Generalizing Decision Making for Automated Driving with an Invariant Environment Representation using Deep Reinforcement Learning

Karl Kurzer¹, Philip Schörner², Alexander Albers³, Hauke Thomsen³, Karam Daaboul¹ and J. Marius Zöllner¹

Abstract—Data driven approaches for decision making applied to automated driving require appropriate generalization strategies, to ensure applicability to the world’s variability. Current approaches either do not generalize well beyond the training data or are not capable to consider a variable number of traffic participants. Therefore we propose an invariant environment representation from the perspective of the ego vehicle. The representation encodes all necessary information for safe decision making. To assess the generalization capabilities of the novel environment representation, we train our agents on a small subset of scenarios and evaluate on the entire diverse set of scenarios. Here we show that the agents are capable to generalize successfully to unseen scenarios, due to the abstraction. In addition we present a simple occlusion model that enables our agents to navigate intersections with occlusions without a significant change in performance.

I. INTRODUCTION

Due to the increase in computational power, modern research for decision making and control has shifted from a model-based towards data-driven approach. While this paradigm shift has shown promising results, it requires careful consideration when it comes to the abstraction of the input data. This implies, that the designer of the input representation must not over-fit to the data, in order to generalize well to unseen data.

Finding the right environment representation is especially important for areas such as automated driving. While the task of driving remains largely the same all over the world, the environment might look very different. One of the biggest challenges in this domain is the crossing of unsignalized intersections, cf. Fig. 1. Conventional model-based approaches are often used for solving simple driving problems, but quickly reach their limits in more complex driving environments, since all possible situations must be anticipated and implemented in advance. A common heuristic strategy is to use the Time-To-Collision (TTC) [1] as a threshold to determine when unsignalized intersections may be crossed safely. Despite its benefits of being reliable and safe for a large number of situations, it requires manual tuning and can be difficult to scale in environments with a large number of road users and incomplete information. Especially in urban areas the complexity rises quickly and intersections are often not completely observable due to occlusions.

Therefore, the problem of crossing intersections is frequently learned with the help of Deep Reinforcement Learning [2]–[6], a reinforcement learning method that uses deep neural networks. In the past, Deep Reinforcement Learning has shown an immense potential in solving high-dimensional sequential decision making problems such as Atari [7], [8], leading up to agents capable of reaching super-human performance on a variety of extremely challenging tasks [9]. However, learning based approaches often require fixed dimensions of the input data. Hence, in the context of automated driving, the input is often limited to a fixed number of traffic participants to be considered or to a fixed discretized bird’s eye view of the environment that contains a large portion of irrelevant information.

In this work, we propose an invariant environment representation (IER), that is independent of the road layout and the number of traffic participants. Further, we train various agents using Deep Reinforcement Learning on a small set of intersection layouts and demonstrate their capability of generalizing the learned behavior to arbitrary intersection layouts.
II. RELATED WORK

Safe decision making and thus correct behavior at intersections poses a major challenge for autonomous vehicles, especially when dealing with occlusions. Recent approaches often apply Partially Observable Markov Decision Processes (POMDP) to account for occluded areas. Model-based approaches make assumptions about potentially occluded traffic participants [10], [11]. They demonstrate that POMDPs are suitable to deal with these situations. However, the size of the state space depends on the number of the known and the assumed traffic participants.

Other approaches applied to intersection scenarios which are based on Reinforcement Learning, frequently use a fixed number of traffic participants [4], [5], [12]. In addition, some represent occlusions with phantom vehicles [4], and others account for changing behaviors of other traffic participants over time with recurrent neural networks [12].

To obtain a fixed size of the state space, Isele et al. rasterize the environment using discretized Cartesian coordinates [2], [3]. For occupied cells, the cells contain information about the expected Time-To-Collision, the normalized heading angles as well as velocity of the corresponding vehicle. They also infer the optimal policy using Reinforcement Learning and validate their approach in five perpendicular intersection scenarios with a varying number of crossing lanes.

Bouton et al. present a different approach to deal with multiple objects to be considered using a scalable decision making process based on reward decomposition [13]. Two different offline POMDP solvers are evaluated in scenarios with occlusions. The computation time grows linearly with the number of other traffic participants. Another work employs scene decomposition to account for multiple traffic participants in their approach for safe reinforcement learning applying QMDPs [6]. Uncertainties, such as occlusions or perception errors, are tracked with a learned belief tracker. The output of the reinforcement algorithm is checked by a model checker to ensure safe outputs. Due to the grid-based representation of the environment the latter approaches add information to the state space that does not contribute to the decision making process.

Although the existing approaches achieve promising results and suggest the applicability of Reinforcement Learning, the ability to allow generalization is still lacking, which in turn is crucial for data-driven methods. The current state of research either restricts the intersection layouts to similar right-angled intersections or limits the number of traffic participants that can be observed by the agent. To address these limitations, we propose an invariant environment representation which is relative to the ego vehicle and accounts for an arbitrary number of vehicles.

III. PROBLEM STATEMENT

The problem of intersection navigation is formulated as a Markov Decision Process (MDP). The ego vehicle, being the agent, chooses an action in each time step. Then the agent collects an immediate reward and the system is transferred to the next state.

The MDP is described by the tuple \( \langle S, A, T, R, \gamma \rangle \) [14].
- \( S \) denotes the state space of the agent.
- \( A \) denotes the action space of the agent.
- \( T : S \times A \times S \to [0, 1] \) is the transition function with \( P(s' | s, a) \) specifying the probability of a transition from state \( s \) to state \( s' \) given the action \( a \) is chosen by the agent.
- \( R : S \times A \times S \to \mathbb{R} \) is the reward function with \( r(s, s', a) \) denoting the resulting reward of the action \( a \) taken in state \( s \).
- \( \gamma \in [0, 1] \) denotes a discount factor controlling the influence of future rewards on the current state.

The policy \( \pi \) of the agent is a mapping from the state to the probability of each available action and is given by \( \pi : S \times A \to [0, 1] \).

The goal of each agent is to maximize its expected cumulative reward in the MDP, starting from its current state: \( G = \sum_{t} \gamma^{t} r(s, s', a) \) where \( t \) denotes the time and \( G \) the return, representing the cumulated discounted rewards. \( V(s) \) is called the state-value function, given by \( V(\pi)(s) = E[G | s, \pi] \). Similarly, the action-value function \( Q(s, a) \) is defined as \( Q^{\pi}(s, a) = E[G | s, a] \), representing the expected return of choosing action \( a \) in state \( s \).

The optimal policy starting from state \( s \) is defined as \( \pi^{*} = \arg \max_{\pi} V^{\pi}(s) \). The state-value function is optimal under the optimal policy: \( \max_{\pi} V = V^{\pi} \). The same is true for the action-value function: \( \max_{\pi} Q = Q^{\pi} \). The optimal policy can be discovered by maximizing over \( Q^{\pi}(s, a) \): \( \pi^{*}(a | s) = \begin{cases} 1, & \text{if } a = \arg \max_{a \in \mathcal{A}} Q^{\pi}(s, a) \\ 0, & \text{otherwise} \end{cases} \) (1)

Once \( Q^{\pi} \) has been determined, the optimal policies can easily be derived. Thus, it is the goal to learn the optimal action-value function \( Q^{\pi}(s, a) \) for all possible state-action combinations of the MDP.

IV. APPROACH

Learning this action-value function can be achieved with Q-Learning. We specifically make use of Deep Q-Learning, which uses function approximation via Deep Neural Networks to estimate the action-value function for all possible state-action combinations [7], [8].

While automated driving as a whole is a hard task, the domain of intersections is particularly complex. During regular driving, a lane following behavior is adapted, where the focus is on the vehicle directly in front. At intersections however, traffic participants can cross the driving path from many directions at different distances simultaneously, hence planning needs to incorporate not just one but multiple traffic participants from different angles. As intersections offer a high degree of complexity, the timing of crossing needs to be precise to ensure safety. In order to learn a policy that is invariant given the environment, i.e. it is generally applicable to any intersection layout, the input representation needs to discard information regarding the road layout.
Therefore, we propose an input representation relative to the ego vehicle and its trajectory using a Frenet coordinate system \cite{15}, in order to ensure an invariant representation of the environment.

A. Invariant Environment Representation

The invariant environment representation (IER) is inspired by other occupancy based approaches, which represent navigable space employing a map representation with specific values \cite{2}, \cite{3}, \cite{16}. The path in front of the ego vehicle is discretized into so called patches. The discretization length depends on the desired look ahead distance. However, it does not only encode the occupancy information, but includes the following values, namely the Time-To-Occupancy (TTO) and the Time-To-Vacancy (TTV), which are both based on the current velocity (constant velocity assumption\(^1\)) of other traffic participants as well as the ego vehicle’s velocity, see Fig. 2. The Time-To-Occupancy (TTO) refers to the time\(^2\) when a patch becomes occupied, the Time-To-Vacancy (TTV) to the time when a patch becomes vacant.

The TTO is defined by (2):

\[
TTO = \frac{s_{\text{start}}}{v},
\]  

(2)

where \(s_{\text{start}}\) is the distance on the traffic participant’s path until the first intersection of the traffic participant (i.e. the front) with the corresponding patch, and \(v\) the velocity of the traffic participant.

The TTV is defined by (3):

\[
TTV = \frac{s_{\text{end}}}{v},
\]  

(3)

where \(s_{\text{end}}\) is the distance on the traffic participant’s path until the last intersection of the traffic participant (i.e. the rear) with the patch.

In order for the IER to be applicable for an arbitrary number of vehicles, multiple traffic participants are unified to one, if the time gap between two traffic participants traveling on the same path is less than a velocity dependent threshold, i.e. there is no space to pass safely. This implies that the Time-To-Vacancy always belongs to the last traffic participant of the union. In addition, we use the indicator function \(\delta_{\text{intersection}}\) to encode information about the start of the intersection for some experiments, see V-B.

Each patch of the IER is fully described by the following values:

- \(TTO_{\text{other}}\)
  Time-To-Occupancy of this patch by other traffic
- \(TTV_{\text{other}}\)
  Time-To-Vacancy of this patch by other traffic
- \(TTO_{\text{other next}}\)
  Next Time-To-Occupancy of this patch by other traffic after the last traffic participant of the union
- \(TTO_{\text{ego}}\)
  Time-To-Occupancy of this patch by the ego vehicle

\(^1\) More advanced approaches for the velocity prediction of other traffic participants could easily be employed, but were not explored.

\(^2\) The term time is used as a synonym for duration.

Fig. 2: Each patch encodes the time when it is going to be occupied by another vehicle, \(TTO_{\text{other}}\) and when it is vacant again, \(TTV_{\text{other}}\), as well as \(TTO_{\text{other next}}\) and \(TTO_{\text{ego}}\) for the ego vehicle. In addition \(\delta_{\text{intersection}}\) is used to indicate that an intersection starts at the respective patch (red). For illustrative purposes, the TTO and TTV sections in the patches are using a color scale from red \((t \rightarrow 0)\) to green \((t \rightarrow 1)\).

- \(\delta_{\text{intersection}}\)
  Intersection bit indicating whether this patch is the first patch of an intersection

In this work, the IER partitions the driving path in front of the ego vehicle along its center line into 50 patches of a length of 1 m and a width corresponding to the width of the ego vehicle. During our work we learned, that we achieved a higher success rate, when we encoded only the first patch of an intersection with another vehicle’s driving path along the ego vehicle’s driving path. Hence we chose this encoding for all experiments. Both Time-To-Occupancy and Time-To-Vacancy are clamped and normalized with \(t_{\text{max}} = 10\) s, cf. (4). The value of \(t_{\text{max}}\) corresponds to a look-ahead of almost 140 m at typical urban velocities of 50 km h\(^{-1}\). In case the patch of the ego vehicle’s driving path does not intersect with another path, the respective values for the Time-To-Occupancy and the Time-To-Vacancy are set to 1:

\[
t = \min\left(\frac{t_{\text{max}}}{t_{\text{max}}}, [0, 1]\right).
\]  

(4)

We use a simple occlusion model, using a worst case assumption enabling the agent to navigate intersections with occluded vehicles. Occluded areas of the lanes are assumed to be occupied by a vehicle with the length of the occlusion on the respective lane. Further, we model this occluded vehicle driving at the speed limit. The encoding in the IER commences in the same way as for non occluded vehicles.

B. Action Space

As we employ Deep Q-Learning, we require a discrete action space. The agent can choose among the following actions in order to control its velocity. Each action is being executed over a step length of 0.4 s.

- Accelerate: 3 m s\(^{-2}\)
- Maintain: 0 m s\(^{-2}\)
- Decelerate: −3 m s\(^{-2}\)
C. Reward Function

The reward function (5) consists of three distinct parts, namely a reward for collision, velocity and acceleration:

\[
    r(s, s', a) = r_{\text{collision}}(s') + r_{\text{velocity}}(s) + r_{\text{acceleration}}(a). \tag{5}
\]

The collision reward (6) includes all collision states as well as states that are deemed a near-collision, i.e. the agent crosses the path of a traffic participant with less than 10 m lateral clearance or is approaching a traffic participant in its path closer than 1 m in longitudinal direction:

\[
    r_{\text{collision}} = -K_c (\mathbb{1}_{\text{collision}} \cup \mathbb{1}_{\text{near-collision}}). \tag{6}
\]

To encourage the agent to drive close to the speed limit of 50 km h\(^{-1}\), larger deviations from it are penalized in the velocity reward (7) with a greater penalty for exceeding the speed limit (\(k_{\text{upper}} > k_{\text{lower}}\)):

\[
    r_{\text{velocity}} = \begin{cases} 
    -k_{\text{upper}} |v - v_{\text{upper}}|, & \text{if } v > v_{\text{upper}} \\
    -k_{\text{lower}} |v - v_{\text{lower}}|, & \text{if } v < v_{\text{lower}}. 
    \end{cases} \tag{7}
\]

Lastly, the acceleration reward (8) penalizes acceleration to foster a smoother driving style:

\[
    r_{\text{acceleration}} = -K_a |a|. \tag{8}
\]

V. EXPERIMENTS

The experiments are conducted using the open source microscopic traffic simulator SUMO\(^3\) [17]. With the Traffic Control Interface from SUMO, all information which is required to compute the IER described in IV-A is retrieved at every simulation step. In order to ensure a wide variability of traffic situations, each intersecting lane uses a randomized vehicle flow. We are starting the simulation with a constant traffic density by setting the probability of emitting a vehicle flow. We are starting the simulation with a constant traffic density by setting the probability of emitting a vehicle per second to 10\% per lane. This leads to a density of approximately 7 vehicles/km and lane. The traffic flow is reduced after 30 s to ensure that situations arise where the agent could pass the intersection safely. This is done by reducing the emitting probability to 5\% (approx. 4 vehicles/km) per lane. It is important to note that all non-ego vehicles controlled by SUMO are set to ignore the ego vehicle. This means that they do not brake to avoid collisions. This is necessary in order for the agent to avoid learning a policy where it would exploit the fact that other agents brake for it, rather than solving the scenario due to its own actions.

A. Scenarios

Figure 3 depicts nine out of thirteen scenarios that the agents were evaluated on. The scenarios were designed with the goal of creating a diverse set in order to be able to demonstrate the applicability of our IER to the varied situations that exist in the real world. The complexity increases from scenario 1 to scenario 9. Scenario 10 and scenario 11 are variations of scenario 1 and scenario 5 with slower traffic. Scenario 12 and 13 are similar to scenario 2, with scenario 12 being a Y-shaped one lane intersection with three directions and scenario 13 an intersection with two lanes from the top and one lane from the bottom. In contrast to related work, we position our agent further away from the intersection (100 m). This expands the task of the agent from a mere stop-or-go decision to velocity control, which results in a less jerky and more human-like decision making, considering how humans adapt their velocity when approaching an intersection with respect to the traffic. In addition, this expands the capabilities of our agent to car following scenarios, see Fig. 3 (Sc01).

B. Training

With the goal to test the generalization capabilities of our IER, we train five different agents on a small subset (Sc02 and Sc07) of all thirteen scenarios. The first three agents are trained on scenarios with increasing complexity and variability, but without occlusions and the \(\mathbb{1}_{\text{intersection}}\) missing in the state space. Based on the results of these three agents, cf. Table III, we choose the agent with the best combination of success rate and early termination rate for the training and evaluation with occlusions.

- \text{agent}_1: Trained on an intersection with a single lane per direction (Fig. 3 (Sc02))
- \text{agent}_2: Trained on an intersection with three lanes per direction (Fig. 3 (Sc07))
- \text{agent}_3: Trained on scenario 2 and scenario 7
- \text{agent}_4: \text{agent}_2 trained with occlusions, but without \(\mathbb{1}_{\text{intersection}}\)
- \text{agent}_5: \text{agent}_2 trained without occlusions, but with \(\mathbb{1}_{\text{intersection}}\)

We utilize an \(\epsilon\)-greedy policy and decay \(\epsilon\) to \(\epsilon_{\text{final}}\) over the course of the first 30\% of the training. Additionally, we employ several powerful extensions to the original DQN [18], such as prioritized experience replay [19] as well as Double-Q learning [20], to improve the stability of the training and alleviate the over-estimation of action-values in Q-learning, respectively.

The most important hyperparameters for the DQN are summarized in Table I. The training is conducted using Stable Baselines [21] in combination with TensorFlow [22].

| Parameter                          | Value         |
|-----------------------------------|---------------|
| Learning rate                     | 2e-4          |
| Discount factor                   | 0.99          |
| \(\epsilon_{\text{final}}\)       | 0.05          |
| Buffer size                       | 50,000        |
| Batch size                        | 256           |
| Double-Q learning                 | true          |
| Prioritized experience replay      | true          |
| MLP network size                  | (60, 60)      |
| Training steps                    | 5M            |

\(1\)https://www.eclipse.org/sumo/
VI. EVALUATION

All agents are evaluated on the scenarios presented in Fig. 3 as well as Sc10-Sc13. Each agent is evaluated for 1000 episodes on each scenario. Occlusions are positioned randomly on both the left and right side of the junction in 100% of all evaluation episodes, cf. Fig 1. A video with the performance of agent 4 on Sc01-Sc09 can be found online 4. The parameters used for our environment are listed in Table II. For reference, we provide a tuned TTC-based agent, that performs similar to our best agent, cf. Table III and Table IV. Here, for each lane, the absolute difference between the Time-To-Occupancy of the intersecting and the

4https://url.kurzer.de/IER

ego vehicle is calculated: $|TTO_{\text{other}} - TTO_{\text{ego}}|$. The baseline agent accelerates to a maximum of 50 km h$^{-1}$ whenever all differences are greater than 1.6 s and decelerates otherwise. In contrast to the constant velocity assumption of the RL agents, $TTO_{\text{ego}}$ is calculated for the TTC-based agent using a constant acceleration assumption.

A. Metrics

The key metric, which is used to evaluate the agent’s performance, is the success rate (SR). It describes the fraction of successful intersection traversals, i.e. reaching the other side of the intersection, while avoiding a collision or near-collision. The frequency with which a collision or near-collision occurs is denoted as early termination rate (ETR). Some agents adopt a policy to decelerate to a standstill before the intersection and wait for an opportunity to pass. When this happens the episode terminates after a maximum of 250 steps (with a step length of 0.4 s, resulting in a maximum episode duration of 100 s), thus: $1 - \text{SR} \neq \text{ETR}$.

B. Results

The results are summarized in Table III (generalization) and Table IV (occlusion). Since other vehicles are not braking for the ego vehicle, some collisions would not occur in real traffic. While this results in a fairly high early termination rate, it does not affect the comparability of the trained agents.
With regard to the generalization capabilities it can be seen, that agent$_1$, which was trained solely on Sc02, performs worse than agent$_2$ and agent$_3$. It outperforms agent$_2$ solely on scenarios with a single lane per direction, which was to be expected. The best performing agent is agent$_2$ that was trained on Sc07 (when considering SR and ETR at the same time). This is likely due to the higher complexity in the experienced states. Similarly, agent$_3$ performs well, and improves upon agent$_2$ in single lane scenarios. Overall, agent$_1$ seems less cautious, since its early termination rate is considerably higher than it is for the other agents, probably because of the less challenging traffic situation depicted in (Sc02). Furthermore, this indicates why there is an increase of almost 48% in early terminations from agent$_2$ to agent$_3$.

The results show that the agents performance on previously unseen scenarios is comparable to the performance of the training scenarios, given that the complexity in the training scenario is sufficiently high. This provides a strong indication that their ability to generalize from a small subset of scenarios to the others is due to the invariant environment representation.

The evaluation with occlusions demonstrates the usefulness of our simple occlusion model. The performance of agent$_3$ drops considerably, which has not experienced any occlusions during training. In contrast, agent$_4$ performs best overall, which was to be expected since it has experienced occlusions during training. The agent, which was trained using $\text{intersection}_3$, agent$_5$ performs significantly better than the other agents. Without occlusions its mean SR and ETR are 81.1% and 13.4% respectively, indicating that it is beneficial for the agent to know where the intersection starts in case of occlusions. From the sum of the SR and the ETR in Table IV, we can further derive that agent$_1$ and agent$_2$ wait more often before the intersection for a gap, instead of creeping into the intersection, contrasting the results without occlusions, see Table III. This leads to more episodes where the maximum number of steps is reached. The agent$_3$ implicitly learns where to expect crossing vehicles and how to smartly explore the intersection. In contrast to the other agents, almost no episode ended due to reaching the end of the episode. During the exploration, the agent creeps into the intersection area in order to perceive the occluded lanes. It should be noted, that the state space of agent$_3$ is larger. Hence a larger network might improve performance, however, this was not evaluated.

### VII. CONCLUSION

In this work we presented a novel input representation for an autonomous agent to learn the task of intersection navigation. The abstraction of the environment through our invariant environment representation enables our trained agents to transfer their knowledge gathered on one intersection to arbitrary intersection layouts. This is especially useful considering the variety of layouts an autonomous agent can experience in the real world. Further, using a simple occlusion model our representation is applicable to intersections that are not fully observable.

### ACKNOWLEDGMENT

We wish to thank the German Research Foundation (DFG) for funding the project Cooperatively Interacting Autonomous Vehicles (CoInCar) within which the research leading to this contribution was conducted. The information as well as views presented in this publication are solely the ones expressed by the authors.

### REFERENCES

[1] J. C. Hayward, “Near-miss determination through use of a scale of danger,” Highway Research Record, 1972.
[2] D. Isele, A. Cosgun, and K. Fujimura, “Analyzing Knowledge Transfer in Deep Q-Networks for Autonomously Handling Multiple Intersections,” arXiv, 2017.
[3] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, “Navigating occluded intersections with autonomous vehicles using deep reinforcement learning,” in Proceedings - IEEE International Conference on Robotics and Automation. Institute of Electrical and Electronics Engineers Inc., 2018, pp. 2034–2039.
[4] D. Kamran, C. F. Lopez, M. Lauer, and C. Stiller, “Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning,” arXiv, 2020.

### TABLE III: Generalization: Success Rate (SR) and Early Termination Rate (ETR)

| Scenario | Metric | agent$_1$ | agent$_2$ | agent$_3$ | agent$_4$ | agent$_5$ | agent$_TTC$ |
|----------|--------|-----------|-----------|-----------|-----------|-----------|-------------|
| Sc01     | SR     | 99.2%     | 91.3%     | 100.0%    | 100.0%    | 0.0%      |             |
|          | ETR    | 0.2%      | 7.2%      | 0.0%      | 0.0%      | 0.0%      |             |
| Sc02     | SR     | 87.4%     | 85.3%     | 86.2%     | 100.0%    | 0.0%      |             |
|          | ETR    | 11.8%     | 9.5%      | 13.8%     | 0.0%      | 0.0%      |             |
| Sc03     | SR     | 80.4%     | 83.7%     | 81.2%     | 96.0%     | 0.0%      |             |
|          | ETR    | 18.8%     | 10.9%     | 18.8%     | 4.0%      | 0.0%      |             |
| Sc04     | SR     | 90.3%     | 89.8%     | 91.6%     | 98.5%     | 0.0%      |             |
|          | ETR    | 8.9%      | 5.3%      | 8.4%      | 1.5%      | 0.0%      |             |
| Sc05     | SR     | 92.7%     | 88.5%     | 95.1%     | 88.0%     | 0.0%      |             |
|          | ETR    | 6.5%      | 5.5%      | 4.9%      | 12.1%     | 0.0%      |             |
| Sc06     | SR     | 83.2%     | 81.3%     | 86.3%     | 87.9%     | 0.0%      |             |
|          | ETR    | 16.1%     | 13.7%     | 13.7%     | 12.3%     | 0.0%      |             |
| Sc07     | SR     | 58.0%     | 83.9%     | 74.1%     | 73.0%     | 0.0%      |             |
|          | ETR    | 41.2%     | 10.9%     | 25.9%     | 27.0%     | 0.0%      |             |
| Sc08     | SR     | 60.5%     | 75.3%     | 69.4%     | 78.5%     | 0.0%      |             |
|          | ETR    | 38.9%     | 16.2%     | 30.7%     | 21.5%     | 0.0%      |             |
| Sc09     | SR     | 52.8%     | 64.8%     | 63.3%     | 85.7%     | 0.0%      |             |
|          | ETR    | 46.5%     | 20.7%     | 36.7%     | 14.4%     | 0.0%      |             |
| Sc10     | SR     | 39.3%     | 97.2%     | 100.0%    | 88.5%     | 0.0%      |             |
|          | ETR    | 60.7%     | 2.8%      | 0.0%      | 11.5%     | 0.0%      |             |
| Sc11     | SR     | 80.9%     | 80.4%     | 92.8%     | 90.4%     | 0.0%      |             |
|          | ETR    | 16.4%     | 5.8%      | 7.2%      | 9.7%      | 0.0%      |             |
| Sc12     | SR     | 89.8%     | 89.6%     | 93.8%     | 82.7%     | 0.0%      |             |
|          | ETR    | 9.4%      | 5.5%      | 6.2%      | 17.4%     | 0.0%      |             |
| Sc13     | SR     | 77.3%     | 83.7%     | 81.1%     | 71.0%     | 0.0%      |             |
|          | ETR    | 22.0%     | 11.0%     | 18.9%     | 29.0%     | 0.0%      |             |
| Mean     | SR     | 76.3%     | 84.2%     | 85.8%     | 87.7%     | 0.0%      |             |
|          | ETR    | 22.9%     | 9.6%      | 14.2%     | 12.3%     | 0.0%      |             |
### TABLE IV: Occlusions: Success Rate (SR) and Early Termination Rate (ETR)

| Scenario | Metric | SR  | agent₂ | agent₄ | agent₆ | agent₁₂ | ETR  |
|----------|--------|-----|--------|--------|--------|---------|------|
| Sc01     |        | 91.3% | 100.0% | 96.0% | 100.0% | 0.0%    | 7.2% |
| Sc02     |        | 70.9% | 91.0%  | 85.7% | 100.0% | 0.0%    | 7.5% |
| Sc03     |        | 60.7% | 87.0%  | 62.0% | 93.7%  | 39.7%   | 8.0% |
| Sc04     |        | 86.2% | 93.0%  | 86.0% | 98.3%  | 3.9%    | 7.0% |
| Sc05     |        | 88.2% | 94.9%  | 91.1% | 86.7%  | 5.6%    | 5.1% |
| Sc06     |        | 73.9% | 89.5%  | 79.2% | 87.1%  | 10.5%   | 10.6% |
| Sc07     |        | 52.9% | 77.3%  | 76.4% | 65.2%  | 7.4%    | 22.4% |
| Sc08     |        | 63.7% | 72.1%  | 47.1% | 71.3%  | 12.4%   | 27.9% |
| Sc09     |        | 41.1% | 63.3%  | 59.1% | 77.0%  | 13.4%   | 32.1% |
| Sc10     |        | 97.2% | 100.0% | 99.1% | 89.1%  | 2.8%    | 0.0%  |
| Sc11     |        | 80.3% | 91.7%  | 80.4% | 89.0%  | 7.0%    | 8.3%  |
| Sc12     |        | 76.6% | 94.0%  | 87.6% | 79.2%  | 4.2%    | 6.0%  |
| Sc13     |        | 49.6% | 83.1%  | 80.0% | 56.6%  | 6.2%    | 14.6% |
| Mean     |        | 71.7% | 87.5%  | 79.2% | 84.1%  | 7.4%    | 11.9% |

[5] S. Bai, R. Wang, D. Chen, J. Xiao, H. Zhang, and W. Liu, “A multi-task reinforcement learning approach for navigating unsignalized intersections,” in 2020 IEEE Intelligent Vehicles Symposium (IV), 2020, pp. 1583–1588.

[6] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, “Safe reinforcement learning with scene decomposition for navigating complex urban environments,” in IEEE Intelligent Vehicles Symposium, Proceedings, vol. 2019-June, pp. 1469–1476.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing Atari with Deep Reinforcement Learning,” arXiv, 2013.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” Nature, vol. 518, pp. 529–533, 2015.

[9] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver, “Mastering atari, go, chess and shogi by planning with a learned model,” arXiv, vol. 588, pp. 604–609, 2019.

[10] C. Hubmann, N. Quetschlich, J. Schulz, J. Bernhard, D. Althoff, and C. Stiller, “A POMDP planner maneuver for occlusions in urban scenarios,” IEEE Intelligent Vehicles Symposium, Proceedings, vol. 2019-June, pp. 2172–2179, 2019.