VectorRL: Interpretable Graph-based Reinforcement Learning for Automated Driving

Christopher Diehl, Tamino Waldeyer, Frank Hoffmann, Torsten Bertram

Institute of Control Theory and Systems Engineering, TU Dortmund
44227 Dortmund, Germany
E-Mail: forename.surname@tu-dortmund.de

1 Introduction

Safe and efficient motion planning and control are crucial components for automated driving. The modeling and prediction of multi-agent interactions in traffic provides a challenge for current decision-making in driving tasks. Often driving policies are designed manually for specific scenarios, which is time-consuming both in development and maintenance. On the other hand, reinforcement learning (RL) learns and improves driving policies in a trial-and-error fashion, with little design and engineering effort. Current RL approaches for automated driving utilize a variety of state-space representations. Hoel et. al [1] propose a feature vector composed of position, speed, and lane information. This representation requires a fixed size input. Huegle et. al [3] employ deep sets [4] to process perceptions of variable dimensionality. However, they do not encode detailed context information. Fixed-size multi-layer grid maps (MLG) [2] easily represent semantic context information in the vehicle's environment. However, they impose a trade-off between computational efficiency, memory consumption, and performance. Recent work [5] in the area of trajectory prediction proposes to encode object and context information as vectors. This comes with the advantage of low discretization errors and computational workload while achieving better performance. To

DOI: 10.58895/ksp/1000138532-4 erschienen in:
Proceedings - 31. Workshop Computational Intelligence : Berlin, 25. - 26. November 2021
DOI: 10.58895/ksp/1000138532 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000138532/

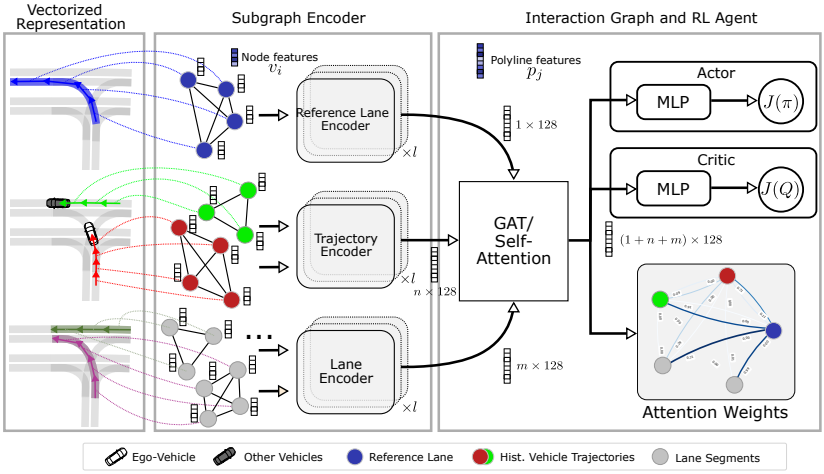

Figure 1: VectorRL system architecture.

overcome the previously described representation complexity, this work proposes a novel graph-based RL approach that relies on a vectorized environment representation. Different attention mechanisms provide insight into the regions and objects relevant to the agent's decision-making. Visualization of the attention states contributes to the interpretability of the learned policy. The graph-based RL approach is evaluated in an urban scenario in a realistic simulation environment. It is compared to several state-of-the-art baselines, which rely on grid-based environment representations. The analysis shows that the graph-based approach outperforms the baselines on all metrics.

2 Vector-based Reinforcement Learning

This section introduces the RL problem formulation and the proposed architecture.

Problem Formulation. Let us model the RL task as a *Markov Decision Process* (MDP), defined by the tuple $\mathscr{M} = (\mathscr{S}, \mathscr{A}, p, r, \gamma)$. $r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ denotes the *reward*. The *policy* $\pi : \mathscr{S} \rightarrow \mathscr{A}$ maps from *states* $\mathbf{s} \in \mathscr{S}$ to a probability distribution over actions $\mathbf{a} \in \mathscr{A}$. The goal is to estimate an *optimal*

policy function $\pi^* = \arg\max_{a \in \mathscr{A}} \sum_{t=1}^{H} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ that maximizes the finite-horizon cumulative reward over the horizon H. $\gamma \in [0 \dots 1]$ denotes a discount factor.

Approach. The RL problem relies on a graph-based state representation. Figure 1 visualizes the architecture of the proposed approach. The planned route, lane information, and all object trajectories are represented as polylines of length $d \in \mathbb{R}$. Each polyline $P_j \in \mathscr{P}$ with index $j \in \mathbb{N}^+$ is mapped onto $n-1$ equidistant vectors $\mathbf{v}_i \in P_j$ with $\mathbf{v}_i = [\mathbf{d}_i^s, \mathbf{a}_i, j]$. $\mathbf{d}_i^s, \mathbf{d}_i^e \in \mathbb{R}^2$ are the 2-D start and end positions w.r.t. the self-driving vehicles coordinate system with vector index $i \in \mathbb{N}^+$. Further, $\mathbf{a}_i$ is a set of attributes. The route and lane polylines contain width, velocity limit, and intersection information. The attributes of the vehicle polylines characterize its width and length, and orientation. Furthermore, polylines contain a node indicator. Following the work of [5], fully connected sub-graphs encode the corresponding information. Global graph models capture the higher-order interactions between sub-graphs. Whereas the original approach relies on a self-attention (SA) [6] mechanism, our approach in addition investigates graph-attention (GAT) mechanisms [7]. The Soft Actor-Critic (SAC) [8] agent employs the resulting embedding as state-space representation. The action $\mathbf{a}_t$ at time t consists of a normalized continuous acceleration $a \in [0 \dots 1]$, braking signal $b \in [0 \dots 1]$, and steering angle $\delta \in [-1 \dots 1]$. The reward function is similar to the work of [2]:

$$r(s_t, a_t) = \lambda_1 r_{\mathrm{v}} + \lambda_2 r_{\mathrm{lat}} + \lambda_3 r_{\mathrm{col}} + \lambda_4 r_{\mathrm{lane}} - 0.1 \tag{1}$$

r_{col}, r_{lat} and r_{v} penalize collisions as well as deviations from the reference lane and the reference velocity, respectively. r_{lane} and the constant term impose high negative reward, in case the vehicle leaves its lane or stops. The main advantage of the proposed scheme is the ability to visualize the individual attention weights as illustrated in Figure 2. A high color saturation indicates a strong attention of the agent to these polylines. The attention visualization provides insights into the decision-making progress of the SDV, which is crucial for the acceptance of learning-based driving policies. While the agent pays close attention to nearby vehicles during merging (Left image of Figure 2), the attention remote vehicles (Right image of Figure 2) remains low, as these do not compromise the immediate safety of the SDV.

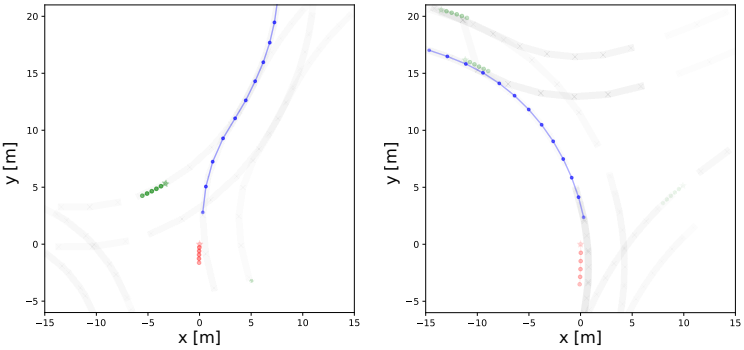

Figure 2: Visualization of the attention weights. The current positions of the SDV (red) and other agents (green) are marked by a star. The history is denoted with circles. Grey indicates the graph of the lane centers. Blue denotes they waypoints of the global route. A high saturation visualizes a high attention weight.

3 Evaluation

This section evaluates the approach in a challenging urban roundabout scenario in the CARLA [9] simulator (Version 0.9.11). The RL algorithm is implemented within Open AI Gym [11]. We compare against multiple baselines considering different metrics. A roundabout scenario in Town 1 based on the work of [2] is constructed for the purpose of policy evaluation.

Baselines. *BEV-OFF*: The approach of [2] first trains an autoencoder (AE) offline. The AE maps a Bird's-eye view (BEV) image to a latent space representation. Then the SAC algorithm is trained based on the latent representation *BEV-ON*: The work of [10] trains the autoencoder together with the RL agent by minimizing a multi-task objective. *VectorRL-SA, VectorRL-GAT*: The proposed approach either employs self-attention or graph-attention for the global graph interaction.

Metrics. *Success Rate* (SR): The proportion of collision-free episodes in which the SDV reaches its final destination. *Progress* (P): The average distance the vehicle travels. *Velocity Tracking Precision* (VTP): Average normalized tracking error of the reference velocity. One indicates the optimal tracking

Table 1: Performance in the roundabout scenario.

Approach	SR [%]	P [m]	VTP	LTE [m]
BEV-OFF	64	83.80 ± 0.81	0.64 ± 0.29	0.40 ± 0.29
BEV-ON	68	91.30 ± 0.79	0.59 ± 0.25	$\mathbf{0.29} \pm 0.28$
VectorRL-GAT	96	108.10 ± 0.80	0.71 ± 0.29	0.41 ± 0.33
VectorRL-SA	**98**	$\mathbf{110.00} \pm 0.80$	$\mathbf{0.73} \pm 0.3$	$\mathbf{0.29} \pm 0.39$

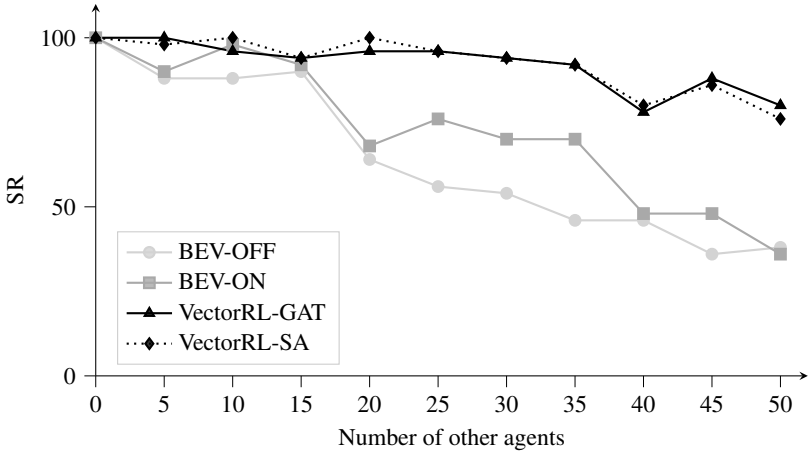

Figure 3: Generalization capabilities using an increasing number of obstacle objects and changing exits.

performance. *Lateral Tracking Error* (LTE) The mean lateral deviation to the reference lane.

Performance. In the first experiment, the SDV is supposed to follow the global route. This route always navigates the agent towards the second exit of the roundabout. During training and testing, 20 other vehicles are spawned at random locations in the vicinity of the roundabout. Testing is performed on 50 randomly generated scenarios, and the results are reported in Table 1. Notice, that the graph-based approaches outperforms the BEV image-based approaches consistently across all metrics.

Generalization. An additional experiment, in which the agent is trained to take the second exit in a scenario with 100 obstacles *spawned over the whole map* evaluates the generalization capability of the state-representations. Note, that this scenario exhibits sparser traffic compared to the original setup. During testing the nominal exit is chosen randomly and moreover the number of agents spawned *in the roundabout* varies as illustrated in Figure 3. The graph-based approaches generalize better to a higher number of agents and achieve a more consistent SR.

4 Conclusion

This work presented a graph-based RL approach for automated driving. The method encodes different semantic information in a vector-based environment representation. The evaluation shows that the proposed approach outperforms other baselines with a grid-based state representation. Future work evaluates graph-based approaches in the offline RL setting, in which the agent learns a policy merely from a static dataset without interactions with the environment.

References

[1] C. Hoel, K. Wolff and L. Laine “Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning”. In: *Intelligent Transportation Systems Conference (ITSC)*. 2018.

[2] J. Chen, B. Yuan and M. Tomizuka “Model-free Deep Reinforcement Learning for Urban Autonomous Driving”. In: *Intelligent Transportation Systems Conference (ITSC)*. 2019.

[3] M. Huegle, G. Kalweit, B. Mirchevska, M. Werling and J. Boedecker “Dynamic Input for Deep Reinforcement Learning in Autonomous Driving”. In: *International Conference on Intelligent Robots and Systems (IROS)*. 2019.

[4] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola “Deep sets”. In: *Proc. of the Advances in Neural Information Processing Systems (NIPS)*. 2017.

[5] J. Gao et al. “VectorNet: Encoding hd maps and agent dynamics from vectorized representation”. In: *Proc. of the Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[6] A. Vaswani et al. “Attention is All you Need”. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*. 2017.

[7] P. Veličković et al. “Graph attention networks”. In *Proc. International Conference on Learning Representations (ICLR)*. 2017.

[8] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In *Proc. of the International Conference on Machine Learning (ICML)*, 2018.

[9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. “CARLA: An open urban driving simulator”. In *Proc. of the Conference on Robot Learning (CORL)*. 2017.

[10] Yarats, D., A. Zhang, I. Kostrikov, B. Amos, J. Pineau und R. Fergus: “Improving Sample Efficiency in Model-Free Reinforcement Learning from Images”. In*Conference on Artificial Intelligence (AAAI)*. 2021.

[11] G. Brockman et al.“Openai gym”. arxiv:1606.01540. 2016.