

Physics-informed Reinforcement Learning for Automated Merging in Dense Traffic

Johannes Fischer*, Alexei Trofimov[†] and Christoph Stiller[‡]

Abstract: Decision-making in interactive traffic situations is a challenging task for automated vehicles. Reinforcement learning (RL) is a promising approach to learn a driving policy from interactions with a simulator or from real driving data. However, reinforcement learning often requires many interactions with the environment and can have difficulties with generalization to unseen situations. To resolve these problems, we propose to use physics-informed deep learning to regularize the RL algorithm with a driver model. In our evaluation we show that this approach leads to improved sample efficiency and better generalization to more challenging scenarios.

Keywords: Driver Model, Merging, Physics-informed, Deep Learning, Reinforcement Learning

1 Introduction

In highly complex and interactive traffic situations it is difficult for automated vehicles to make optimal decisions. Previous work has used Reinforcement Learning (RL) to learn a cooperative policy for navigating interactive traffic situations [1]–[4]. Other works have used imitation learning to learn human-like driving behavior [5], [6].

A well-known problem with reinforcement learning is that it requires many interactions with the environment to learn a policy. Moreover, it can exhibit bad generalization when used in domains that are different from the training domain. At last, it can also be difficult for the RL algorithm to converge to the optimal policy.

In this work, we introduce a new approach to improve the sample efficiency and generalization of RL algorithms. Based on the principles of Physics-Informed Deep Learning (PIDL), we regularize a policy gradient algorithm with a physics model that approximately follows the desired behavior. PIDL is known to improve sample efficiency and generalization [7]. Furthermore, it can also help guide the policy towards the desired behavior. [8] has used PIDL with behavior cloning for automated driving.

We apply the resulting algorithm to an interactive merging scenario. For this reason, we use the Gap Approaching Intelligent Driver Model (GAP-IDM) [9] as the physical model. The GAP-IDM is a driver model designed to smoothly approach traffic gaps while considering distances to multiple vehicles. In our evaluation, we consider different traffic

*Johannes Fischer, M.Sc., is PhD student at the Institute of Measurement and Control Systems (MRT) at Karlsruhe Institute of Technology (KIT), johannes.fischer@kit.edu.

[†]Alexei Trofimov, B.Sc., is student at Karlsruhe Institute of Technology (KIT)

[‡]Prof. Dr.-Ing. Christoph Stiller is head of the Institute of Measurement and Control Systems (MRT) at Karlsruhe Institute of Technology (KIT), stiller@kit.edu.

conditions on the target lane. In particular, the policy has to navigate environments with more challenging traffic densities than the training environment.

2 Technical background

In this section, we will give a brief introduction to the reinforcement learning problem in fully and partially observable environments, the physics-informed deep learning approach and the Gap Approaching Intelligent Driver Model.

2.1 Reinforcement Learning

Sequential decision-making problems can be described as **Markov Decision Processes (MDPs)**, where the environment’s state s_t can be influenced by actions a_t at discrete time steps t . The decision-making agent follows a **policy**, which is a mapping from states to a probability distribution over actions. After each step, the agent receives a reward $\mathcal{R}(s_t, a_t)$ and the environment stochastically transitions to a new state s_{t+1} depending on the current state and the action. The goal is to find the policy π that maximizes the sum of cumulative discounted rewards $\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)$ [10].

Reinforcement Learning (RL) tackles this problem by learning a policy π through interactions with the environment. **Proximal Policy Optimization (PPO)** is an online policy gradient method that strikes a balance between exploration and exploitation while effectively constraining policy updates to become more sample-efficient [11].

2.2 Partially Observable Environments

In a **Partially Observable Markov Decision Process (POMDP)**, the agent does not have access to the environment’s state s_t but only to noisy observations o_t . To make optimal decisions, the agent has to infer a probability distribution over the environment’s state $p(s_t | a_{0:t-1}, o_{1:t})$ given the history of previous actions and observations. This distribution is called the belief b_t and is the input for a policy in a POMDP. Every POMDP can be interpreted as an MDP with beliefs b as states [12]. This so-called **belief MDP** can also be solved using RL algorithms [1].

2.3 Physics-informed Deep Learning

At the intersection between data-driven methods and model-based methods, **Physics-Informed Deep Learning (PIDL)** emerges as an approach that integrates physical knowledge into the training process of neural networks. To this end, the loss function is augmented with a term that enforces physical constraints [7].

In a standard supervised deep learning task a neural network ϕ_θ is regressed on labelled data (x_i, y_i) with the mean squared error loss $L(\theta) = \sum_i (\phi_\theta(x_i) - y_i)^2$.

If the data is known to satisfy a functional relation $f(x, y) = 0$, PIDL can be used to enforce this relation by augmenting the loss function with a term that penalizes the violation of this relation. The loss function is then given by $L(\theta) = \sum_i [(\phi_\theta(x_i) - y_i)^2 + \lambda f(x_i, y_i)^2]$, where λ is a hyperparameter to weigh the regularization with the physical relation.

2.4 Gap Approaching Intelligent Driver Model

Simple car-following driver models like the Intelligent Driver Model (IDM) [13] face challenges in interactive scenarios like lane changes or merging [9]. To address these issues, the GAP-IDM extends the IDM by considering multiple front and rear target vehicles and enabling smoothly approaching traffic gaps even when the vehicle is not initially aligned with the gap [9]. The acceleration of the GAP-IDM is given by

$$a_{\text{GAP-IDM}}(v, s_f, v_f, s_r, v_r) = a_{\text{max}} \cdot \left(1 - \left(\frac{v}{v_{\text{des}}} \right)^4 - \left(\frac{s^*(v, v_f)}{g(s_f)} \right)^2 + \left(\frac{s^*(v_r, v)}{g(s_r)} \right)^2 \right) \quad (1)$$

with the desired distance

$$s^*(v, v_f) = s_{\text{des}} + \max \left(0, vT + \frac{v(v - v_f)}{2\sqrt{a_{\text{max}} \cdot d_{\text{cmf}}}} \right)$$

where v is the ego velocity, s_f, v_f, s_r, v_r are the signed distance and velocity of the most relevant front and rear target vehicles, respectively, g is a distance rectifier and the parameters $(a_{\text{max}}, d_{\text{cmf}}, v_{\text{des}}, s_{\text{des}}, T)$ are the maximum acceleration, comfortable deceleration, desired velocity, minimum desired distance, and time headway. For this work, we use the shifted softplus rectifier $g_{\alpha, \beta}(s) = \frac{1}{\beta} \log(1 + \alpha + \exp(\beta s))$ with a sharpness parameter $\beta > 0$ and a shifting parameter $\alpha \geq 0$.

3 Autonomous Merging Problem

In this section we describe the merging scenario that is used for evaluating our automated merging approach. It represents a highway on-ramp situation with dense traffic on the main roadway, where the merging vehicle has to find a suitable gap and assess the cooperation of other drivers [1]. While human drivers can assess and manage such a situation through experience and their individual driving style, it is extremely challenging for an automated system to learn intelligent behavior in this scenario. In this section, we provide an overview on how the merging scenario is modeled. A more detailed description can be found in prior work [1], [4].

3.1 Environment Description

The environment, as visualized in Fig. 1, consists of a merge lane where the agent is placed, and a main lane with dense traffic where each vehicle has a varying level of cooperativeness, which impacts their behavior. The behavior of the vehicles on the main lane is modeled by the cooperative IDM (C-IDM) [1]. This means, they will generally follow IDM behavior with respect to the vehicle in front. Additionally, they will yield to the merging vehicle based on the time to reach the merge point (TTM) if $\text{TTM}_{\text{merge}} < c \cdot \text{TTM}_{\text{main}}$, where $c \in [0, 1]$ is their cooperation parameter. That is, if the merging vehicle is expected to reach the merge point before the main lane vehicle reaches the merge point, weighted by a factor of c . As a consequence, the agent has to show its merging intent for other vehicles to react and yield.

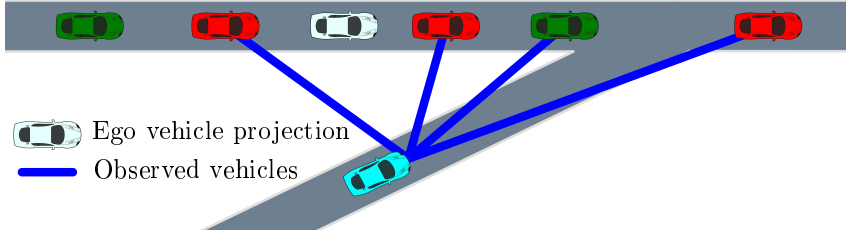


Figure 1: The ego vehicle (cyan) has to merge onto the main road where cooperative vehicles (green) might yield while non-cooperative vehicles (red) ignore it. The ego vehicle observes the vehicles most relevant for merging.

3.2 Agent Modeling

The agent observes its own physical state and the physical state of the four most relevant vehicles within its field of view. These vehicles are the vehicles before and after the merge point and before and after the projection of the ego vehicle onto the main lane, as illustrated in Fig. 1. The cooperation levels of the vehicles are not observable. Therefore, the agent must infer a belief over the cooperation to perform the merging maneuver. In this work, we use the Bayesian filter for inferring the belief that was introduced in prior work [1]. This modeling approximates the cooperation belief as a Bernoulli distribution for each vehicle, resulting in a low-dimensional belief state.

At each discrete time step of $\Delta t = 1$ s, the agent can choose between three jerk levels $\{1 \text{ m/s}^3, 0 \text{ m/s}^3, -1 \text{ m/s}^3\}$. At each time step the agent is rewarded a positive reward of +100 if reaching the goal behind the merge point, a negative reward of -100 if colliding with another vehicle, and a negative reward of $-0.1 \cdot (a_t^2 + j_t^2)$ for making use of acceleration a_t or jerk j_t . An episode terminates after reaching the goal, after a collision or after 100 time steps.

4 Physics-informed Reinforcement Learning

In our approach, we use a driver model as a physical equation to regularize a reinforcement learning policy. We begin with describing how the GAP-IDM is used as a physical equation in the merging scenario and then illustrate how this equation is used in the reinforcement learning algorithm.

4.1 GAP-IDM as Regularization

To use the GAP-IDM in a PIDL architecture, it needs to be reformulated as a function that maps the agent’s observation to an acceleration. GAP-IDM is designed to output accelerations to approach a target gap, but does not decide on which gap to target. Hence, the target gap needs to be determined from the observation beforehand.

To this end, we use a neural network to predict the target gap from the observation. We formulate this problem as a classification task, where the model decides between four possible target gaps. The first gap is always the one behind the vehicle directly in front of the merging point. The other three gaps are the subsequent gaps on the main lane, as illustrated in Fig. 2.

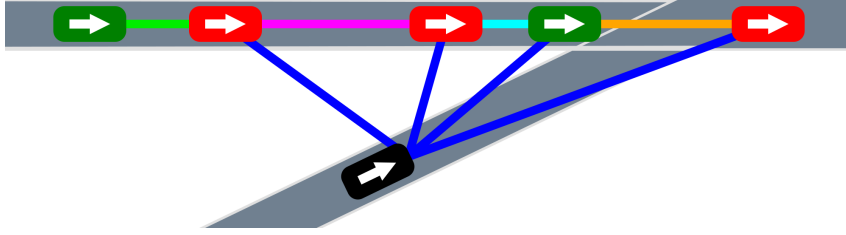


Figure 2: The four gaps considered by the ego vehicle (orange, cyan, magenta and green).

To produce human-like behavior, the classifier should be fit to human driving data. In this work, we use driving data generated with a trained PPO policy as a surrogate for human driving data. To ensure that this results in a good policy, we explicitly provide the cooperation levels of observed vehicles as an additional input to the agent. The generated data is used to train the gap classifier based on the gap chosen by the PPO policy.

4.2 Physics-informed Proximal Policy Optimization

The idea is to use the trained gap classifier to select the target vehicles for the GAP-IDM and use the GAP-IDM to regularize the reinforcement learning algorithm to produce actions that result in a behavior closer to the GAP-IDM. In this work, we will use the PPO algorithm, but the idea can be used with other policy gradient algorithms in the same way.

We incorporate the physics model $f(s, a) = 0$ into the PPO loss by an additional loss term with a regularization weight λ . For a mini-batch of states \mathcal{B} and the problem's action space \mathcal{A} , the loss term is defined as

$$L^{\text{phy}} = \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \left(\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \pi(a|s) f(s, a)^2 \right). \quad (2)$$

This residual loss term is minimized by increasing the probability $\pi(a|s)$ of selecting actions with a low mean squared model error and decreasing the probability of selecting actions with a high mean squared model error. The weight λ can be decreased during training to initially use the physics model as guidance but gradually recover the original PPO objective. We refer to the resulting algorithm as Physics-Informed Proximal Policy Optimization (PI-PPO).

4.3 PI-PPO in the Merging Scenario

To apply PI-PPO to the merging scenario, we use the error between the acceleration resulting from the chosen jerk action and the predicted acceleration of the GAP-IDM as the physics model. Furthermore, the environment is only partially observable. For this reason, we use the belief b as input to the policy, which contains the physical state observations and the inferred cooperation beliefs. Therefore, the GAP-IDM loss for the partially observable merging environment is given as

$$L^{\text{phy}} = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \left(\frac{1}{|\mathcal{A}|} \sum_{j \in \mathcal{A}} \pi(j|b) (\text{acc}^e + j \cdot \Delta t - \text{acc}^{\text{phy}}(b))^2 \right) \quad (3)$$

Parameter	Traffic condition	
	Moderate	Dense
N_{\min}	4	8
N_{\max}	8	12
p_{spawn}	1.0	0.3
$v_{\text{des},\min}$	4 m/s	4 m/s
$v_{\text{des},\max}$	6 m/s	6 m/s

Table 1: Scenario-specific parameters.

where \mathcal{A} is the action space of available jerk levels j , \mathcal{B} is a mini-batch of beliefs b , acc^e is the current ego acceleration, Δt is the time step, and $\text{acc}^{\text{phy}}(b)$ is the predicted acceleration of the GAP-IDM.

5 Experiments and Evaluation

We evaluate our approach on the merging scenario described in Section 3. To assess how well our algorithm generalizes to unseen situations, we consider different traffic conditions on the target lane. The traffic scenarios vary in the uniformly sampled number of vehicles on the main lane N , the probability that vehicles re-enter the main-lane after reaching its end p_{spawn} , and their uniformly sampled desired speed v_{des} according to Table 1. The vehicles on the main lane are simulated according to C-IDM with parameters $a_{\max} = 2 \text{ m/s}^2$, $d_{\text{cmf}} = 2 \text{ m/s}^2$, $s_{\text{des}} = 2 \text{ m}$, $T = 1.5 \text{ s}$. All agents are trained in the moderate traffic scenario and evaluated on more dense traffic compared to the training environment.

To train the gap classifier, an oracle agent with full knowledge of the latent cooperation levels of other vehicles is trained in the moderate traffic scenario. This is done to ensure that the data used to train the gap classifier is of high quality. The gap classifier is also only trained on data collected in moderate traffic.

The training parameters for PPO and PI-PPO were separately optimized using random search and are provided in Table 2. The parameters for the GAP-IDM used in PI-PPO are the same as the C-IDM parameters, except for the desired speed: We choose $v_{\text{des}} = 15 \text{ m/s}$ larger than the desired speed of the vehicles on the main lane to ensure that the agent is not incentivized to slow down before merging. The parameter values of the shifted softplus rectifier g are chosen as $\alpha = 5$ and $\beta = 0.3$.

The training progress in the moderate traffic scenario is depicted in Fig. 3. The physics-informed algorithm can be seen to converge faster compared to the uninformed PPO algorithm since the GAP-IDM loss term is able to guide the policy towards reasonable behavior. This illustrates the improved sample efficiency of the physics-informed algorithm.

To minimize deviation due to the random initialization of the traffic scene, each algorithm is evaluated on 1000 episodes in the test settings. Table 3 shows the results of the evaluation in moderate and dense traffic, measured by the average episode return, the success rate and the average episode length. Unsuccessful episodes are those that result in a collision. The performance of PI-PPO and PPO is similar on the training environ-

Parameter	Value
Neural network architecture	3 dense layers, (128, 128, 64) nodes
Activation function	ELU (except for output layer)
Epochs per batch	8
Optimizer	Adam [14]
Learning rate	$8 \cdot 10^{-4}$
Batch size	800
Training steps	$1 \cdot 10^6$
Discount factor γ	0.95
Clip ratio	0.15
Critic loss weight	0.5
Entropy regularization weight	$1 \cdot 10^{-3}$ (PI-PPO), $8 \cdot 10^{-3}$ (PPO)
Physics loss weight λ	0.2
Scheduling for λ	Linear to zero in the first 30% training steps

Table 2: Parameters used for training the PPO and PI-PPO agents.













Traffic condition	Algorithm	Episode return	Success rate [%]	Episode length
Moderate	PI-PPO	88.7 	95.8 	16.0 
	PPO	87.2 	95.3 	16.1 
Dense	PI-PPO	85.4 	94.5 	18.5 
	PPO	69.0 	86.4 	17.6 

Table 3: Evaluation results on moderate and dense traffic scenarios.

ment with moderate traffic density. However, on the more challenging test environments, PI-PPO outperforms PPO in terms of average return and success rate.

6 Conclusions and Future Work

In this work, we present a framework for physics-informed reinforcement learning in an automated merging scenario. We use the GAP-IDM as a driver model to regularize the policy gradient algorithm PPO with a policy loss term that penalizes deviations from the driver model. As our evaluation shows this leads to improved convergence properties, higher sample efficiency, and better generalization abilities to unseen traffic conditions compared to the physics-uninformed algorithm.

In our future research we want to investigate to which extent physics-informed deep learning can be used in the context of imitation learning. Since algorithms like Adversarial Inverse Reinforcement Learning employ a policy as the generator [15], PI-PPO could be used as a drop-in replacement. This could lead to faster and more stable training. Another interesting possibility is to test the algorithm on real driving data. This includes fitting the gap classifier and the driver model to real driving data and then training PI-PPO on this data.

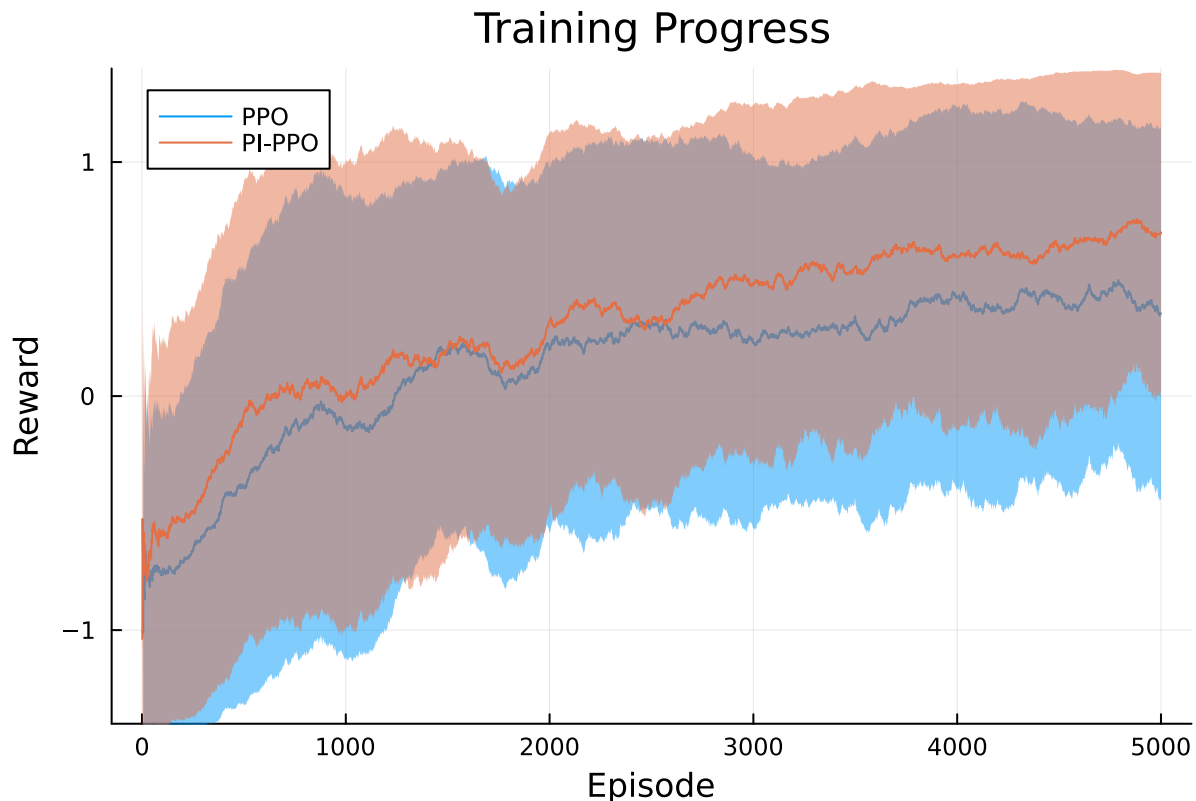


Figure 3: Moving average of the reward per episode during training with a sliding window size of $w = 250$ episodes for the PPO and PI-PPO algorithms. The darker line represents the moving average and the lighter line the moving standard deviation. Only the first 5000 episodes of training are shown.

References

- [1] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, “Cooperation-Aware Reinforcement Learning for Merging in Dense Traffic,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct. 2019, pp. 3441–3447.
- [2] D. Kamran, T. Engelgeh, M. Busch, J. Fischer, and C. Stiller, “Minimizing Safety Interference for Safe and Comfortable Automated Driving with Distributional Reinforcement Learning,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, Sep. 2021, pp. 1236–1243.
- [3] M. Sackmann, H. Bey, U. Hofmann, and J. Thielecke, “Learning a Diverse and Cooperative Policy for Predicting Roundabout Traffic Situations,” in *14. Uni-DAS e.V. Workshop Fahrerassistenz Und Automatisiertes Fahren*, 2022-05-09/2022-05-11, 2022.
- [4] J. Fischer, E. Bührle, D. Kamran, and C. Stiller, “Guiding Belief Space Planning with Learned Models for Interactive Merging,” in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Macau, China, Oct. 2022, pp. 2542–2549.

- [5] J. Fischer, C. Eyberg, M. Werling, and M. Lauer, “Sampling-based Inverse Reinforcement Learning Algorithms with Safety Constraints,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, Sep. 2021, pp. 791–798.
- [6] M. Sackmann, H. Bey, U. Hofmann, and J. Thielecke, “Modeling Driver Behavior using Adversarial Inverse Reinforcement Learning,” in *2022 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2022, pp. 1683–1690.
- [7] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019, ISSN: 00219991.
- [8] Z. Mo, X. Di, and R. Shi, “A Physics-Informed Deep Learning Paradigm for Car-Following Models,” *Transportation Research Part C: Emerging Technologies*, vol. 130, p. 103 240, Sep. 2021, ISSN: 0968090X. arXiv: 2012.13376 [cs, eess].
- [9] J. Fischer, E. Bührle, and C. Stiller, “Gap Approaching Intelligent Driver Model for Interactive Simulation of Merging Scenarios,” in *2023 IEEE Intelligent Vehicles Symposium (IV)*, Anchorage, United States, Jun. 2023, pp. 1–8.
- [10] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, Second edition, ser. Adaptive Computation and Machine Learning. Cambridge, MA London: The MIT Press, 2018, ISBN: 978-0-262-03924-6.
- [11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” *arXiv:1707.06347 [cs]*, Aug. 2017. arXiv: 1707.06347 [cs].
- [12] M. Kochenderfer, *Decision Making Under Uncertainty - Theory and Application*, 1st. MIT Press, 2015, ISBN: 0-262-02925-1 978-0-262-02925-4.
- [13] M. Treiber, A. Hennecke, and D. Helbing, “Congested Traffic States in Empirical Observations and Microscopic Simulations,” *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000, ISSN: 1063-651X, 1095-3787.
- [14] 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*, Y. Bengio and Y. LeCun, Eds., 2015.
- [15] J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.