2021 IEEE Intelligent Transportation Systems Conference (ITSC)
Indianapolis, USA. September 19-21, 2021

Learning to Drive at Unsignalized Intersections using Attention-based Deep Reinforcement Learning

Hyunki Seong[1], Chanyoung Jung[1], Seungwook Lee[1], and David Hyunchul Shim[1]

Abstract— **Driving at an unsignalized intersection is a complex traffic scenario that requires both traffic safety and efficiency. At the unsignalized intersection, the driving policy does not simply maintain a safe distance for all vehicles. Instead, it pays more attention to vehicles that potentially have conflicts with the ego vehicle, while guessing the intentions of other vehicles. In this paper, we propose an attention-based driving policy for handling unprotected intersections using deep reinforcement learning. By leveraging attention, our policy learns to focus on more spatially and temporally important features within its egocentric observation. This selective attention enables our policy to make safe and efficient driving decisions in various congested intersection environments. Our experiments show that the proposed policy not only outperforms other baseline policies in various intersection scenarios and traffic density conditions but also has interpretability in its decision process. Furthermore, we verify our policy model's feasibility in real-world deployment by transferring the trained model to a full-scale vehicle system. Our model successfully performs various intersection scenarios, even with noisy sensory data and delayed responses. Our approach reveals more opportunities for implementing generic and interpretable policy models in real-world autonomous driving.**

I. Introduction

Unsignalized intersections are one of the most challenging traffic scenarios in urban environments. Individual drivers should decide whether to cross over (or turn to) their route without signalized protection at the intersections. These activities often have the potential for vehicle-to-vehicle conflicts, which leads to the fact that a majority of fatal car crashes involve intersection areas [1]. The drivers have to address three key challenges of the urban intersection environment: 1) It is a dense traffic environment where social interactions should be considered. As a human driver does, the ego vehicle should pay selective attention to the nearby vehicles with higher collision risks, estimating the other vehicles' intentions. 2) It is a multi-agent environment with partial observability. The ego vehicle does not have the full knowledge of its surrounding vehicles, such as their decisions, intentions, and states. Even directly observable states, such as the position of other vehicles, are not often obtainable due to occlusion. 3) It is a traffic scenario that requires not only safety but also efficiency. The decision of a vehicle affects its neighbors, and the influence propagates across multiple traffic flows. Therefore, overly conservative decisions can cause unnecessary braking, which generates traffic delays and phantom traffic jams.

[1]All authors are with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. hynkis@kaist.ac.kr, cy.jung@kaist.ac.kr, seungwook1024@kaist.ac.kr, hcshim@kaist.ac.kr

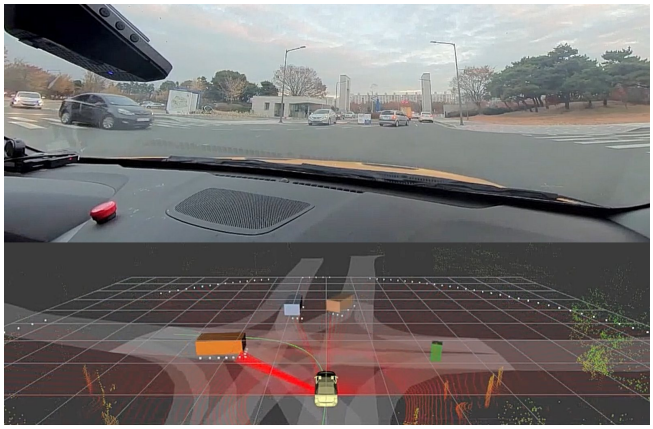


Fig. 1: **Top:** Our policy model making an unprotected left turn in a real-world deployment. **Bottom:** Attention-based policy pays more attention to a specific area (shown in red) to generate feasible and efficient driving actions. For visualization, the attention weights are smoothed with a Gaussian filter.

Most of the conventional approaches [2]–[5] use the time-to-collision (TTC) as a safety indicator to handle these challenges at intersections. These approaches are easy to design in a well-defined environment owing to the clarity of their algorithms. However, they compute TTC without considering other drivers' intentions, which can generate overly cautious behaviors and cause traffic delays at intersections.

Several strategies utilize communication and scheduling in the autonomous driving literature [6]–[9]. They can share information regarding the individual vehicle's state and environment at an intersection through communication. They perform joint scheduling with efficient control policies. However, communication-based approaches have a fundamental limitation in that expensive infrastructure should be installed, which has low real-world scalability.

The aforementioned limitations encourage the investigation of learning-based approaches to address the challenges of intersection handling. With the emergence of deep neural networks, optimal policies can be learned directly from expert data [10]–[12] or interactively from experience data in training environments [13]–[18]. Although conventional neural networks are successfully applied in various driving tasks, the human driver's selective concentration mechanism is difficult to learn. It is also unclear how the decision-making process works in neural networks, which may decrease their reliability in real-world autonomous driving.

In this paper, we propose an attention-based deep reinforcement learning (RL) framework to learn human-like driving behavior at unsignalized intersections. The attention

2021 IEEE International Intelligent Transportation Systems Conference (ITSC) | 978-1-7281-9142-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ITSC48978.2021.9564720

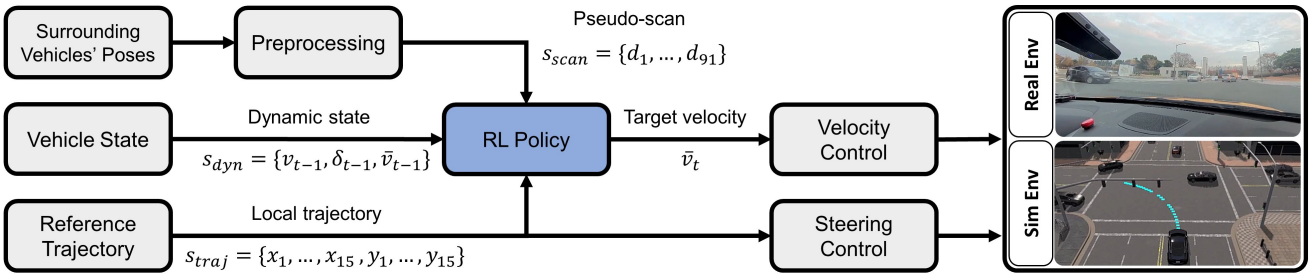


Fig. 2: The overall system architecture of our algorithm pipeline. Our policy model obtains three egocentric sensory inputs: pseudo-scan, dynamic state, and local trajectory. The policy model infers feasible target velocity commands that are fed to a low-level velocity controller. Steering angle commands are computed by a steering control module using the local trajectory. Our system pipeline is transferable from a simulated vehicle to a full-scale vehicle in the real world without major modifications.

mechanism enables our policy network to learn spatially and temporally important features within the sensory input and, thus, to focus more attention on a few relevant surrounding vehicles at intersections. This selective attention significantly improves our network's performance and interpretability, which is a major improvement over the previous learning-based approaches. We design a policy network to exploit egocentric history states for handling multiple intersection scenarios without direct access to the internal states of other vehicles. Contrary to other state-of-the-art approaches, our policy model has a notable generalization in various environments, such as 3-way, 4-way, and 5-way intersections, and roundabout, where the model does not encounter during the training procedure. Furthermore, we validate our policy model in real-world deployment by transferring the trained model from the simulation without any fine-tuning. We summarize our contributions as follows:

- We proposed a deep RL framework for driving at unsignalized intersections using a spatio-temporal attention mechanism.
- We designed a policy network which is capable of handling multiple intersection scenarios based on egocentric sensory inputs.
- We evaluated our proposed policy model in various urban intersection environments and validated our model in real-world deployment without any fine-tuning.

II. Related Work

A. Time-to-Collision-based approaches

Most of the conventional studies on intersection handling use TTC. TTC is the time required for two vehicles to collide with each other if they maintain their current velocities and the same paths [2]. Because TTC is a proximal indicator for safety, it is developed to judge when to enter an intersection [3]. It is often combined with hierarchical planning to perform rule-based decision making [4], [5]. These approaches are easy to design with human engineers' heuristic knowledge. However, they compute the TTC with the constant-velocity assumption, which ignores the intentions of other vehicles. Moreover, they cannot handle various traffic scenarios using handcrafted knowledge alone.

B. Communication-based approaches

Several frameworks use communication to schedule every vehicle at the intersection. All vehicles share their internal states through vehicular communication [6]. A centralized controller is designed to use a reservation-based approach in a multi-agent intersection environment [7] or to control each vehicle by eliminating potential conflicts between vehicles at the intersection [8]. Several studies have formulated scheduling as a model predictive control (MPC) problem to generate optimized trajectories of all vehicles in the area [9]. However, most of these frameworks require expensive infrastructure for vehicular communication. This is a major factor in reducing the scalability of communication-based methods in real-world deployment.

C. Machine Learning-based approaches

To overcome the aforementioned limitations, researchers began to use machine learning-based approaches for autonomous driving. Two major categories are commonly used: imitation learning (IL) and RL.

The goal of IL is to learn a policy directly from expert demonstrations. This approach trains an end-to-end network in a supervised manner to map sensor observations to control commands [10]. Some studies infer both steering and velocity control simultaneously by using multimodal sensory input [11] or encourage desirable behaviors by using additional losses [12]. Learning expert data in an end-to-end manner has shown preliminary success, but its performance is significantly affected by training data, which requires a large human-annotated dataset.

In contrast, RL does not suffer from the drawbacks of the imitation-based approach by leveraging reward functions for updating the policy. A number of RL applications have been demonstrated for autonomous driving tasks such as lane following [13], lane change [14], and high-speed racing [15]. For intersection driving, a deep Q-network (DQN) is applied with several sets of action maneuvers [16] and a masking mechanism for restricting its finite action space [17]. Despite their outstanding performance, the discrete action space does not fully describe complex driving strategies, which may reduce generality in urban intersection scenarios. Tram et al. designed a hierarchical method using a high-level DQN to

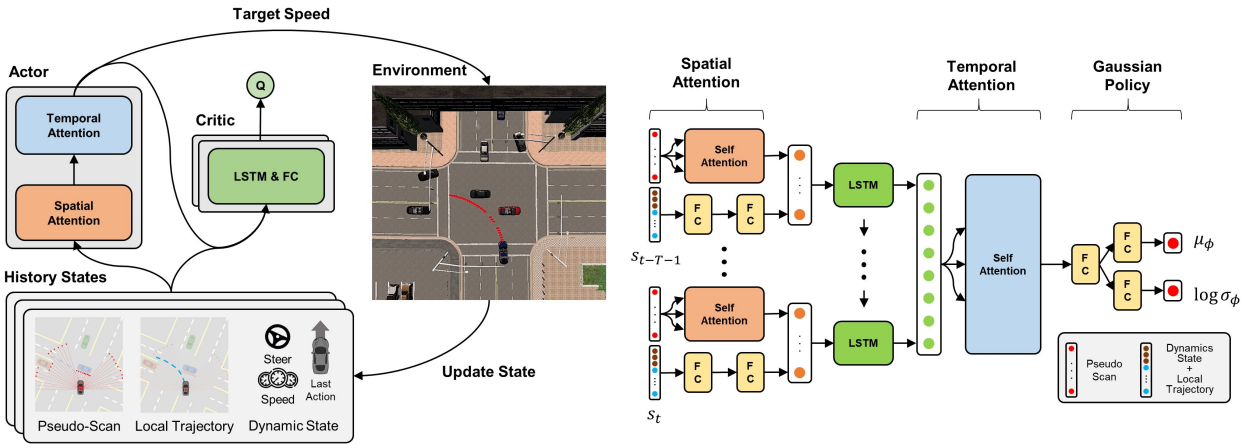


Fig. 3: **Left:** Proposed actor-critic algorithm for learning to drive at intersections. We represent three types of states: pseudo-scan, local trajectory, and dynamic state. We designed a policy model composed of spatial and temporal attention modules. A soft actor-critic architecture is used to train the policy model. **Right:** Our proposed policy network architecture based on the self-attention modules.

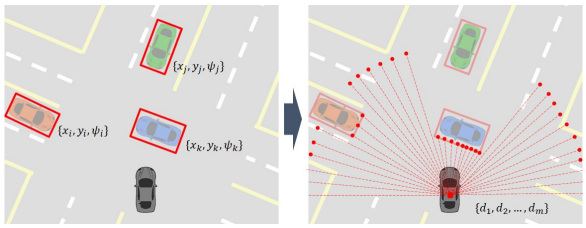


Fig. 4: Visualization of the pseudo-scan. Only the vehicles in the line of sight are reflected in the representation of the scan feature. The green vehicle, which is occluded by the blue vehicle, is not considered when the pseudo-scan is preprocessed.

set an optimization problem and a low-level MPC to compute optimized trajectories [18]. However, their method is bounded to intersection scenarios with a handful of vehicles, limiting their algorithm's scalability.

III. Methodology

A. State Representation

We represent the state of the agent in three components: $\boldsymbol{s_t} = \{s_{scan}, s_{traj}, s_{dyn}\}$, where s_{scan}, s_{traj}, and s_{dyn} are a pseudo-scan, local trajectory, and dynamic state, respectively. Fig. 4 illustrates the pseudo-scan feature $s_{scan} = \{d_1, ..., d_{91}\} \in \mathbb{R}^{91}$, which is a scan-like feature converted from the poses $p = \{x, y, \psi\}$ of the surrounding vehicles. It is a set of range measurements with a 180° semicircle area with a 2° resolution in front of the agent. This is represented by 91 range measurements within 40 m range. The scan feature is an egocentric sensory data that can handle a variable number of surrounding agents. The range data in the pseudo-scan are only measured by the vehicles in the line of sight of the ego agent. This occluded feature reflects partially observable conditions in real-world driving. The local trajectory $s_{traj} = \{x_1, ..., x_{15}, y_1, ..., y_{15}\} \in \mathbb{R}^{30}$ consists of x and y local path positions within the forward 15 m from the ego vehicle coordinate. The local trajectory is a feature of the driving scenario at the intersection. Because the trajectory can represent the geometry of various intersection scenarios, it is more scalable than discrete route information (e.g., left/right and straight). The dynamic state $s_{dyn} = \{v_t, \delta_t, a_{t-1}\} \in \mathbb{R}^3$ consists of the velocity, steering angle, and previous action of the ego agent. Using the vehicle states and the previous action, the policy model can learn the mechanism by which the policy infers the dynamics of the vehicle system [19]. All of the input features are defined as egocentric, which increases the feasibility of the policy model in real-world deployment. The final input for our policy network is a sequence of T history states.

B. Action Representation

The action $\boldsymbol{a_t}$ is defined as the target velocity $\bar{v}_t$ of the ego agent in a continuous space. It is possible to represent the sequential reasoning that the human driver conducts at intersections, such as slowing down, stopping, and speeding up after inferring the intentions of other vehicles. Because vehicles at intersections do not need to drive at high speeds, we set the range of the target velocity to $[0, 40km/h]$.

C. Attention-based Policy Network

A schematic illustration of the spatio-temporal attention-based policy network is shown in Fig. 3 (right). We implemented the attention mechanism using self-attention and integrated a spatial and temporal attention module. Spatial attention is combined with the pseudo-scan feature, and temporal attention is applied to the past T states' LSTM output features. We apply the squashed Gaussian policy [20] for our network to compute bounded continuous actions.

Self-Attention Mechanism: The mechanism is implemented to mimic a human's selective concentration on relevant information. The mechanism can be designed to map a query and a set of key–value pairs to an output, which is a weighted summation of the values. The weight assigned to each value is a similarity between the query and the corresponding key, which can be considered equal to the attention weight.

We use the self-attention module with scaled dot product attention [21] that computes the attention from the input

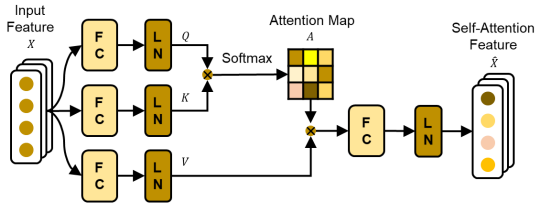


Fig. 5: Scheme of the self-attention module. The module has three fully connected (FC) layers for the query, key, and value projectors.

feature itself. The architecture of the self-attention module is illustrated in Fig. 5. In the attention module, an input matrix $X \in \mathbb{R}^{n \times f}$ consisting of n data nodes, each with an f-dimensional feature vector, is initially multiplied by three separate linear projectors to compute a query matrix Q, key matrix K, and value matrix V. The set of the normalized attention map A is then computed as

$$A(Q, K) = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{1}$$

where $Q \in \mathbb{R}^{n \times d_k}$ and $K \in \mathbb{R}^{n \times d_k}$, where $\frac{1}{\sqrt{d_k}}$ is a scaling factor and d_k is the dimension of the queries and keys in Q and K. Each projector is composed of a fully connected (FC) layer with layer normalization (LN) for stable learning [22]. The projectors learn how to compute the queries, keys, and values required for the attention weight. The value matrix $V \in \mathbb{R}^{n \times d_v}$ is multiplied by the attention matrix $A \in \mathbb{R}^{n \times n}$. The self-attention feature $\hat{X} \in \mathbb{R}^{n \times g}$ is computed by an additional FC and LN layer and then passed to the next network layer with average computation.

Spatial Attention: The policy network consists of a spatial attention branch and a feedforward branch. The attention branch infers the spatial importance of the surrounding vehicles from the pseudo-scan input s_{scan}. All the layers in the spatial attention module have FC layers with 64 neurons. Because the dynamic state s_{dyn} and the local trajectory s_{traj} do not contain any spatial information related to the surrounding actors, they are concatenated and processed by the feedforward branch with two FC layers with 64 neurons.

Temporal Attention: The output features from two branches are concatenated and fed to the LSTM with T frames of history states as shown in Fig. 3 (right). The hidden state output of the LSTM is then fed to a temporal attention module that extracts the temporal importance of the history states. The module enables our ego agent to concentrate on a few related state frames along the temporal domain to handle partially observable driving conditions (e.g., occlusion occurrence and velocities of other vehicles). The layers of the temporal attention module consisted of FC layers with 128 neurons.

Squashed Gaussian Policy: The output feature from the temporal module is passed to an FC layer of 128 units. The layer has two heads that return the mean μ_ϕ and logarithm of the standard deviation σ_ϕ defined by the reparameterization trick of the Gaussian policy [23]. Because samples from the Gaussian policy are unbounded, a squashing function, hyperbolic tangent, is applied to ensure that the actions have a finite range of $[-1, 1]$.

$$\mathbf{a}_t = \tanh(\mu_\phi + \sigma_\phi \odot \xi), \quad \xi \sim N(0, I). \tag{2}$$

The bounded actions are then denormalized to the range of the target velocity.

D. Soft Actor-Critic

We configure the soft actor–critic (SAC) network architecture with double Q-learning [20], which is shown in Fig. 3 (left). We define the attention-based policy network π_ϕ as the actor and two Q-function networks Q_{θ_1} and Q_{θ_2} as the critics. The Q-function networks are modeled with LSTM and FC layers to infer the dynamics of the ego vehicle [19].

SAC is a training algorithm that exhibits the state-of-the-art performance on a range of continuous control benchmark tasks. It is a stochastic off-policy algorithm in continuous action space, which is highly sample-efficient and stable with respect to hyperparameters. The algorithm updates the policy model π_ϕ by exploring near-optimal policies by minimizing the following objective function:

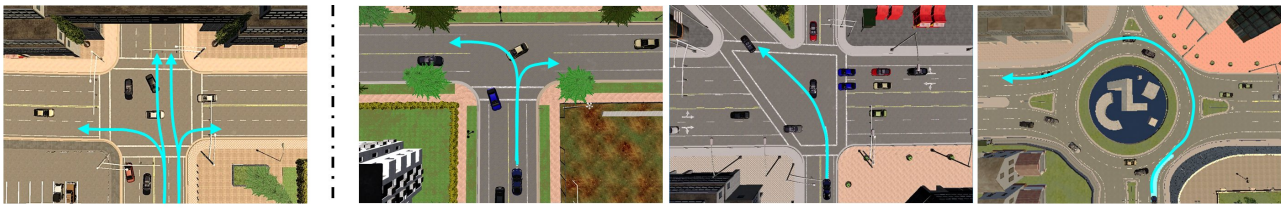
$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D}[\mathbb{E}_{a_t \sim \pi_\phi}[\alpha log(\pi_\phi(a_t|s_t)) - Q_\theta(s_t, a_t)]], \tag{3}$$

which consists of the Q-function Q_θ and the entropy of the policy with temperature parameter α. The temperature parameter determines the effect of the entropy term so that it can regulate the stochasticity of the policy. The extension [20] of the SAC automatically tunes the parameter α. In addition, the extension uses clipped double Q-learning with a delayed target Q-function $Q_{\bar{\theta}}$ [24] to reduce overestimation of Q-values and stabilize the training. Algorithm 1 summarizes the SAC algorithm, where π and θ_1 and θ_2 are the parameters for the policy network and Q-functions, respectively, and D represents a replay buffer. $\bar{\theta}_1$ and $\bar{\theta}_1$ are the parameters of the target Q-functions that are initially copied from θ_1 and θ_2 and then updated by Polyak averaging [25] with an interpolation factor, τ.

Algorithm 1 Soft Actor-Critic with Double Q-learning

Input: ϕ, θ_1, θ_2
Output: Optimized parameters ϕ, θ_1, θ_2

$\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2, D \leftarrow \emptyset$
for each episode **do**
  while not done **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $D \leftarrow D \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end while
  for each gradient step **do**
    $\theta_i \leftarrow \theta_i - \lambda_Q \nabla_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \nabla_\phi J_\pi(\phi)$
    $\alpha \leftarrow \alpha - \lambda \nabla_\alpha J(\alpha)$
    $\bar{\theta}_i \leftarrow \tau\theta_i + (1 - \tau)\bar{\theta}_i$ for $i \in \{1, 2\}$
  end for
end for



(a) 4-way (b) 3-way (c) 5-way (d) Roundabout

Fig. 6: Four environments are designed for intersection handling. The ego agent is trained in (a) and evaluated from (a) to (d). The intersection scenarios for each environment are illustrated by cyan color arrows.

(a) 3-way (b) 4-way

Fig. 7: Two environments for real-world deployment in KAIST.

E. Reward Function Design

The reward function is composed of three terms: collision penalty r_1, desired velocity r_2, and time interval r_3. r_1 encourages the ego agent to decrease the velocity at the moment of a collision so that it can minimize the physical impulse of the ego agent itself. We also add a small constant value at r_1 to penalize the actions that cause collisions with others. r_2 motivates the agent to drive at a desired velocity of $20km/h$ at intersections. r_3 encourages the ego agent to maintain a proper time interval $\Delta t = |t_{ego} - t_{other}|$ from other agents with respect to the crossing points. As the time interval increases, the potential of a collision decreases. r_3 is a dense signal that complements the drawback of the sparse reward term, r_1. All the reward terms are normalized to [0,1]. The final reward is a weighted combination of three terms: $R = w_1 r_1 + w_2 r_2 + w_3 r_3 = 100 r_1 + r_2 + r_3$.

TABLE I: Reward terms for driving at intersections, where v_t is the current ego velocity, $\bar{v}_t$ is the target velocity, and Δt is the time interval between the ego and other vehicles.

Reward Terms	Reward Functions
Collision Penalty	$r_1 = \begin{cases} -(0.1 + \frac{v_t}{40}) & \text{if collision occurs} \\ 0 & \text{otherwise} \end{cases}$
Desired Velocity	$r_2 = \frac{1}{20} \begin{cases} \bar{v}_t & \bar{v}_t \leq 20km/h \\ 40 - \bar{v}_t & \bar{v}_t > 20km/h \end{cases}$
Time Interval	$r_3 = -exp(-\Delta t)$

IV. Experiments

A. Environment Setup

We trained and evaluated the proposed policy in a simulated urban intersection using CARLA. It is a high-fidelity simulator with urban town maps where roads, traffic lights, and traffic agents are well configured [26]. We designed a training environment based on a 4-way intersection in Town 5 in CARLA. We set up an unsignalized intersection by keeping traffic lights green and controlled traffic density by adjusting the number of traffic agent spawns. To increase the scenario diversity, we randomized the relevant variables of the traffic agents, such as spawn point, driving direction, desired velocity, and controller gain. We conducted four scenarios: an unprotected left turn, straight drive in the first lane, unprotected straight drive, and a right turn in the second lane. Each episode was reset if the ego agent 1) drove at the intersection without collision, 2) collided with other vehicles, or 3) spent a maximum of 200 time steps.

The overall software architecture of the proposed system is shown in Fig. 2. We obtained the positions of the surrounding vehicles directly from the simulator and converted them to the pseudo-scan s_{scan} to feed it into the policy model. The model infers the desired target velocity $\bar{v}_t$ at 10 Hz, and a low-level proportional-integral (PI) velocity controller computed the throttle and brake commands for the ego vehicle at a higher frequency. A steering control module computed the steering commands for the vehicle to track the local path given by the intersection scenario.

B. Evaluation

We evaluated the performance of our spatio-temporal attention-based policy (STA+SAC) and compared it with those of other three baseline policies: TTC+intelligent driver model (TTC+IDM), LSTM+SAC, and FC+SAC. The TTC+IDM is a conventional method that performs the safety check using the TTC and computes proper velocity commands using the IDM [27]. All the trainable policies (STA/LSTM/FC+SAC) are trained by SAC and receive 8

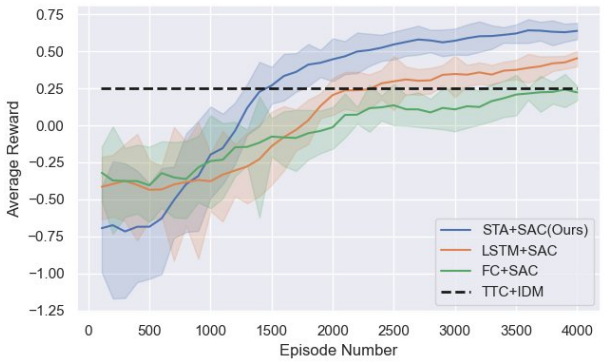


Fig. 8: Learning curves of the average reward of different evaluation policies.

TABLE II: Quantitative Performance Evaluation

Method	4-way Left Turn				4-way Straight				4-way Right Turn			
	Success (-) ↑	Colli. (-) ↓	Avg.V (km/h) ↑	BrakeT (sec) ↓	Success (-) ↑	Colli. (-) ↓	Avg.V (km/h) ↑	BrakeT (sec) ↓	Success (-) ↑	Colli. (-) ↓	Avg.V (km/h) ↑	BrakeT (sec) ↓
STA+SAC	**0.87**	**0.05**	**12.78**	**2.96**	**0.92**	**0.03**	13.15	**2.78**	**0.97**	0.02	16.07	**1.16**
LSTM+SAC	0.81	0.13	11.43	3.46	0.89	0.05	**13.38**	3.22	0.95	0.04	**16.65**	1.85
FC+SAC	0.66	0.18	8.03	5.53	0.75	0.14	7.93	5.41	0.95	0.05	14.41	2.46
TTC+IDM	0.72	0.20	9.43	4.00	0.80	0.18	9.98	5.26	0.94	**0.02**	14.65	1.98

TABLE III: Quantitative Evaluation of Robustness

Method	3-Way Left Turn				5-Way Diagonal Left Turn				Roundabout			
	Success (-) ↑	Colli. (-) ↓	Avg.V (km/h) ↑	BrakeT (sec) ↓	Success (-) ↑	Colli. (-) ↓	Avg.V (km/h) ↑	BrakeT (sec) ↓	Success (-) ↑	Colli. (-) ↓	Avg.V (km/h) ↑	BrakeT (sec) ↓
STA+SAC	0.88	0.06	13.31	1.66	0.78	0.12	13.26	2.39	0.82	0.16	15.92	1.01

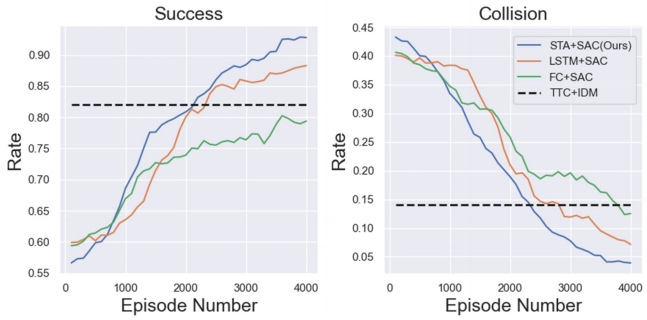


Fig. 9: Learning curves of the success/collision rates of different evaluation policies. The performance of the nontrainable policy, TTC+IDM, is shown as a dotted line.

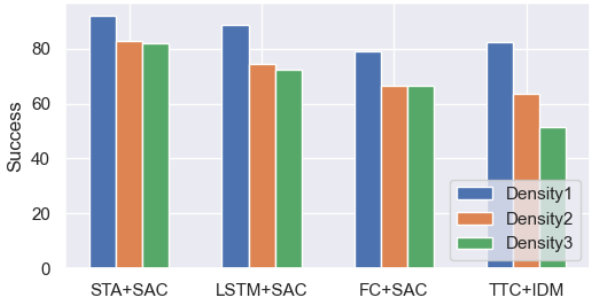


Fig. 10: Performance of different evaluation policies deployed in various traffic density environments (from Density1 to 3).

previous history states, but their network architectures are different. The LSTM+SAC has FC layers and an LSTM layer without the attention modules, and the FC+SAC has FC layers only.

1) Learning curves: We investigated the learning curves by comparing the average reward and success/collision rates of the different evaluation policies. Each trainable policy was evaluated every 100 episodes until the 4000th in the map (a) (Fig. 6). All policies performed 100 episodes of each left-turn/straight/right-turn scenario with the same number of surrounding vehicles in the environment. The results show that the learning curves of the success/collision rates are comparable to those of the average reward (Fig. 8 and 9). Our STA+SAC model learns faster and has higher reward and success rate convergence than the other evaluation policies. It shows a collision rate below 0.05, which other policies cannot reach until the terminal episode. Although the LSTM+SAC has a higher performance than the FC+SAC and the TTC+IDM, it cannot distinguish irrelevant information from sensory input and has an upper bound to the effectiveness of driving strategies. Additionally, although the FC+SAC receives the history states, similar to the other trainable policies, it is still unable to predict the surrounding vehicles properly, which leads to its performance convergence to that of the conservative TTC+IDM policy.

2) Quantitative results: We applied four metrics for the quantitative evaluation: Success (success rate), Colli. (collision rate), Avg.V (average velocity), and BrakeT (braking time). Table II summarizes the performance of the policies. Whereas the STA+SAC and the LSTM+SAC are comparable in Avg.V and BrakeT, the STA+SAC has a better performance and generalization considering all intersection scenarios. Contrary to the LSTM+SAC, the STA+SAC can utilize not only the awareness of the surrounding vehicles but also the relative importance of each car, that is, how much it should be considered. This enables the STA+SAC to have better success/collision rates and shorter brake time, which indicates that the attention-based policy has better performance and efficiency. Moreover, in the three scenarios, the STA+SAC showed no significant difference in the success rate (within 10%), while the LSTM+SAC, FC+SAC, and TTC+IDM has 14%, 29%, 22%, respectively. This demonstrates that our STA+SAC generalizes well on various intersection scenarios.

C. Robustness

1) Unseen intersection types: To evaluate the robustness of our policy to different environments, we experimented

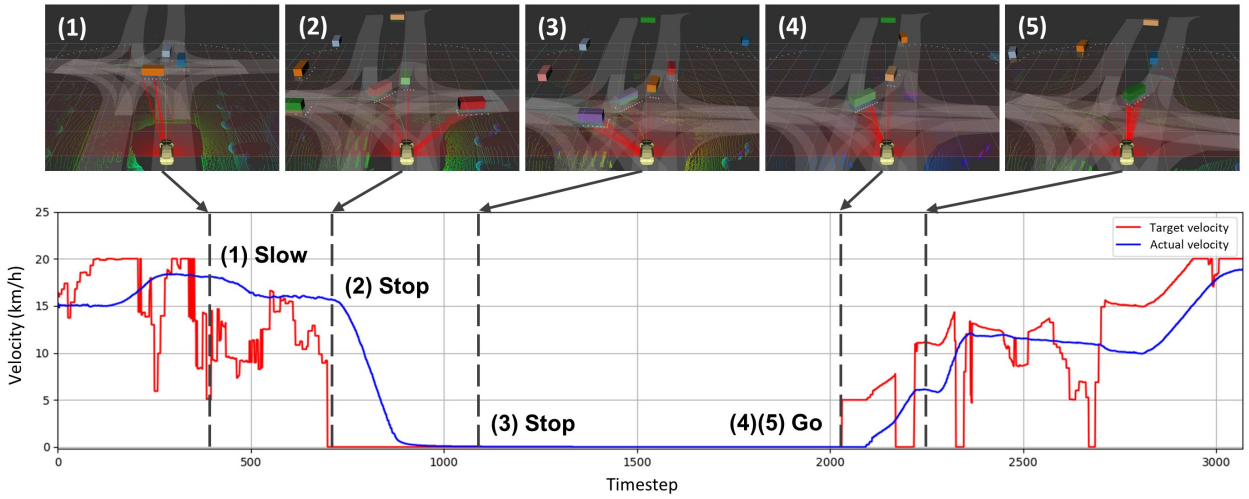


Fig. 11: Recorded trajectories of the target velocity and actual ego vehicle velocity during an unprotected left turn at the 4-way intersection.

with three different types of intersections (Fig. 6 (b-d)).

3-way intersection: The 3-way intersection is a T-shaped environment with two lanes and narrow road branches. The ego agent performed the unprotected left-turn scenario in this environment. Our policy shows a success rate of 0.88 and a collision rate of 0.06, which is close to the result of the 4-way left-turn scenario. The 3-way intersection has a smaller area than the 4-way intersection, resulting in a higher average velocity and a shorter brake time of the ego agent.

5-way intersection: The 5-way intersection is not a common environment with five road branches, where traffic becomes easily congested. The ego agent performed a diagonal left-turn scenario that did not exist in the learning environment. Our policy can extrapolate its maneuver to an unseen diagonal driving scenario. This is because the policy can leverage the local trajectory s_{traj} with the surrounding vehicle's features to negotiate with oncoming vehicles from various directions. However, because there are more road segments than in the previous training environments, the ego agent has more conflict points with the surrounding vehicles, resulting in a higher collision rate of 0.12.

Roundabout: The roundabout is a circular intersection where traffic is allowed to flow in only one direction. The ego agent should handle the unprotected right-turn scenario when entering the roundabout, followed by two additional merging scenarios with oncoming vehicles from other road branches. During each merging, the ego agent should negotiate with other vehicles converging to a counter-clockwise road, which is comparable to the unprotected left-turn. The subsequent negotiations increase the ego agent's potential collision risks, resulting in a higher collision rate of 0.16; however, our policy shows a notable generalization in different intersection geometries with a success rate exceeding 0.80.

2) Traffic density: We further studied the loss in the success rate of our model with increasing the traffic density of the intersection. We configured three types of traffic density by controlling the number of surrounding vehicle spawns. From Density1 to 3, the number of spawns is 5, 8, and 10, respectively. Density3 is the most congested environment, as all lanes are occupied by vehicles in most traffic situations. The increased traffic caused frequent congestion at intersections, which could increase the collision rate of the policies. Nonetheless, the STA+SAC was still able to achieve the highest success rate exceeding 0.80, with the lowest performance decrement (-10%). The performance of the TTC+IDM was degraded by up to -30%. This demonstrates the fundamental limitation of conventional approaches that cannot generalize well to different environments without additional tuning processes.

D. Interpretability

A visualization of the attention is illustrated in Fig. 1. With self-attention, we can explicitly interpret where the ego agent is more focused during its reasoning. During the unprotected left turn, the ego agent had a strong attention to the vehicle that would cross the ego's local trajectory (orange box, left side of the ego agent). However, the ego had little attention to the vehicle waiting for its right turn (gray box, front side). This shows that the attention-based policy model performs the observation considering not all the surrounding vehicles but only the vehicles that influence the ego agent when making decisions.

E. Real-World Deployment

To experiment with our policy model's feasibility in real-world deployment, we transferred the trained policy model in the simulation to a full-scale vehicle without any fine-tuning. The full-scale system is equivalent to the simulated one, except for additional perception and localization modules. We used a 3D object detector [28] to estimate the positions of the surrounding vehicles $p = \{x, y, \psi\}$. The localization module was implemented based on GPS and IMU data. The policy model was deployed in two general environments in KAIST (Fig. 7) and performed the unprotected left-turn, straight, and right-turn scenarios without manual traffic control. Fig. 11 illustrates the trajectories of the target velocity and actual ego vehicle velocity during an unprotected left turn

at the 4-way intersection. The policy model slowed down initially (1) and stopped the ego vehicle immediately before the intersection due to the vehicles ahead (2). The policy decided to keep stopping, as the traffic was still congested (3), and started the ego vehicle predictively when the last vehicle of the traffic exited the intersection (4). Despite noisy observations from the perception module, the policy model mainly attended to the observations' meaningful features, excluding unrelated information. The policy performed safe driving decisions, considering the delayed response of the ego vehicle system. The results of real-world deployment in various traffic scenarios are shown in the supplementary video https://youtu.be/dk4spYwvik8.

V. Conclusions

In this paper, we proposed an attention-based policy to drive at unsignalized intersections. The state is designed as a set of egocentric sensory data, and the action is represented in a continuous space for complex driving strategies. Our policy model made safe decisions based on the attention-applied egocentric information in various unprotected intersection scenarios. The experiments demonstrated the remarkable performance of the proposed policy compared with those of other baseline policies in three simulated intersection scenarios. Moreover, we demonstrated the robustness of our policy to 1) three different intersections and 2) three different traffic density environments. Furthermore, the attention-based policy visualized the direction in which the policy focused more, which led to a feasible real-world implementation. We intend to extend this work to design a generic policy that can handle not only intersections but also more diverse traffic scenarios, which is the fundamental direction of our future work. We expect that our approach will improve the scope for designing generic and interpretable policies that can be deployed in real-world urban driving.

VI. Acknowledgement

This work is the result of a research project supported by SK Hynix Inc.

References

[1] E.-H. Choi, "Crash factors in intersection-related crashes: An on-scene perspective," Tech. Rep., 2010.

[2] J. C. Hayward, "Near miss determination through use of a scale of danger," 1972.

[3] A. Sobhani, W. Young, S. Bahrololoom, M. Sarvi, *et al.*, "Calculating time-to-collision for analysing right turning behaviour at signalised intersections," *Road & Transport Research: A Journal of Australian and New Zealand Research and Practice*, vol. 22, no. 3, p. 49, 2013.

[4] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[5] A. Cosgun, L. Ma, J. Chiu, J. Huang, M. Demir, A. M. Anon, T. Lian, H. Tafish, and S. Al-Stouhi, "Towards full automated drive in urban environments: A demonstration in gomentum station, california," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1811–1818.

[6] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, 2016.

[7] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[8] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, 2012.

[9] L. Riegger, M. Carlander, N. Lidander, N. Murgovski, and J. Sjöberg, "Centralized mpc for autonomous intersection crossing," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. IEEE, 2016, pp. 1372–1377.

[10] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[11] Z. Yang, Y. Zhang, J. Yu, J. Cai, and J. Luo, "End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 2289–2294.

[12] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[13] 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 *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8248–8254.

[14] C.-J. Hoel, K. Wolff, and L. Laine, "Automated speed and lane change decision making using deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2148–2155.

[15] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Duerr, "Super-human performance in gran turismo sport using deep reinforcement learning," *arXiv preprint arXiv:2008.07971*, 2020.

[16] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2034–2039.

[17] D. Isele, A. Nakhaei, and K. Fujimura, "Safe reinforcement learning on autonomous vehicles," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–6.

[18] T. Tram, I. Batkovic, M. Ali, and J. Sjöberg, "Learning when to drive in intersections by combining reinforcement learning and model predictive control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3263–3268.

[19] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1–8.

[20] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[22] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[23] J. Schulman, N. Heess, T. Weber, and P. Abbeel, "Gradient estimation using stochastic computation graphs," *arXiv preprint arXiv:1506.05254*, 2015.

[24] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1587–1596.

[25] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[26] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[27] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[28] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.