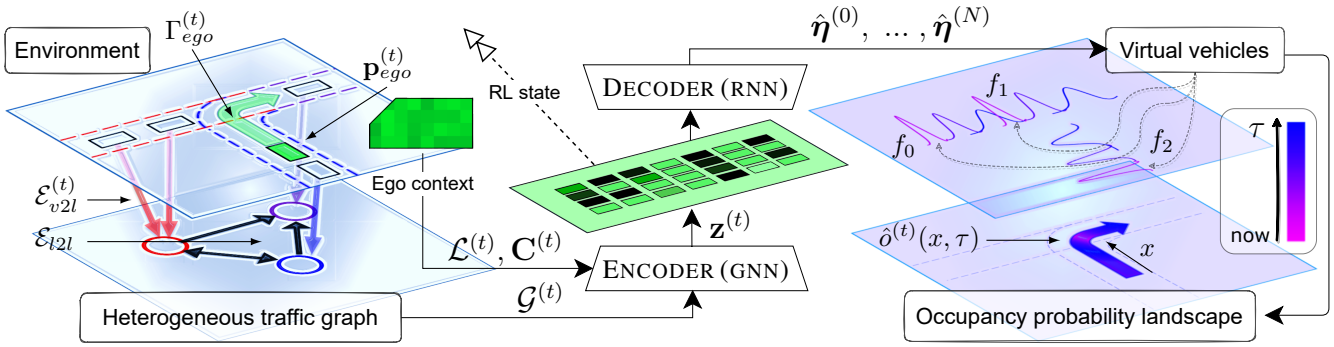


Fig. 2: Overview of our proposed SRL architecture. The pre-trained latent state $\mathbf{z}^{(t)}$ is extracted as input for an RL-based motion planner.

B. Graph neural networks (GNNs)

As the graph-compatible counterpart of traditional encoder architectures, GNNs present a framework for applying SRL to traffic environments. Within the context of the widely adopted *message passing* paradigm [35], GNNs compute neighborhood-aware hidden representations via the permutation-invariant aggregation of *edge messages*, i.e., neural encodings transmitted from a node to its (outgoing) neighbors. This facilitates the propagation of task-relevant information flow across the graph, which, depending on the learning problem, can be summarized on the graph level via readout operations [36]. As necessitated by the multi-modal traffic graph formulation assumed in this work, GNNs are also extendable to heterogeneous graph inputs [37], [38].

C. Applications to motion planning

Autoencoder-based representation models [39] have been used in a multitude of existing works for learning latent states based on e.g. rasterized bird's eye view images [40]–[44] or on-board sensor data [45]. However, they do not leverage the structural biases induced by the road network [46]. In line with our approach, [47] and [48] use GNN-based encoders to learn structurally-aware state representations, but in the context of RL-based robotic manipulators.

Occupancy prediction as a standalone learning task is widely covered in existing works [49]–[53]. With the objective of encoding traffic scenes similar to ours (albeit not in the context of motion planning), encoders for learning representations of occupancy maps have been proposed in [54]–[56]. Using graphical or otherwise spatially-aware encoders similar to ours, recent works such as [57]–[61] predict occupancy grids [62] as an intermediate learning target for guiding the training of neural motion planners. However, these approaches do not provide global, low-dimensional representations appropriate for decoupled RL agents. In contrast to our work, they also suffer from the lossy nature of grid-wise occupancy discretization [63].

III. Methodology

In the following, we outline the details of our architecture, which is illustrated in Fig. 2.

A. Definitions

1) Heterogeneous traffic graph: As originally proposed in [64], we model road networks as decomposable into atomic, interconnected road segments (i.e. *lanelets*). Following a graph-based modeling paradigm, we accordingly let the dynamic traffic environment at time t be formalized by the heterogeneous graph tuple $\mathcal{G}^{(t)} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)}, \mathcal{X}_{\mathcal{V}}^{(t)}, \mathcal{X}_{\mathcal{E}}^{(t)})$, where $\mathcal{V}^{(t)} = (\mathcal{V}_v^{(t)}, \mathcal{V}_l)$ index the vehicle (V) and lanelet (L) nodes, $\mathcal{E}^{(t)} = (\mathcal{E}_{v2l}^{(t)}, \mathcal{E}_{l2l})$ define the corresponding vehicle-to-lanelet (V2L) and lanelet-to-lanelet (L2L) edges, while $\mathcal{X}_{\mathcal{V}}^{(t)} = (\mathbf{X}_v^{(t)}, \mathbf{X}_l)$ and $\mathcal{X}_{\mathcal{E}}^{(t)} = (\mathbf{X}_{v2l}^{(t)}, \mathbf{X}_{l2l})$ contain node and edge-level graph features. Here, the time-dependent V2L edges equate to the physical presence of a vehicle on a given lanelet, while the static L2L edges are implied by the road network topology. The type of topological relationship between adjacent lanelets can be assumed to strongly influence the characteristics of the traffic flow between them. As highlighted in Fig. 3, we therefore distinguish between

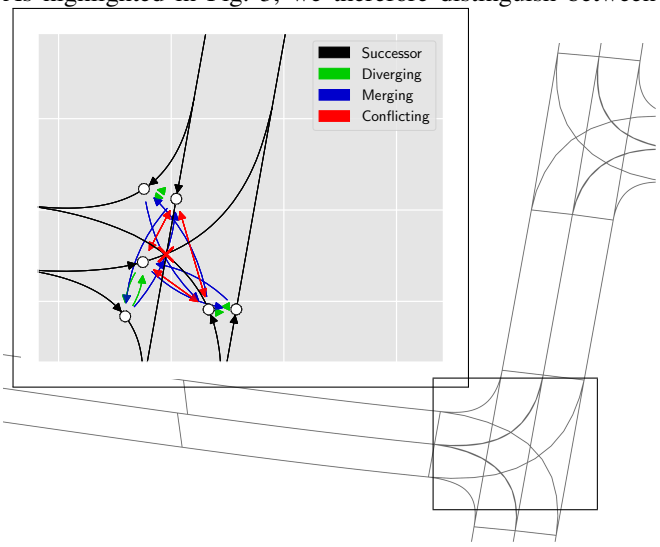


Fig. 3: Lanelet graph example highlighting the topological variation in L2L relationships. The red crosses indicate conflict points.

semantically different L2L edges, assumed to be encoded in $\mathbf{X}_{l2l}$. Furthermore, our approach incorporates other natural feature choices including, but not exclusive to, velocity (V) and heading error (V2L).

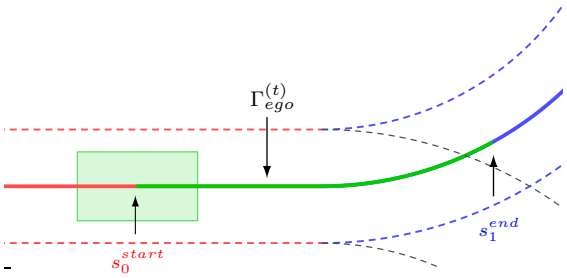


Fig. 4: Ego reference path $\Gamma_{ego}^{(t)}$ composed of two lanelets.

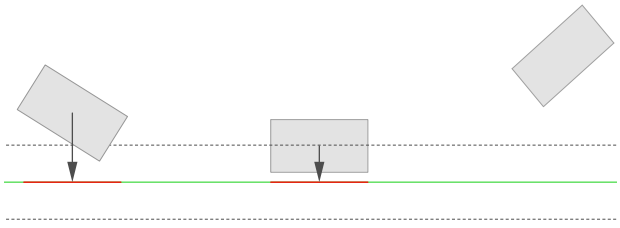


Fig. 5: $o^{(t)}$ is derived from projecting vehicles onto $\Gamma_{ego}^{(t)}$.

2) Planning context: The state of the ego vehicle is set as $(\mathbf{p}_{ego}^{(t)}, v_{ego}^{(t)}, \lambda_{ego})$, referring to its center position, speed, and length, respectively. Next, we let the ego reference path $\Gamma_{ego}^{(t)} : [0, \zeta] \to \mathbb{R}^2$ of length ζ be parameterized by arclength s, and impose the natural constraint that $\Gamma_{ego}^{(t)}(0) = \mathbf{p}_{ego}^{(t)}$, and that it follows the centerline of a connected, traffic-compliant sequence of lanelets in $\mathcal{V}_l$. The corresponding sequence of node indices is denoted by $\mathcal{L}^{(t)}$. Further, we let s_i^{start} and s_i^{end} be the start- and endpoint coordinates of the i^{th} element, as defined within the arclength-parameterized coordinate frame of the centerlines. As seen in Fig. 4, this lets us define the ego-centered planning context in continuous terms. Finally, we let d_i denote lanelet length, and let d_i^{prior} be the cumulative length of $\Gamma_{ego}^{(t)}$ prior to i, i.e., its topological start offset. Then, we let the context matrix $\mathbf{C}^{(t)}$ contain the row vectors $\mathbf{c}_i = \left[s_i^{start},\, s_i^{end},\, d_i,\, d_i^{prior}\right]$. In our encoder pipeline, $\mathbf{C}^{(t)}$ is leveraged for attentional message aggregation [65], which is necessary for conditioning the final encodings on the spatial context of the ego vehicle.

B. Occupancy as representation objective

As occupancy explicitly expresses drivable and non-drivable space, it can be considered as the foundational environment characteristic in a motion planning context [66]. In the following, we consider occupancy in the longitudinal lane domain, as this is the most relevant for driving and simplifies our modelling assumptions. Given the coordinate tuple (s, τ), where s maps to a position on $\Gamma_{ego}^{(t)}$ and τ is a time offset relative to t, we accordingly let ground-truth occupancy be defined by $o^{(t)} : [0, \zeta] \times \mathbb{R} \to \{0, 1\}$. As shown in Fig. 5, this is derived from a path projection of the vehicles that overlap with the road surface.

C. Encoder architecture

Our encoding pipeline is denoted by

$$\mathbf{z}^{(t)} = \textsc{Encoder}(\mathcal{G}^{(t)}; \mathcal{L}^{(t)}, \mathbf{C}^{(t)}), \tag{1}$$

with $\mathbf{z}^{(t)} \in \mathbb{R}^Z$ being the final output of a multilayered message passing and aggregation procedure. The GNN-based encoder is designed to facilitate the probabilistic propagation of traffic flow across the given lanelet network, before returning graph-level outputs based on a vectorized aggregation of the lanelet nodes' hidden states. In the following, we use $\Theta_\square$ to denote trainable, non-linear functions, Σ to denote a permutation-invariant aggregation operation, and assume the activation functions $\rho_\square$ to be applied after each layer.

1) Vehicle-to-lanelet: Unlike the vehicle-to-vehicle paradigm proposed in e.g. [5], we capture V2V interaction effects by embedding their joint presence onto the lanelet graph, enabling the subsequent context-aware message passing layers to model social effects more precisely. Letting lanelet and vehicle nodes be indexed by i and j, respectively, we compute initial lanelet-level encodings $\mathbf{h}_{i,0}^{(t)} \in \mathbb{R}^H$ as

$$\mathbf{h}_{i,0}^{(t)} = \Theta_l(\mathbf{x}_i) + \sum_{\forall j \in \mathcal{V}_v^{(t)} \,|\, (j,i) \in \mathcal{E}_{v2l}^{(t)}} \Theta_{v2l}([\mathbf{x}_i, \mathbf{x}_j^{(t)}, \mathbf{x}_{ij}^{(t)}]),$$

where $(\mathbf{x}_i,\, \mathbf{x}_j^{(t)})$ and $\mathbf{x}_{ij}^{(t)}$ are the node and edge features.

2) Lanelet-to-lanelet: Next, L successive message passing layers are used for facilitating the propagation of traffic dynamics across the road network. With l denoting the layer index, we recursively update the hidden lanelet states according to

$$\mathbf{h}_{i,l+1}^{(t)} = \mathbf{h}_{i,l}^{(t)} + \sum_{\forall j \in \mathcal{V}_l \,|\, (j,i) \in \mathcal{E}_{l2l}} \Theta_{l2l}([\mathbf{h}_{i,l}^{(t)}, \mathbf{x}_{ij}^{(t)}]),$$

3) Ego-attentional readout: Subsequently, an aggregation layer is used to obtain the graph-level hidden states $\mathbf{h}_{ego}^{(t)} \in \mathbb{R}^H$. To satisfy our need for the representations being ego-conditioned, the aggregation of $\mathbf{h}_{i,L}^{(t)}$ is weighted by attention scores $\boldsymbol{\alpha}^{(t)}$ computed from $\mathbf{C}^{(t)}$. Specifically, we let

$$\boldsymbol{\alpha}^{(t)} = \text{softmax}\left(\Theta_c(\mathbf{C}^{(t)})\right), \tag{2}$$

$$\mathbf{h}_{ego}^{(t)} = \sum_{i \in \mathcal{L}^{(t)}} \alpha_i^{(t)} \mathbf{h}_{i,L}^{(t)}. \tag{3}$$

4) Bottleneck layer: We obtain the final low-dimensional latent representations $\mathbf{z}^{(t)}$ by applying the downscaling layer

$$\mathbf{z}^{(t)} = \Theta_z(\mathbf{h}_{ego}^{(t)}). \tag{4}$$

D. Decoder architecture

The decoding objective is to map $\mathbf{z}^{(t)}$ to a probabilistic parameterization of the occupancy landscape according to

$$\begin{aligned} o^{(t)}(s,\tau) &\sim \text{Bernoulli}(\hat{o}^{(t)}(s,\tau)), \\ \hat{o}^{(t)}(s,\tau) &= \textsc{Decoder}(s, \tau;\, \mathbf{z}^{(t)}). \end{aligned} \tag{5}$$

However, as an abstraction of something tangible (i.e., the presence of vehicles), directly predicting occupancy based on $[\mathbf{z}^{(t)}, s, \tau]$ prohibits us from exploiting the physical priors of our application domain. As a result, the lack of constraints imposed on the decoder might lead to overparameterized predictions that are inconsistent with the data [67]. By instead operating on the vehicle domain, our decoding approach imposes architecturally-enforced safeguards against nonsensical occupancy expressions. As an example, this entails conformance to plausible limits for vehicle lengths.

1) Virtual vehicles: Assuming future occupancy probability to be a multi-modal manifestation of possible vehicle trajectories, we model it by superimposing recurrently decoded *virtual vehicles*, i.e., non-deterministic phantom vehicles that are constrained in their occupancy expression from physical priors.

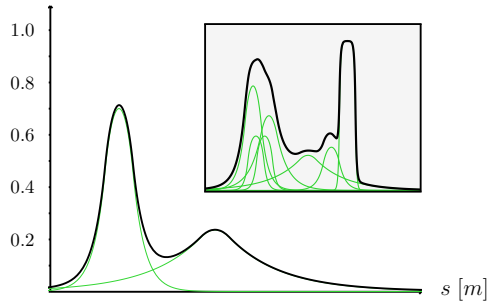


Fig. 6: Spatial cross-section of the joint occupancy probability footprint $\hat{o}^{(t)}(s)$ (black) induced by a set of two virtual vehicles. In the top right, more virtual vehicles are added for comparison.

As illustrated by Fig. 6, their stochastic formulation allows a differentiable and permutation-invariant expression for the joint occupancy footprint. By translating the learning task to the global domain, this circumvents the otherwise difficult set prediction task [68]–[71]. Formally, we let the state of a virtual vehicle q be defined by the three-tuple $(\lambda_q, \mathcal{I}_q, s_q)$, where $\lambda_q \in \mathbb{R}$ denotes its length, $\mathcal{I}_q \in \{0, 1\}$ is an existence indicator, and $s_q : \mathbb{R} \to [0, \zeta]$ returns its center position at time $t + \tau$ given $\mathcal{I}_q = 1$. Further, we let $o_q : [0, \zeta] \times \mathbb{R} \to \{0, 1\}$ return its occupancy, i.e.,

$$o_q(s, \tau) = \begin{cases} 1 & \text{if} \quad |s - s_q(\tau)| < \frac{\lambda_q}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

2) Stochastic formulation: To facilitate a stochastic model of the virtual vehicles, we let $f_s^{(q)} : \mathbb{R} \times \mathbb{R} \to [0, 1]$ be the time-varying probability density function (PDF) for their position, as parameterized by s. Further, we denote the corresponding cumulative distribution function (CDF) as $F_s^{(q)}$. Notably, their probability domains differ from that of lane occupancy, given that $o^{(t)}$ is formalized as an infinite number of Bernoulli distributions in spatio-temporal space. Unlike $f_s^{(q)}$, occupancy is thus not required to sum up to one for a given τ. To illustrate this, we can consider the extreme cases 1) no traffic (i.e., $o^{(t)}(s) = 0 \; \forall \; s$) and 2) the presence of an infinitely long vehicle (i.e., $o^{(t)}(s) = 1 \; \forall \; s$).

By addressing the accumulation of uncertainty, the *Fokker-Planck* [72] equation is a well-suited stochastic modeling framework for vehicular motion [73]. Since non-linear behavior patterns can be modelled via superimposed virtual vehicles, we assume a simplified dynamics model with linear drift and diffusion parameters. As shown in [74], the corresponding Gaussian solution is

$$f_s^{(q)}(s; \tau, \hat{\eta}_{\mathbf{s}}^{(q)}) = \frac{1}{2\sqrt{\pi \hat{\eta}_{s,1} \tau}} \exp\left(-\frac{(s - \hat{\eta}_{s,0} - \hat{\eta}_{s,2}\tau)^2}{4\hat{\eta}_{s,1}\tau} \right),$$
$$F_s^{(q)}(s; \tau, \hat{\eta}_{\mathbf{s}}^{(q)}) = -\frac{1}{2} \operatorname{erf}\left(\frac{\hat{\eta}_{s,2}\tau + \hat{\eta}_{s,0} - s}{2\sqrt{\hat{\eta}_{s,1}\tau}} \right). \tag{7}$$

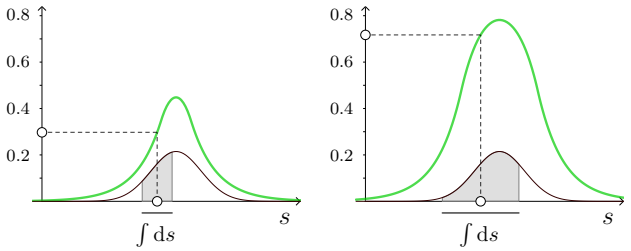
Then, we let a virtual vehicle q be stochastically parameterized by $\hat{\boldsymbol{\eta}}^{(q)} = [\hat{\eta}_i^{(q)}, \hat{\eta}_\lambda^{(q)}, \hat{\eta}_{s,0}^{(q)}, \hat{\eta}_{s,1}^{(q)}, \hat{\eta}_{s,2}^{(q)}]$ according to

$$\begin{aligned} P(\mathcal{I}_q = 1) &= \hat{\eta}_i^{(q)}, \\ P(s_q(\tau) = s \mid \mathcal{I}_q = 1) &= f_s^{(q)}(s; \tau, \hat{\eta}_{\mathbf{s}}^{(q)}), \\ P(o_q(s, \tau) = 1) &= \hat{o}_q(s, \tau), \end{aligned} \tag{8}$$

where $\hat{\eta}_i^{(q)}$ is the predicted existence probability, $\hat{\eta}_\lambda^{(q)}$ is the predicted length and $\hat{\eta}_{\mathbf{s}}^{(q)}$ contains the parameter coefficients for $f_s^{(q)}$. A stochastic application of Eq. 6 gives us that

$$\begin{aligned} \hat{o}_q(s, \tau) &= \hat{\eta}_i^{(q)} \int_{s - \hat{\eta}_\lambda^{(q)}/2}^{s + \hat{\eta}_\lambda^{(q)}/2} f_s^{(q)}(s'; \tau, \hat{\eta}_{\mathbf{s}}^{(q)}) \, \mathrm{d}s' \\ &= \hat{\eta}_i^{(q)} (F(s + \hat{\eta}_\lambda^{(q)}; \tau, \hat{\eta}_{\mathbf{s}}^{(q)}) - F(s - \hat{\eta}_\lambda^{(q)}; \tau, \hat{\eta}_{\mathbf{s}}^{(q)})), \end{aligned} \tag{9}$$

as is exemplified in Fig. 7.



(a) Small virtual vehicle. (b) Large virtual vehicle.

Fig. 7: Spatial cross-section of probabilistic occupancy $\hat{o}_q(s)$ (green) for a single virtual vehicle is derived from $f_s^{(q)}$ and $\hat{\eta}_\lambda^{(q)}$ via a symmetrically bounded integral of length λ_q around s.

E. Joint occupancy probability landscape

To derive a differentiable expression for the global equivalent $\hat{o}^{(t)}(s, \tau)$, we consider a set of N virtual vehicles $\mathcal{Q}^{(t)} = \{q_1^{(t)}, ..., q_N^{(t)}\}$. A reasonable approach is to consider the probability of *at least one* virtual vehicle occupying s at time $t + \tau$. We assume them to be independent, as this is necessary to obtain a tractable expression for the above. This can appear unsound in light of the reactive nature of traffic. However, our framework assumes no association between virtual and actual vehicles, let alone a one-to-one correspondence. Similar to a Gaussian mixture model, they must be viewed as atomic representations of

future trajectories that may or may not materialize. Given a sufficiently large N, the assumption of neither independence nor linearity is prohibitive for the modeling of the joint occupancy landscape. With this in mind, we have that

$$\hat{o}^{(t)}(s,\tau) = 1 - \prod_{q \in \mathcal{Q}^{(t)}} \left(1 - \hat{o}_q(s,\tau)\right). \tag{10}$$

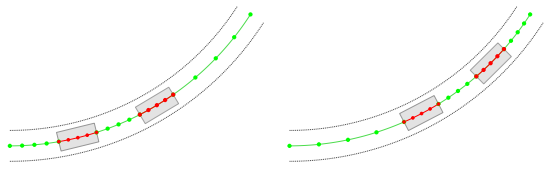
1) Recurrent decoding: We employ a recurrent neural network Θ_{RNN} to decode a fixed number of N parameterizations $\{\hat{\boldsymbol{\eta}}^{(0)}, ..., \hat{\boldsymbol{\eta}}^{(N)}\}$ from $\mathbf{z}^{(t)}$. To enforce physical priors, we let Θ_{RNN} internally transforms $\hat{\boldsymbol{\eta}}^{(q)}$ via rescaled sigmoid functions so as to conform to $\boldsymbol{\eta}_{\min} \leq \hat{\boldsymbol{\eta}}^{(q)} \leq \boldsymbol{\eta}_{\max}$.

F. Spatio-temporal occupancy loss

A naive approach towards defining the loss function is to consider the binary cross-entropy over a grid discretization of the spatio-temporal domain [75]. By leveraging the continuous output domain of our decoder, we instead compute the loss segment-wise in a boundary-aware fashion. We let $\mathcal{O}_p^{(t+\tau)}$ and $\mathcal{O}_n^{(t+\tau)}$ denote the occupied and non-occupied connected components of $\Gamma_{ego}^{(t)}$, respectively. With $\mathcal{P}$ returning the power set, we have that $\mathcal{O}_p^{(t+\tau)} \subseteq \{\Omega \in \mathcal{P}([0,\zeta]) \mid \forall s \in \Omega : o^{(t)}(s,\tau) = 1\}$ and $\mathcal{O}_n^{(t+\tau)} \subseteq \{\Omega \in \mathcal{P}([0,\zeta]) \mid \forall s \in \Omega : o^{(t)}(s,\tau) = 0\}$. By letting $d(\Omega)$ denote path length, we define our decoding loss as

$$\begin{aligned}
\ell_p^{(t)}(\tau) &= \sum_{\Omega \in \mathcal{O}_p^{(t+\tau)}} \frac{1}{d(\Omega)} \int_{s \in \Omega} \log\left(\hat{o}^{(t)}(s,\tau)\right) \mathrm{d}s, \\
\ell_n^{(t)}(\tau) &= \sum_{\Omega \in \mathcal{O}_n^{(t+\tau)}} \frac{1}{d(\Omega)} \int_{s \in \Omega} \log\left(1 - \hat{o}^{(t)}(s,\tau)\right) \mathrm{d}s, \\
\ell^{(t)} &= \int_{\tau=0}^{T} \delta_\ell^\tau \left(\ell_p^{(t)}(\tau) + \ell_n^{(t)}(\tau)\right) d\tau,
\end{aligned} \tag{11}$$

where $T \in \mathbb{R}$ is the considered time horizon, $\delta_\ell \in [0,1]$ is a discount factor for reflecting the diminishing significance of future occupancy and the normalization factor $1/d(\Omega)$ is introduced to avoid the dependence on segment length and to counteract the class imbalance of occupancy. As they are otherwise intractable, the spatial integrals in Eq. III-F are approximated numerically with resolution R_ℓ as illustrated by Fig. 8. Similarly, our implementation discretizes the temporal integral over T_D timesteps.



(a) $\tau = 0.0\,\mathrm{s}$. (b) $\tau = 0.1\,\mathrm{s}$.

Fig. 8: Evaluation of $\ell^{(t)}$ over a constant reference path for two subsequent timesteps τ. The segment-wise loss integrals are approximated numerically based on R_ℓ evenly spaced samples. For illustrative purposes, we set $R_\ell = 5$.

IV. Experiment

A. Dataset

By leveraging the OpenStreetMap[1] API, our dataset is comprised of a diverse set of urban locations sampled from within the Munich metropolitan region. The collected road networks are then populated with vehicles using the traffic simulator SUMO [76]. The dataset contains 1000 simulated scenarios, jointly amounting to 28 hours of traffic.

B. Reinforcement learning agent

For our experiment, we extract $\mathbf{z}^{(t)}$ as inputs for a PPO-based [77] RL agent together with the velocity of the ego vehicle $v_{ego}^{(t)}$. At every timestep t, the encoder is conditioned on the navigation context generated by a high-level route planner. We intentionally limit the agent to longitudinal acceleration control along the given navigation path to focus on the effect of the learned representations. Corresponding to $\Gamma_{ego}^{(t)}$, these plans span across multiple lanelets and include heterogeneous map structures such as intersections. Given the weights $\mathbf{w}_r \in \mathbb{R}^4$, the reward $r^{(t)} \in \mathbb{R}$ is defined as $r^{(t)} = \mathbf{w}_r \cdot [r_p^{(t)}, r_c^{(t)}, r_v^{(t)}, r_{cv}^{(t)}]$, where $r_p^{(t)}$ is a dense path progression reward, $r_c^{(t)}$ is a sparse penalty imposed on collisions, $r_v^{(t)}$ is a linear over-speeding penalty, while $r_{cv}^{(t)}$ penalizes expected occupancy conflict as computed by

$$r_{cv}^{(t)} = \int_{\tau=0}^{T} \delta_{cv}^\tau \int_{\tilde{s}^{(t)}(\tau)-\lambda_{ego}/2}^{\tilde{s}^{(t)}(\tau)+\lambda_{ego}/2} \hat{o}^{(t)}(s,\tau)\, \mathrm{d}s\, \mathrm{d}\tau, \tag{12}$$

with $\tilde{s}^{(t)}(\tau) = \tau v_{ego}^{(t)}$ being a constant-velocity extrapolation of the ego position and $\delta_{cv} \in [0,1]$ being the discount factor.

C. Baseline approaches

We compare the RL performance against multiple baselines trained with identical reward configurations:

1) **V2V**: The vehicle-to-vehicle (i.e., not map-aware) policy network as proposed in [5].
2) **V2L**: An adoption of our Encoder architecture as policy network (i.e., without pre-training).
3) **MLP**: State encodings trained using the feed-forward network Θ_{MLP} for decoding $\hat{o}^{(t)}(s,\tau)$ from $[\mathbf{z}^{(t)}, s, \tau]$.

D. Implementation and training

Our implementation was built with PyTorch [78], and further uses the PyTorch Geometric extension [79] for the GNN-based encoding layers. The PPO implementation from Stable Baselines 3 [80] was used for RL experiments. Parameters for our implementation setup are given in Table I.

The model training was conducted on an NVIDIA A100 Tensor Core GPU for 48 hours using the Adam [81] optimizer. Subsets of the collected dataset were used for training (90%) and test (10%) purposes. To aid generalization, planning contexts for the model were resampled for each mini-batch. The RL agents were trained for 10^6 steps using replays of the collected scenarios, with the the start and goal positions being randomly sampled for each episode.

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

TABLE I: Selected hyperparameters for our experiments[a].

Encoding dimensions (H, Z)	256, 32
Number of L2L layers (L)	4
Aggregation method (Σ)	max
Activation functions ($\rho_\square$)	tanh
Encoding layers (Θ_{v2l}, Θ_{l2l}, Θ_c, Θ_z)	Linear
Recurrent decoder network (Θ_{RNN})	LSTM(256)
Decoded virtual vehicles (N)	12
Planning horizon (T, T_D, ζ)	2.4 s, 60, 45 m
Integration method (see Fig. 8)	Trapezoidal (R_ℓ = 40)
Discount factors (δ_ℓ, δ_{cv})	0.99, 0.95
Baseline decoder (Θ_{MLP})	MLP(256, 128)

[a]We refer to our implementation for further details, including the feature choices ($\mathcal{X}_\mathcal{V}^{(t)}$, $\mathcal{X}_\mathcal{E}^{(t)}$) and the decoding constraints ($\boldsymbol{\eta}_{\text{lb}}$, $\boldsymbol{\eta}_{\text{ub}}$): `https://gitlab.lrz.de/tum-cps/commonroad-geometric/`

V. Results

The occupancy decoding loss and downstream RL success rate are presented in Table II. As is evident, our proposed model achieves better decoding performance than the unconstrained baseline. This indicates that our approach mitigates the effects of the information bottleneck caused by the encoding process. Specifically, it is likely that our physically constrained hypothesis space streamlines the training process, and with that eases the encoder's task of minimizing the bottleneck-induced information loss. Further, a qualitative assessment of the decoded probability landscapes shown in Fig. 1 and Fig. 9 suggests that our model is able to accurately predict the environment to a degree where the corresponding latent encodings can be assumed to support intelligent planning decisions.

TABLE II: Empirical evaluation results

(a) SRL decoding loss

model	ℓ (train)	ℓ (test)
Ours	**1.045**	**0.995**
MLP	1.210	1.289

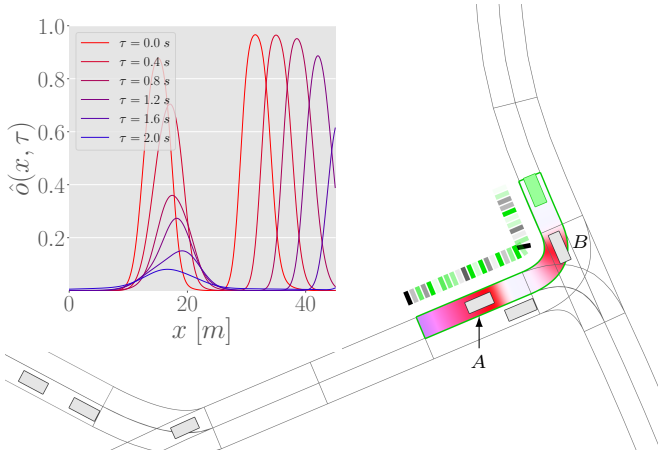
(b) RL performance

agent	goal reach (%)
Ours	**72.9**
V2V-Graph	39.9
V2L-Graph	49.0
MLP	54.0

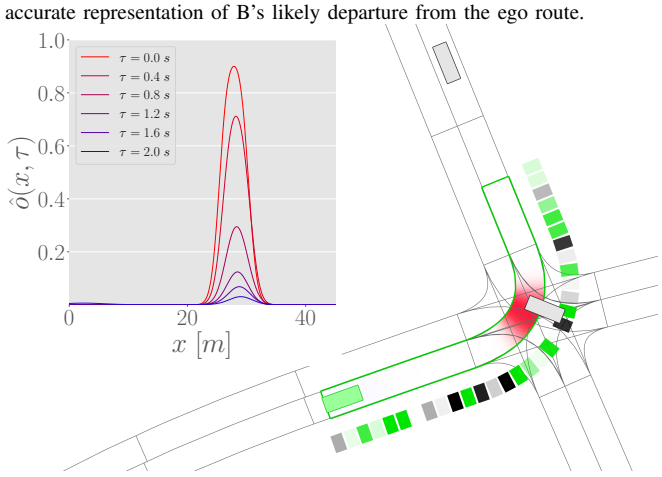
The effectiveness of our representations is confirmed by the results of the RL experiments. The representation-enhanced agent's improved success rate suggests that the pre-trained representations simplify its motion planning task: effectively, they free the agent from the responsibility of modeling its own surroundings, allowing it to concentrate its learning efforts on the lower-level control aspects of motion planning. As traffic modeling is a complex endeavour that is more easily tackled in a supervised setting, it is thus not surprising that the simplification of the RL task leads to better final performance.

VI. Conclusion

In this work, we have developed a neural encoder-decoder architecture for learning latent state representations for autonomous driving. Facilitated by a graph neural network-based encoder employed for compressing the ego vehicle's surroundings, our approach offers significant benefits



(a) The rapid intensity decay of the leftmost probability mode suggests an accurate representation of B's likely departure from the ego route.



(b) Vehicle from another lane intersecting $\Gamma_{ego}^{(t)}$. The occupancy probability map has a transient peak located at the intersection point.

Fig. 9: Decoded occupancy probability landscapes $\hat{o}^{(t)}(x,\tau)$ visualized together with the corresponding encodings $\mathbf{z}^{(t)}$.

compared to previous works: as opposed to feature spaces narrowly tailored to operating in specific traffic settings, the validity of our learned representations naturally extends to arbitrary environments. By conditioning on the underlying road network topology, our encoding framework possesses a demonstrated ability to improve the performance of RL-based motion planning in heterogeneous driving environments.

Acknowledgements

This research was funded by the German Research Foundation grant AL 1185/7-1 and the German Federal Ministry for Digital and Transport in the project KoSi.

References

[1] B. R. Kiran, I. Sobh, V. Talpaert, *et al.*, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2022.

[2] A. Tampuu, T. Matiisen, M. Semikin, *et al.*, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1364–1384, 2022.

[3] B. Jiang, "A topological pattern of urban street networks: Universality and peculiarity," *Physica A: Statistical Mechanics and its Applications*, vol. 384, no. 2, pp. 647–655, 2007.

[4] M. Huegle, G. Kalweit, M. Werling, *et al.*, "Dynamic interaction-aware scene understanding for reinforcement learning in autonomous driving," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4329–4335, 2020.

[5] P. Hart and A. Knoll, "Graph neural networks and reinforcement learning for behavior generation in semantic environments," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1589–1594.

[6] T. Ha, G. Lee, D. Kim, *et al.*, "Road graphical neural networks for autonomous roundabout driving," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 162–167.

[7] M. Liang, B. Yang, R. Hu, *et al.*, "Learning lane graph representations for motion forecasting," in *ECCV*, 2020.

[8] W. Zeng, M. Liang, R. Liao, *et al.*, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, 2021, pp. 532–539.

[9] H. Zhao, J. Gao, T. Lan, *et al.*, "Tnt: Target-driven trajectory prediction," in *CoRL*, 2020.

[10] B. Kim, S. Park, S. S. Lee, *et al.*, "Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 631–14 640, 2021.

[11] L. Zhang, P. Li, J. Chen, *et al.*, "Trajectory prediction with graph-based dual-scale context fusion," *CoRR*, vol. abs/2111.01592, 2021. arXiv: 2111.01592.

[12] F. Janjos, M. Dolgov, and J. M. Zöllner, "Starnet: Joint action-space prediction with star graphs and implicit global-frame self-attention," *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 280–286, 2022.

[13] J. Ngiam, V. Vasudevan, B. Caine, *et al.*, "Scene transformer: A unified architecture for predicting future trajectories of multiple agents," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[14] T. Gilles, S. Sabatini, D. Tsishkou, *et al.*, "Gohome: Graph-oriented heatmap output for future motion estimation," in *International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9107–9114.

[15] X. Mo, Z. Huang, Y. Xing, *et al.*, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9554–9567, 2022.

[16] P. Henderson, R. Islam, P. Bachman, *et al.*, "Deep reinforcement learning that matters," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

[17] L. Buşoniu, T. d. Bruin, D. Tolić, *et al.*, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018, 00194.

[18] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, *et al.*, "Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis," *Machine Learning*, vol. 110, no. 9, pp. 2419–2468, 2021.

[19] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, *et al.*, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, 2018.

[20] J. Munk, J. Kober, and R. Babuška, "Learning state representation for deep actor-critic control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 4667–4673.

[21] S. Parisi, S. Ramstedt, and J. Peters, "Goal-driven dimensionality reduction for reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4634–4639.

[22] B. Boots, S. M. Siddiqi, and G. J. Gordon, "Closing the learning-planning loop with predictive state representations," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 954–966, 2011. eprint: https://doi.org/10.1177/0278364911404092.

[23] Z. D. Guo, B. A. Pires, B. Piot, *et al.*, "Bootstrap latent-predictive representations for multitask reinforcement learning," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[24] K.-H. Lee, I. Fischer, A. Z. Liu, *et al.*, "Predictive information accelerates learning in rl," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[25] S. Recanatesi, M. Farrell, G. Lajoie, *et al.*, "Predictive learning as a network mechanism for extracting low-dimensional latent space representations," *Nature Communications*, vol. 12, 2021.

[26] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 2455–2467.

[27] A. Nouri and M. L. Littman, "Dimension reduction and its application to model-based exploration in continuous spaces," en, *Machine Learning*, vol. 81, no. 1, pp. 85–98, 2010, 00040.

[28] B. Prakash, M. Horton, N. R. Waytowich, *et al.*, "On the use of deep autoencoders for efficient embedded reinforcement learning," in *Proceedings of the 2019 on Great Lakes Symposium on VLSI*, 2019, pp. 507–512.

[29] R. Bassily, S. Moran, I. Nachum, *et al.*, "Learners that use little information," in *Proceedings of Algorithmic Learning Theory*, 2018, pp. 25–55.

[30] R. Jonschkowski and O. Brock, "Learning state representations with robotic priors," *Auton. Robots*, vol. 39, no. 3, pp. 407–428, 2015.

[31] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[32] J. Scholz, M. Levihn, C. L. I. Jr., *et al.*, "A physics-based model prior for object-oriented mdps," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014, pp. 1089–1097.

[33] R. Stewart and S. Ermon, "Label-free supervision of neural networks with physics and domain knowledge," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 2576–2582.

[34] R. Jonschkowski, R. Hafner, J. Scholz, *et al.*, "Pves: Position-velocity encoders for unsupervised learning of structured state representations," *CoRR*, vol. abs/1705.09805, 2017. arXiv: 1705.09805.

[35] J. Gilmer, S. S. Schoenholz, P. F. Riley, *et al.*, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2017, pp. 1263–1272.

[36] J. Zhou, G. Cui, Z. Zhang, *et al.*, *Graph neural networks: A review of methods and applications*, cite arxiv:1812.08434, 2018.

[37] M. Schlichtkrull, T. N. Kipf, P. Bloem, *et al.*, "Modeling relational data with graph convolutional networks," in *The Semantic Web*, 2018, pp. 593–607.

[38] X. Wang, H. Ji, C. Shi, *et al.*, "Heterogeneous graph attention network," *The World Wide Web Conference*, 2019.

[39] M. Tschannen, O. F. Bachem, and M. Lučić, "Recent advances in autoencoder-based representation learning," in *Bayesian Deep Learning Workshop, NeurIPS*, 2018.

[40] B. Toghi, R. Valiente, R. Pedarsani, *et al.*, "Towards learning generalizable driving policies from restricted latent representations," *CoRR*, vol. abs/2111.03688, 2021. arXiv: 2111.03688.

[41] K. Sama, Y. Morales, N. Akai, *et al.*, "Driving feature extraction and behavior classification using an autoencoder to reproduce the velocity styles of experts," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1337–1343.

[42] J. Zhao, J. Fang, Z. Ye, *et al.*, "Large scale autonomous driving scenarios clustering with self-supervised feature extraction," *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021.

[43] A. Kendall, J. Hawke, D. Janz, *et al.*, "Learning to drive in a day," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8248–8254.

[44] P. Cai, H. Wang, Y. Sun, *et al.*, "Dignet: Learning scalable self-driving policies for generic traffic scenarios with graph neural networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2021, pp. 8979–8984.

[45] J. Dong, S. Chen, Y. Li, *et al.*, "Spatio-weighted information fusion and DRL-based control for connected autonomous vehicles," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[46] P. W. Battaglia, J. B. Hamrick, V. Bapst, *et al.*, *Relational inductive biases, deep learning, and graph networks*, 2018.

[47] F. Zhang, Y. Chen, H. Qiao, *et al.*, "Surrl: Structural unsupervised representations for robot learning," *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[48] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, *et al.*, "Graph networks as learnable physics engines for inference and control," in *International Conference on Machine Learning*, PMLR, 2018, pp. 4470–4479.

[49] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[50] N. Mohajerin and M. Rohani, "Multi-step prediction of occupancy grid maps with recurrent neural networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[51] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-term occupancy grid prediction using recurrent neural networks," *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[52] T. Gilles, S. Sabatini, D. Tsishkou, *et al.*, "Home: Heatmap output for future motion estimation," *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021.

[53] P. Kaniarasu, G. C. Haynes, and M. Marchetti-Bowick, "Goal-directed occupancy prediction for lane-following actors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3270–3276.

[54] O. Rákos, T. Bécsi, S. Aradi, *et al.*, "Learning latent representation of freeway traffic situations from occupancy grid pictures using variational autoencoder," *Energies*, vol. 14, no. 17, 2021.

[55] L. A. Marina, B. Trasnea, T. Cocias, *et al.*, "Deep grid net (dgn): A deep learning system for real-time driving context understanding," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 399–402.

[56] M. Itkina, K. Driggs-Campbell, and M. J. Kochenderfer, "Dynamic environment prediction in urban scenes using recurrent representation learning," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019.

[57] E. Amirloo, M. Rohani, E. Banijamali, *et al.*, "Self-supervised simultaneous multi-step prediction of road dynamics and cost map," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[58] P. Hu, A. Huang, J. M. Dolan, *et al.*, "Safe local motion planning with self-supervised freespace forecasting," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 2021, pp. 12 732–12 741.

[59] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[60] A. Sadat, S. Casas, M. Ren, *et al.*, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," *Lecture Notes in Computer Science*, pp. 414–430, 2020.

[61] R. Mahjourian, J. Kim, Y. Chai, *et al.*, "Occupancy flow fields for motion forecasting in autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5639–5646, 2022.

[62] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[63] M. Ilievski, S. Sedwards, A. Gaurav, *et al.*, *Design space of behaviour planning for autonomous driving*, 2019.

[64] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 420–425.

[65] P. Veličković, G. Cucurull, A. Casanova, *et al.*, "Graph Attention Networks," *International Conference on Learning Representations*, 2018, accepted as poster.

[66] S. Söntges and M. Althoff, "Computing the drivable area of autonomous road vehicles in dynamic road scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1855–1866, 2018.

[67] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Auton. Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[68] Y. Zhang, J. Hare, and A. Prügel-Bennett, "Deep Set Prediction Networks," in *Advances in Neural Information Processing Systems 32*, 2019. eprint: 1906.06565.

[69] S. H. Rezatofighi, V. K. B. G., A. Milan, *et al.*, "DeepSetNet: Predicting Sets with Deep Neural Networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[70] Y. Zhang, J. Hare, and A. Prügel-Bennett, "FSPool: Learning set representations with featurewise sort pooling," 2019. eprint: 1906.02795.

[71] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[72] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry and the natural sciences*, Third. Springer-Verlag, 2004, vol. 13, pp. xviii+415.

[73] J. Hinkel, "Applications of physics of stochastic processes to vehicular traffic problems," Ph.D. dissertation, Citeseer, 2007.

[74] H. Risken, "Fokker-Planck Equation," in *The Fokker-Planck Equation: Methods of Solution and Applications*, 14320, Springer Berlin Heidelberg, 1984, pp. 63–95.

[75] C. M. Bender, P. Emmanuel, M. K. Reiter, *et al.*, "Practical integration via separable bijective networks," in *International Conference on Learning Representations (ICLR)*, 2022.

[76] P. A. Lopez, E. Wiessner, M. Behrisch, *et al.*, "Microscopic Traffic Simulation using SUMO," en, in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 01236, 2018, pp. 2575–2582.

[77] J. Schulman, F. Wolski, P. Dhariwal, *et al.*, "Proximal policy optimization algorithms.," *CoRR*, vol. abs/1707.06347, 2017.

[78] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

[79] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[80] A. Raffin, A. Hill, A. Gleave, *et al.*, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[81] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.