Deep Occupancy-Predictive Representations for Autonomous Driving

Eivind Meyer, Lars Frederik Peiss, and Matthias Althoff

***Abstract*— Manually specifying features 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 architecture encodes the probabilistic occupancy map as a proxy for obtaining pre-trained state representations of the environment. By leveraging a map-aware traffic graph formulation, our agent-centric encoder generalizes 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 traffic 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 for self-driving, due to the prevalence of ad hoc feature vectors 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 variable number and lack of a canonical ordering of other traffic participants is incompatible with fixed-sized feature vectors. 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 policies, 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], simplifying the learning problem. Recently, GNN architectures that unify traffic and infrastructure have been proposed for the related task of vehicle trajectory prediction [7]–[12]. 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 [13]–[15].

To mitigate the challenging nature of the learning task, we instead formulate a representation objective and design a state representation model detached from the RL training loop. As opposed to letting the agent directly infer control signals from a multi-modal graph representation, we use spatio-temporal occupancy map prediction, as depicted in Fig. 1, as a learning proxy for environment *understanding*. As illustrated in Fig. 2, we specifically develop a GNN-based encoder-decoder model whose intermediate latent states serve as low-dimensional, pre-trained state representations. Notably, the flexible nature of our encoder 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 propose a novel occupancy prediction framework that constrains the decoding space in accordance with a priori known physical priors for vehicle motion. We implement our approach using CommonRoad-Geometric (*crgeo*) [16], a PyTorch-based framework offering a standardized graph extraction pipeline for traffic scenarios. Our source code is available at `https://github.com/CommonRoad/crgeo-learning`.

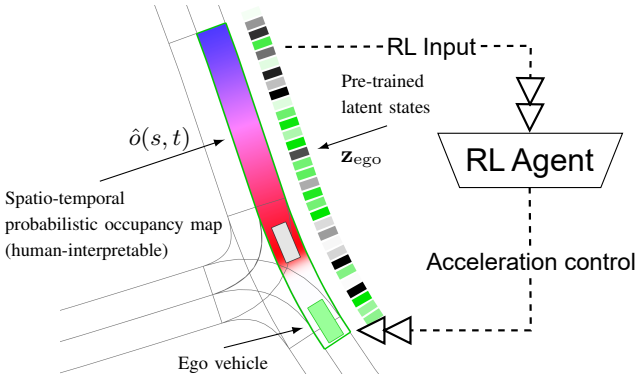


Fig. 1: Our spatio-temporal representation model learns a continuous parameterization of the probabilistic occupancy map $\hat{o}(s,t)$. The red and blue coloring scheme signifies high occupancy probability in the short and long-term future, respectively. The pre-trained intermediate states $\mathbf{z}_{\text{ego}}$ are extracted as inputs for an RL agent controlling the longitudinal acceleration of the ego vehicle.

II. Related work

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

A. State representation learning (SRL)

By learning encoded representations of the surroundings, SRL methods enhance the performance of RL agents operating in high-dimensional, complex environments [17]–[19]. An *agent-centric* approach is generally preferred, so that the learned representations are aligned with the planning context [20]. Further desired requirements for representations are they enable predicting the future world state [21]–[25] (as opposed to merely reconstructing the present) and that they are *low-dimensional* [26], [27]. To mitigate the trade-off between dimensionality reduction and expressiveness, SRL can be supported by incorporating knowledge about the world as *representation priors* [18], [28], [29]. Imposing structural

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

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Fulltext available at doi:10.1109/ICRA48891.2023.10160559.

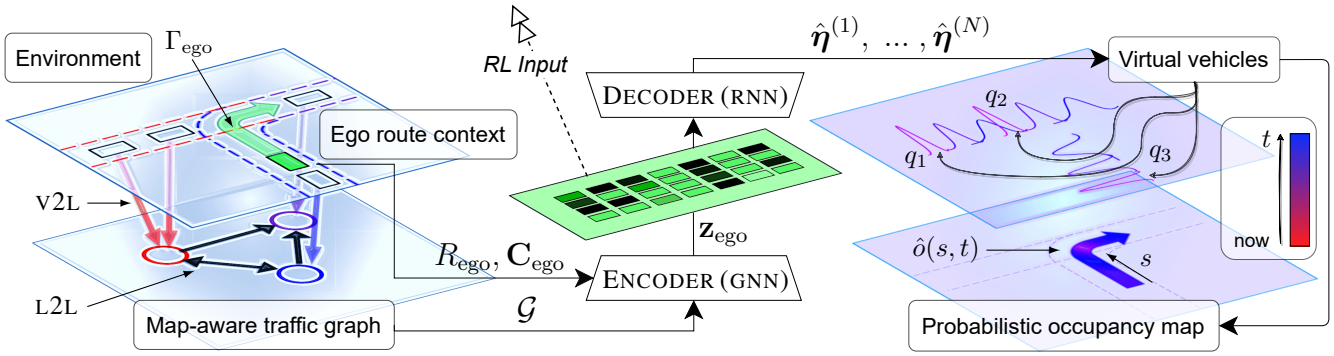


Fig. 2: Overview of our proposed architecture: The fixed-sized latent states $\mathbf{z}_{\text{ego}}$, which are encoded by an ego-conditioned, heterogeneous graph neural network, are extracted as inputs for an RL-based motion planner. Our novel decoder architecture, which infers the probabilistic occupancy map from reconstructed virtual vehicles, is used for pre-training the encoder and can be disabled during inference.

constraints on the representations, e.g. by enforcing correspondence to physically plausible world states, improves their generalization and downstream effectiveness [30], [31].

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 [32], 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 [33]. As necessitated by the multi-modal traffic graph formulation assumed in this work, GNNs are also extendable to heterogeneous graph inputs [34], [35].

C. Applications to motion planning

Autoencoder-based representation models [36] have been used in a multitude of existing works for learning latent states based on, e.g., rasterized bird's eye view images [37]–[41] or on-board sensor data [42]. However, they do not leverage the structural biases [43] induced by the road network topology. In line with our approach, [44] uses a GNN-based encoder to learn structurally-aware state representations, but in the context of RL-based robotic manipulators.

Road occupancy as a standalone prediction task is widely covered in existing works [45]–[49]. With the objective of encoding traffic scenes similar to ours (albeit not in the context of motion planning), encoder architectures for learning representations of occupancy maps have been proposed [50]–[52]. Using graphical or otherwise spatially-aware encoders similar to ours, recent works such as [53]–[57] predict occupancy grids [58] 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 [59].

III. Methodology

Next, we outline the details of our approach.

A. Definitions

1) Heterogeneous traffic graph: As originally proposed in [60], we model road networks as atomic, interconnected road segments (i.e., *lanelets*). We formalize the dynamic traffic environment at the current time step by the heterogeneous graph tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X_{\mathcal{V}}, X_{\mathcal{E}})$, where $\mathcal{V} = (\mathcal{V}_{\text{V}}, \mathcal{V}_{\text{L}})$ indexes the vehicle (V) and lanelet (L) nodes, $\mathcal{E} = (\mathcal{E}_{\text{V2L}}, \mathcal{E}_{\text{L2L}})$ defines the corresponding vehicle-to-lanelet (V2L) and lanelet-to-lanelet (L2L) edges, and $X_{\mathcal{V}} = (\mathbf{X}_{\text{V}}, \mathbf{X}_{\text{L}})$ and $X_{\mathcal{E}} = (\mathbf{X}_{\text{V2L}}, \mathbf{X}_{\text{L2L}})$ contain node and edge-level graph features, respectively. Here, the time-dependent V2L edges relate to the physical presence of a vehicle on a given lanelet, whereas the static L2L edges are implied by the road network topology. Our approach incorporates the default graph features provided by *crgeo* [16], e.g. velocity (V) and vehicle-lanelet heading difference (V2L).

2) Planning context: The state of the ego vehicle is chosen as the tuple $(\mathbf{p}_{\text{ego}}, v_{\text{ego}})$, consisting of its x-y center position and longitudinal speed. We further denote its length as λ_{ego}. Next, we let the reference path $\Gamma_{\text{ego}} : [0, \zeta_{\text{ego}}] \to \mathbb{R}^2$ of length ζ_{ego} be parameterized by arclength s, and impose the natural constraint that $\Gamma_{\text{ego}}(0) = \mathbf{p}_{\text{ego}}$. As illustrated in Fig. 3, we assume that Γ_{ego} follows the centerline of a connected, traffic-compliant sequence of lanelets. The corresponding sequence of lanelet node indices in $\mathcal{V}_{\text{L}}$ is denoted by R_{ego}. Further, we let s_j^{start} and s_j^{end} denote the start and endpoint coordinates of the j^{th} element in R_{ego}, as defined within the arclength-parameterized coordinate frame of the centerlines. Also, we let d_j denote lanelet length, and let d_j^{prior} be the aggregated length of the path segments preceding j. Finally, we let the spatial context matrix $\mathbf{C}_{\text{ego}}$ contain the row vectors $\mathbf{c}_j = [s_j^{\text{start}}, s_j^{\text{end}}, d_j, d_j^{\text{prior}}]$.

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 [61].

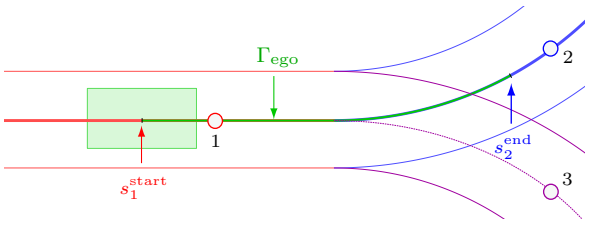


Fig. 3: Ego reference path Γ_{ego} composed of two successive lanelets given by $R_{\text{ego}} = [1, 2]$.

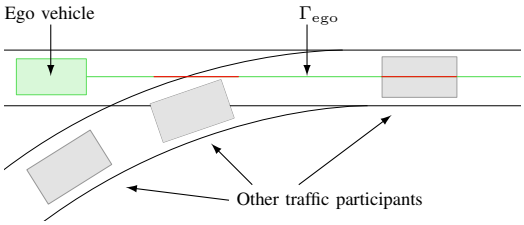


Fig. 4: The ground-truth path occupancy o is derived from projecting the vehicles in $\mathcal{V}_{\text{V}}$ onto Γ_{ego}. The occupied and non-occupied path segments are colored red and green, respectively.

We consider occupancy solely in the longitudinal direction, as this is more relevant than the lateral direction and simplifies our modelling assumptions. For a given spatio-temporal coordinate vector $[s, t] \in \mathbb{R}^2$, future path occupancy on Γ_{ego} can be formalized as $o : [0, \zeta_{\text{ego}}] \times \mathbb{R} \to \{0, 1\}$. As shown in Fig. 4, $o(s, t)$ is derived from a path projection of the vehicles that overlap with the road surface at time t.

C. Encoder architecture

Our encoding pipeline is formalized as

$$\mathbf{z}_{\text{ego}} = \textsc{Encoder}(\mathcal{G}, R_{\text{ego}}, \mathbf{C}_{\text{ego}}), \tag{1}$$

with the low-dimensional representations $\mathbf{z}_{\text{ego}} \in \mathbb{R}^Z$ being the final output of the encoder. The GNN-based encoder is designed to facilitate the probabilistic propagation of traffic participants across the given lanelet network, which is used as a neural infrastructure for social message passing. Next, we outline the encoding steps. In general, we use $\Theta_\square$ to denote trainable, nonlinear functions, Σ to denote a permutation-invariant aggregation operation, and assume the activation function ρ to be applied after each step.

1) Vehicle-to-lanelet: Unlike the vehicle-to-vehicle paradigm proposed in e.g. [5], we capture V2V interaction effects in a map-agnostic fashion by first embedding the vehicles onto the lanelet graph. Letting vehicle and lanelet nodes be indexed by i and j, respectively, we compute the initial hidden lanelet states $\mathbf{h}_j^{(0)} \in \mathbb{R}^H$ as

$$\mathbf{h}_j^{(0)} = \Theta_{\text{L}}(\mathbf{x}_j) + \sum_{(i,j)\in\mathcal{E}_{\text{V2L}}} \Theta_{\text{V2L}}([\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_{i\to j}]).$$

where the node and edge features $(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{x}_{i\to j}$ denote the corresponding row vectors in $X_{\mathcal{V}}$ and $X_{\mathcal{E}}$, respectively.

2) Lanelet-to-lanelet: Next, a total of L successive L2L message passing layers are used to facilitate the propagation of information flow across the lanelet network. With the superscript l denoting the layer index, we recursively update the hidden lanelet states according to the update equation

$$\mathbf{h}_j^{(l+1)} = \mathbf{h}_j^{(l)} + \sum_{(j',j)\in\mathcal{E}_{\text{L2L}}} \Theta_{\text{L2L}}([\mathbf{h}_{j'}^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{x}_{j'\to j}]),$$

3) Ego-attentional readout operation: Then, an attentional readout layer [62] is used to compute graph-level hidden states $\mathbf{h}_{\text{ego}} \in \mathbb{R}^H$. To obtain ego-centered representations, the aggregation of $\mathbf{h}_j^{(L)}$ is weighted by attention scores $\boldsymbol{\alpha}_{\text{ego}}$ based on the ego vehicle's spatial context $\mathbf{C}_{\text{ego}}$, i.e.,

$$\boldsymbol{\alpha}_{\text{ego}} = \text{softmax}\left(\Theta_C(\mathbf{C}_{\text{ego}})\right),$$
$$\mathbf{h}_{\text{ego}} = \sum_{j\in R_{\text{ego}}} \alpha_{\text{ego},j} \mathbf{h}_j^{(L)}.$$

4) Downscaling layer: Finally, a downscaling MLP layer is applied to obtain the final latent states $\mathbf{z}_{\text{ego}} = \Theta_z(\mathbf{h}_{\text{ego}})$, intended to be used as state observations for the RL agent.

D. Decoder architecture

Our proposed decoder maps $\mathbf{z}_{\text{ego}}$ to a continuous parameterization of the probabilistic occupancy map according to

$$\begin{aligned} \hat{o}(s,t) &= \textsc{Decoder}(\mathbf{z}_{\text{ego}}, s, t), \\ o(s,t) &\sim \text{Bernoulli}(\hat{o}(s,t)). \end{aligned} \tag{2}$$

However, as an abstraction of something tangible (i.e., the presence of vehicles), predicting $o(s, t)$ in an unconstrained fashion using, e.g., a regular MLP network, might lead to overparameterized predictions that are inconsistent with the data [63]. The large hypothesis space is especially problematic given the low-dimensional nature of $\mathbf{z}_{\text{ego}}$ and the resulting information bottleneck. Instead, our novel decoding architecture prevents nonsensical predictions of the occupancy map by inferring it from a decoded set of time-evolving probability distributions referred to as *virtual vehicles*. This enforces temporal and spatial consistency on the output space and streamlines the learning task. By exploiting the physical priors of our application domain, our method further guarantees that the decoded occupancy maps conform to plausible limits for e.g. vehicle length and velocity.

1) Virtual vehicles: We parameterize the probabilistic occupancy map $\hat{o}(s, t)$ via recurrently decoded virtual vehicles. As outlined in Section III-C, our fixed-sized intermediate representations $\mathbf{z}_{\text{ego}}$ are computed by aggregating the elements in $\mathcal{G}$. Due to the resulting node-level data association loss, it is not viable to reconstruct vehicle instances in a way comparable to related trajectory prediction works [7]–[12]. Without assuming an association between decoded and actual vehicles, our virtual vehicle formulation instead enables a differentiable and permutation-invariant parameterization of the joint probabilistic occupancy map $\hat{o}(s, t)$. This translates the learning problem to the graph-level (i.e., global) domain, yielding a feasible training target.

2) Formal definition: Formally, we let the state of a decoded virtual vehicle q of length $\lambda_q \in \mathbb{R}$ be defined by the tuple $(\mathcal{I}_q, p_q)$, where $\mathcal{I}_q : \mathbb{R} \to \{0, 1\}$ is an existence indicator at time t, and $p_q : \mathbb{R} \to \mathbb{R}$ models its time-dependent longitudinal center position on Γ_{ego}. Further, we let $o_q : [0, \zeta_{\text{ego}}] \times \mathbb{R} \to \{0, 1\}$ return its path occupancy

$$o_q(s,t) = \begin{cases} 1 & \text{if} \quad \mathcal{I}_q(t) = 1 \wedge |s - p_q(t)| < \frac{\lambda_q}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This expression differs from the vehicle occupancy o considered in Fig. 4, as virtual vehicles serve as non-deterministic, atomic proxies for modelling future occupancy flow.

3) Stochastic formulation: We let $f_p : \mathbb{R} \times \mathbb{R} \to [0, 1]$ denote the time-varying probability density function (PDF) for p_q. As motivated in [64], the *Fokker-Planck* [65] equation can be used for modelling microscopic traffic flow as a stochastic process. In this framework, p_q is described by a partial differential equation influenced by stochastic forces, addressing the accumulation of uncertainty over time with regards to the vehicle position. As the model training requires a differentiable inference procedure, we use a simplified behavior model with a tractable solution. Specifically, we assume linear drift and diffusion terms given by $\hat{\eta}_{p,v}$ and $\hat{\eta}_{p,d}$, as well as a positional offset $\hat{\eta}_{p,0}$ corresponding to the initial vehicle position. Using ";" to separate the distributions' input space from their given parameterizations, this results in the time-evolving Gaussian solution [65]

$$\begin{aligned} f_p(s; t, \hat{\eta}_p) &= \frac{1}{\sqrt{4\pi\hat{\eta}_{p,d}t}} \exp\left(-\frac{(s - \hat{\eta}_{p,0} - \hat{\eta}_{p,v}t)^2}{4\hat{\eta}_{p,d}t}\right), \\ F_p(s; t, \hat{\eta}_p) &= -\frac{1}{2} \operatorname{erf}\left(\frac{\hat{\eta}_{p,v}t + \hat{\eta}_{p,0} - s}{2\sqrt{\hat{\eta}_{p,d}t}}\right), \end{aligned} \tag{4}$$

with $F_p(s; t, \hat{\eta}_p) = \int_{-\infty}^{s} f_p(s'; t, \hat{\eta}_p^{(q)})\, \mathrm{d}s'$ denoting the cumulative distribution function. Further, we let $f_{\mathcal{I}} : \mathbb{R} \to [0, 1]$ denote the predicted existence probability for q at a given t. Delayed appearances or disappearances of vehicles on Γ_{ego} due to, e.g., lane changes cannot be captured by a constant existence probability $f_{\mathcal{I}}(t) = \hat{\eta}_{\mathcal{I},0}$. By also introducing the temporal offset parameter $\hat{\eta}_{\mathcal{I},\tau}$, we instead use the time-dependent parameterization given by

$$\begin{aligned} m_l(t, \hat{\eta}_{\mathcal{I},\tau}) &= \sigma(\tau_R(t - \hat{\eta}_{\mathcal{I},\tau}(1+\tau_C) + \tau_C)), \\ m_r(t, \hat{\eta}_{\mathcal{I},\tau}) &= \sigma(\tau_R(1 - t + \hat{\eta}_{\mathcal{I},\tau}(1+\tau_C) + \tau_C)), \\ m(t, \hat{\eta}_{\mathcal{I},\tau}) &= m_l(t, \hat{\eta}_{\mathcal{I},\tau}) \cdot m_r(t, \hat{\eta}_{\mathcal{I},\tau}), \\ f_{\mathcal{I}}(t; \hat{\eta}_{\mathcal{I},0}, \hat{\eta}_{\mathcal{I},\tau}) &= \hat{\eta}_{\mathcal{I},0} \cdot m(t, \hat{\eta}_{\mathcal{I},\tau}), \end{aligned} \tag{5}$$

with σ denoting the sigmoid function, and τ_C and τ_R being fixed model hyperparameters. As visualized in Fig. 5, the temporal masking effect produced by the left and right shifting mechanism m provides the modelling flexibility for $f_{\mathcal{I}}$ to predict vehicles leaving or entering Γ_{ego} at future time instances. Given the vector parameterization $\hat{\boldsymbol{\eta}}^{(q)} = [\hat{\eta}_{\lambda}^{(q)}, \hat{\eta}_{\mathcal{I},0}^{(q)}, \hat{\eta}_{\mathcal{I},\tau}^{(q)}, \hat{\eta}_{p,0}^{(q)}, \hat{\eta}_{p,d}^{(q)}, \hat{\eta}_{p,v}^{(q)}]$, where $\hat{\eta}_{\lambda}^{(q)}$ is the decoded length, and $\hat{\eta}_{\mathcal{I}}^{(q)}$ and $\hat{\eta}_{p}^{(q)}$ contain the respective coefficients for $f_{\mathcal{I}}$ and f_p, we model q stochastically as

$$\begin{aligned} P(\mathcal{I}_q(t) = 1) &= f_{\mathcal{I}}(t; \hat{\eta}_{\mathcal{I}}^{(q)}), \\ P(p_q(t) = s) &= f_p(s; t, \hat{\eta}_p^{(q)}). \end{aligned} \tag{6}$$

Next, we derive the unimodal occupancy map $\hat{o}_q(s, t)$, referred to as the probabilistic occupancy *footprint* of a single virtual vehicle q. We assume that $\mathcal{I}_q(t)$ and $p_q(t)$ are independent random variables and use the shorthand notation $s_{\pm} = s \pm \hat{\eta}_{\lambda}^{(q)}/2$ to denote the occupancy bounds for a given s. As is illustrated in Fig. 6, applying (3) and (6) results in

$$\begin{aligned} \hat{o}_q(s,t) &= P(o_q(s,t) = 1) \\ &= P(\mathcal{I}_q(t) = 1) P(|s - p_q(t)| < \hat{\eta}_{\lambda}^{(q)}/2) \\ &= f_{\mathcal{I}}(t; \hat{\eta}_{\mathcal{I}}^{(q)}) \int_{s_-}^{s_+} f_p(s'; t, \hat{\eta}_p^{(q)})\, \mathrm{d}s', \\ &= f_{\mathcal{I}}(t; \hat{\eta}_{\mathcal{I}}^{(q)})(F_p(s_+; t, \hat{\eta}_p^{(q)}) - F_p(s_-; t, \hat{\eta}_p^{(q)})). \end{aligned} \tag{7}$$

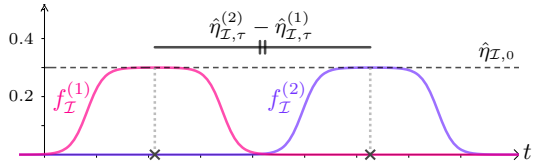
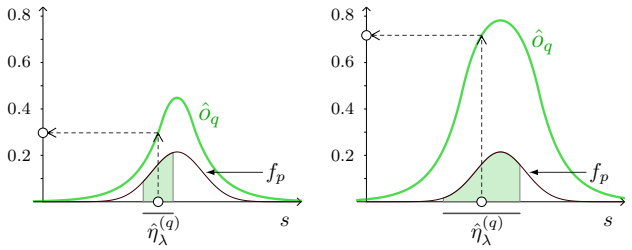


Fig. 5: Existence probability $f_{\mathcal{I}}(t)$ for two virtual vehicles with identical baseline existence probability $\hat{\eta}_{\mathcal{I},0} = 0.3$ and different temporal offset parameters $(\hat{\eta}_{\mathcal{I},\tau}^{(1)}, \hat{\eta}_{\mathcal{I},\tau}^{(2)})$.



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

Fig. 6: The predicted occupancy footprint $\hat{o}_q(s, t)$ of a virtual vehicle q at a future time t is derived via a symmetrically bounded integral of length $\hat{\eta}_{\lambda}^{(q)}$ centered at s. Here, the two vehicles have identical positional PDFs f_p, but different lengths $\hat{\eta}_{\lambda}^{(q)}$. As expected, a larger $\hat{\eta}_{\lambda}^{(q)}$ induces a larger predicted occupancy footprint.

E. Joint occupancy probability

We next derive an expression for the predicted joint occupancy $\hat{o}(s, t)$. To facilitate tractable inference, we assume the virtual vehicles to be independent. For a set of virtual vehicles $\mathcal{Q} = \{q_1, ..., q_N\}$, the probability of *at least one* vehicle occupying Γ_{ego} at the location $[s, t]$ is given by

$$\hat{o}(s,t) = 1 - \prod_{q \in \mathcal{Q}} (1 - \hat{o}_q(s,t)), \tag{8}$$

as exemplified in Fig. 7. Given a sufficiently large N, neither the independence assumption nor the simplified linear motion model assumed in (4) pose significant restrictions on the modeling capacity of our decoder, as (8) allows complex behavior patterns and the corresponding occupancy maps to be modeled via superimposed virtual vehicles.

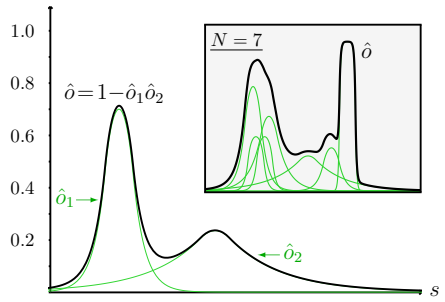


Fig. 7: Spatial cross-section of the probabilistic occupancy map $\hat{o}(s,t)$ induced by two virtual vehicles at a given time t. In the top right subplot, more virtual vehicles are added for comparison.

1) Recurrent decoding: We employ a recurrent neural network Θ_D and a downscaling layer Θ_q to decode the virtual vehicle parameterizations $\{\hat{\boldsymbol{\eta}}^{(1)}, ..., \hat{\boldsymbol{\eta}}^{(N)}\}$ from $\mathbf{z}_{\text{ego}}$. We transform the raw outputs of Θ_q via rescaled sigmoid functions so as to conform to $\boldsymbol{\eta}_{\min} \leq \hat{\boldsymbol{\eta}}^{(q)} \leq \boldsymbol{\eta}_{\max}$. The parameter bounds $\boldsymbol{\eta}_{\min}$ and $\boldsymbol{\eta}_{\max}$ correspond to reasonable priors for plausible vehicle behavior in traffic and are assumed to be fixed hyperparameters.

F. Spatio-temporal occupancy loss

A naive approach towards defining the loss function is to consider the binary cross-entropy over a uniform grid discretization of the spatio-temporal occupancy domain [66]. Instead, we compute the loss segment-wise in a boundary-aware fashion by leveraging the continuous output domain of our occupancy decoder. As shown in Fig. 8, we let $\mathcal{O}_p^{(t)}$ and $\mathcal{O}_n^{(t)}$ contain the occupied and non-occupied connected segments of Γ_{ego} at time t. The respective path segments are indexed as $\Omega_\square \subseteq [0, \zeta_{\text{ego}}]$ according to their topological order. Minimizing the binary cross-entropy, we then define our decoding loss as

$$\ell_p(t) = -\sum_{\Omega \in \mathcal{O}_p^{(t)}} \frac{1}{|\Omega|} \int_{s\in\Omega} \log\left(\hat{o}(s,t)\right)\, \mathrm{d}s,$$

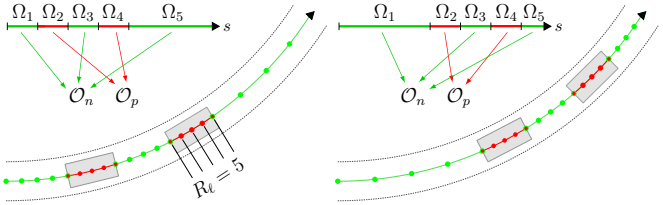
$$\ell_n(t) = -\sum_{\Omega \in \mathcal{O}_n^{(t)}} \frac{1}{|\Omega|} \int_{s\in\Omega} \log\left(1-\hat{o}(s,t)\right)\, \mathrm{d}s,$$

$$\ell = \int_{t=0}^{T} \delta_\ell^t \big(\ell_p(t) + \ell_n(t)\big)\, \mathrm{d}t, \tag{9}$$

where $T \in \mathbb{R}$ is the considered time horizon and $\delta_\ell \in [0,1]$ is a discount factor for reflecting the diminishing significance of future occupancy. With $|\Omega|$ denoting the arclength of the respective path segment, the normalization factor $1/|\Omega|$ is introduced to avoid the dependence on segment length and to counteract the class imbalance caused by road surfaces being predominantly unoccupied. As the spatial integrals are otherwise intractable, we approximate them numerically with resolution R_ℓ as illustrated in Fig. 8a. Similarly, we discretize the temporal integral over T_D time steps.

IV. Numerical Experiments

Next, we describe the collection of our simulated traffic dataset and introduce the numerical RL experiments that evaluate the effectiveness of our learned state representations.



(a) $t = 0.0\,\text{s}$. (b) $t = 0.1\,\text{s}$.

Fig. 8: Loss evaluation for two subsequent time instances. The segment-wise integrals over the path regions in $\mathcal{O}_p^{(t)}$ and $\mathcal{O}_n^{(t)}$ are approximated numerically based on R_ℓ evenly spaced samples.

A. Dataset

Using the OpenStreetMap [67] 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 [68]. In total, the dataset contains 1000 simulated scenarios, jointly amounting to 28 hours of traffic.

B. Reinforcement learning agent

For our experiments, we extract $\mathbf{z}_{\text{ego}}$ as inputs for a PPO-based [69] RL agent together with the current ego velocity v_{ego}. At each time step, the encoder is conditioned on the navigation context generated by a high-level route planner. The resulting reference routes span across multiple lanelets and include heterogeneous map structures such as intersections. We intentionally limit the agent to longitudinal acceleration control along the reference path to focus on the effect of the learned representations. Given the weighting coefficients $\mathbf{w} \in \mathbb{R}^4$, the reward r is defined as $r = \mathbf{w} \cdot [r_{path}, r_{collision}, r_{speed}, r_{\hat{o}}]$, where r_{path} is a dense path progression reward, $r_{collision}$ is a sparse penalty imposed on collisions, r_{speed} is a linear over-speeding penalty, and $r_{\hat{o}}$ penalizes expected occupancy conflict from

$$r_{\hat{o}} = \int_{t=0}^{T} \delta_r^t \int_{\tilde{s}(t)-\lambda_{\text{ego}}/2}^{\tilde{s}(t)+\lambda_{\text{ego}}/2} \hat{o}(s,t)\, \mathrm{d}s\, \mathrm{d}t, \tag{10}$$

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

C. Baseline approaches

We compare our PPO agent's performance against multiple baselines trained with identical reward configurations:

1) **V2V**: A vehicle-to-vehicle (i.e., not map-aware) GNN policy network as proposed in [5].
2) **V2L**: A direct adoption of our Encoder as policy network (i.e., without pre-training $\mathbf{z}_{\text{ego}}$).
3) **Naive**: The pre-trained representations resulting from using a feed-forward network $\text{MLP}(256, 128)$ for decoding $\hat{o}(s,t)$ from $[\mathbf{z}_{\text{ego}}, s, t]$ without architectural constraints. This naively assumes occupancy to be an independent property for each spatio-temporal coordinate vector $[s,t]$.

D. Implementation and training

The end-to-end training of the representation model was conducted on an NVIDIA A100 Tensor Core GPU for 48 hours using the Adam optimizer [70] with the hyperparameters listed in Table I. Subsets of the collected dataset were used for training (90%) and testing (10%). To aid generalization, random planning contexts for the encoder were resampled for each mini-batch during training. The PPO agents were implemented using the Stable-Baselines 3 framework [71] and trained for 10^6 steps on simulated replays of the scenarios, with start and goal positions for the ego vehicle being randomly sampled for each episode.

TABLE I: Selected hyperparameters for our experiments. We refer to our online available implementation for further details.

Encoding layers ($\Theta_{\text{L}}, \Theta_{\text{V2L}}, \Theta_{\text{L2L}}, \Theta_C, \Theta_z$)	Linear
Encoding dimensions (H, Z)	256, 32
Number of L2L layers (L)	4
Aggregation function (Σ)	max
Activation function (ρ)	tanh
Reference path length (ζ_{ego})	45 m
Decoding layers (Θ_D, Θ_q)	LSTM(256), Linear
Number of virtual vehicles (N)	12
Temporal masking constants (τ_R, τ_C)	6.0, 0.7
Decoding horizon (T, T_D)	2.4 s, 60
Spatial integration method	Trapezoidal($R_\ell = 40$)
Temporal occupancy discount factor (δ_ℓ)	0.99

V. Results

As is evident from the results reported in Table II, 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 pipeline. Specifically, it is likely that the simpler hypothesis space streamlines the training process. Further, a qualitative assessment of the decoded probability maps shown in Fig. 9 suggests that our model is able to accurately predict complex environments to a degree where the intermediate encodings $\mathbf{z}_{\text{ego}}$ will enable intelligent planning decisions for the agent.

TABLE II: Empirical evaluation results on the test dataset.

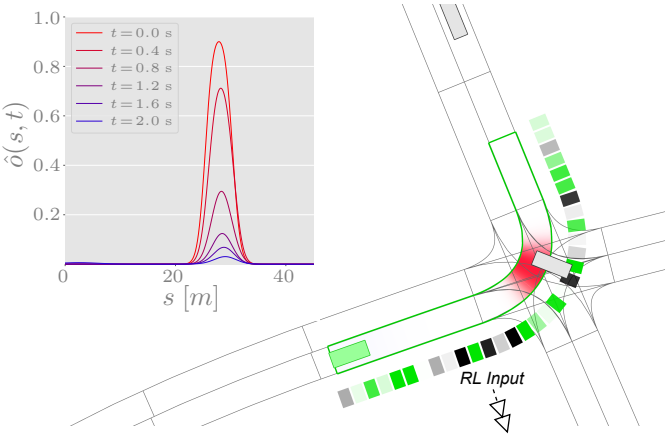
(a) Occupancy decoding loss.

model	ℓ
Ours	**1.045**
Naive	1.210

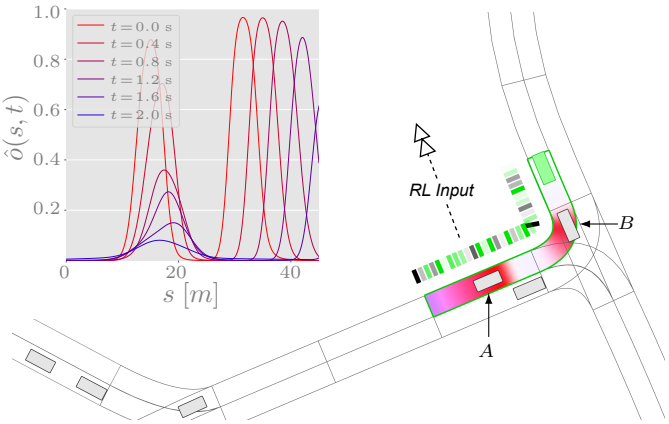
(b) Downstream RL performance.

agent	goal reach (%)
Ours	**72.9**
V2V	39.9
V2L	49.0
Naive	54.0

The effectiveness of our representations is confirmed by the results of the conducted RL experiments. The improved success rate of the representation-enhanced agent indicates that the pre-trained representations simplify its motion planning task. Effectively, using them as state observations frees the agent from the responsibility of modeling its own surroundings, allowing it to concentrate its learning capacity 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 unsurprising that the simplification of the RL task improves the final performance.



(a) Vehicle from another lane intersecting Γ_{ego}. The occupancy probability map has a temporary peak located at the intersection point.



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

Fig. 9: Decoded occupancy maps $\hat{o}(s,t)$ visualized together with color encodings of the corresponding latent representations $\mathbf{z}_{\text{ego}}$, which are extracted as state observations for the RL agent.

VI. Conclusion

We present a novel encoder-decoder architecture for learning latent state representations that enhance the performance of RL-based motion planning in heterogeneous driving environments. Our approach recurrently decodes and aggregates virtual vehicles to predict the spatio-temporal occupancy map. This ensures that the representation space is constrained to physically plausible predictions, which further enhances the model's interpretability and reduces its black-box nature. By using a heterogeneous GNN encoder to compress the traffic surroundings, our approach offers a significant benefit compared to previous works; as opposed to feature vectors narrowly tailored to specific traffic settings, our approach naturally extends to arbitrary road networks.

Acknowledgements

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

References

[1] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "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, D. Fishman, and N. Muhammad, "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, and J. Boedecker, "Dynamic interaction-aware scene understanding for reinforcement learning in autonomous driving," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4329–4335.

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

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

[7] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020, pp. 541–556.

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

[9] B. Kim, S. Park, S. S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. S. Kim, and J. W. Choi, "LaPred: Lane-aware prediction of multi-modal future trajectories of dynamic agents," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 631–14 640.

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

[11] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "GOHOME: Graph-oriented heatmap output for future motion estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9107–9114.

[12] X. Mo, Z. Huang, Y. Xing, and C. Lv, "Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9554–9567, 2022.

[13] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *AAAI Conference on Artificial Intelligence*, 2018.

[14] L. Buşoniu, T. d. Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in Control*, vol. 46, pp. 8–28, 2018.

[15] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester, "Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis," *Machine Learning*, vol. 110, no. 9, pp. 2419–2468, 2021.

[16] E. Meyer, M. Brenner, B. Zhang, M. Schickert, B. Musani, and M. Althoff, "Geometric deep learning for autonomous driving: Unlocking the power of graph neural networks with CommonRoad-Geometric," 2023. arXiv: 2302.01259.

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

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

[19] M. Schwarzer, N. Rajkumar, M. Noukhovitch, A. Anand, L. Charlin, R. D. Hjelm, P. Bachman, and A. C. Courville, "Pretraining Representations for Data-Efficient Reinforcement Learning," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 12 686–12 699.

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

[21] 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.

[22] Z. D. Guo, B. A. Pires, B. Piot, J.-B. Grill, F. Altché, R. Munos, and M. G. Azar, "Bootstrap latent-predictive representations for multitask reinforcement learning," in *Proc. of the International Conference on Machine Learning (ICML)*, 2020, pp. 3875–3886.

[23] K.-H. Lee, I. Fischer, A. Z. Liu, Y. Guo, H. Lee, J. Canny, and S. Guadarrama, "Predictive information accelerates learning in RL," in *Proc. of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2020, pp. 11 890–11 901.

[24] S. Recanatesi, M. Farrell, G. Lajoie, S. Deneve, M. Rigotti, and E. Shea-Brown, "Predictive learning as a network mechanism for extracting low-dimensional latent space representations," *Nature Communications*, vol. 12, no. 1, pp. 1–13, 2021.

[25] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 2455–2467.

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

[27] R. Bassily, S. Moran, I. Nachum, J. Shafer, and A. Yehudayoff, "Learners that use little information," in *Proc. of the International Conference on Algorithmic Learning Theory (ALT)*, 2018, pp. 25–55.

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

[29] 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.

[30] J. Scholz, M. Levihn, C. L. I. Jr., and D. Wingate, "A physics-based model prior for object-oriented MDPs," in *Proc. of the International Conference on Machine Learning (ICML)*, 2014, pp. 1089–1097.

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

[32] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. of the International Conference on Machine Learning (ICML)*, 2017, pp. 1263–1272.

[33] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, no. 1, pp. 57–81, 2020.

[34] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proc. of the European Semantic Web Conference*, 2018, pp. 593–607.

[35] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *Proc. of the The World Wide Web Conference*, 2019, pp. 2022–2032.

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

[37] B. Toghi, R. Valiente, R. Pedarsani, and Y. P. Fallah, "Towards learning generalizable driving policies from restricted latent representations," 2021. arXiv: 2111.03688.

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

[39] J. Zhao, J. Fang, Z. Ye, and L. Zhang, "Large scale autonomous driving scenarios clustering with self-supervised feature extraction," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 473–480.

[40] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8248–8254.

[41] P. Cai, H. Wang, Y. Sun, and M. Liu, "DiGNet: Learning scalable self-driving policies for generic traffic scenarios with graph neural networks," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8979–8984.

[42] J. Dong, S. Chen, Y. Li, P. Y. J. Ha, R. Du, A. Steinfeld, and S. Labi, "Spatio-weighted information fusion and DRL-based control for connected autonomous vehicles," in *Proc. of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[43] P. W. Battaglia, J. B. Hamrick, V. Bapst, *et al.*, "Relational inductive biases, deep learning, and graph networks," 2018. arXiv: 1806.01261.

[44] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, "Graph networks as learnable physics engines for inference and control," in *Proc. of the International Conference on Machine Learning (ICML)*, 2018, pp. 4470–4479.

[45] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2056–2063.

[46] N. Mohajerin and M. Rohani, "Multi-step prediction of occupancy grid maps with recurrent neural networks," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 600–10 608.

[47] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-term occupancy grid prediction using recurrent neural networks," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9299–9305, 2019.

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

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

[50] O. Rákos, T. Bécsi, S. Aradi, and P. Gáspár, "Learning latent representation of freeway traffic situations from occupancy grid pictures using variational autoencoder," *Energies*, vol. 14, no. 17, 2021, paper 5232.

[51] L. A. Marina, B. Trasnea, T. Cocias, A. Vasilcoi, F. Moldoveanu, and S. M. Grigorescu, "Deep Grid Net (DGN): A deep learning system for real-time driving context understanding," in *Proc. of the IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 399–402.

[52] M. Itkina, K. Driggs-Campbell, and M. J. Kochenderfer, "Dynamic environment prediction in urban scenes using recurrent representation learning," in *Proc. of the IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 2052–2059.

[53] E. Amirloo, M. Rohani, E. Banijamali, J. Luo, and P. Poupart, "Self-supervised simultaneous multi-step prediction of road dynamics and cost map," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8494–8503.

[54] P. Hu, A. Huang, J. M. Dolan, D. Held, and D. Ramanan, "Safe local motion planning with self-supervised freespace forecasting," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021, pp. 12 732–12 741.

[55] S. Casas, A. Sadat, and R. Urtasun, "MP3: A unified model to map, perceive, predict and plan," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 403–14 412.

[56] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2020, pp. 414–430.

[57] R. Mahjourian, J. Kim, Y. Chai, M. Tan, B. Sapp, and D. Anguelov, "Occupancy flow fields for motion forecasting in autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5639–5646, 2022.

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

[59] M. Ilievski, S. Sedwards, A. Gaurav, A. Balakrishnan, A. Sarkar, J. Lee, F. Bouchard, R. De Iaco, and K. Czarnecki, "Design space of behaviour planning for autonomous driving," 2019. arXiv: 1908.07931.

[60] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 420–425.

[61] 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.

[62] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations (ICLR)*, 2018.

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

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

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

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

[67] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, https://www.openstreetmap.org, 2017.

[68] P. A. Lopez, E. Wiessner, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, and P. Wagner, "Microscopic Traffic Simulation using SUMO," in *Proc. of the International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.

[69] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms.," 2017. arXiv: 1707.06347.

[70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[71] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.