

# CADRE: A Cascade Deep Reinforcement Learning Framework for Vision-Based Autonomous Urban Driving

Yinuo Zhao<sup>\*1</sup>, Kun Wu<sup>\*2</sup>, Zhiyuan Xu<sup>3</sup>, Zhengping Che<sup>3</sup>,  
Qi Lu<sup>3</sup>, Jian Tang<sup>†3</sup>, Chi Harold Liu<sup>1</sup>

<sup>1</sup> Beijing Institute of Technology

<sup>2</sup> Syracuse University

<sup>3</sup> Midea Group

ynzhao@bit.edu.cn, kwu102@syr.edu, {xuzhy70, chezp, luqi8, tangjian22}@midea.com, liuchi02@gmail.com

## Abstract

Vision-based autonomous urban driving in dense traffic is quite challenging due to the complicated urban environment and the dynamics of the driving behaviors. Widely-applied methods either heavily rely on hand-crafted rules or learn from limited human experience, which makes them hard to generalize to rare but critical scenarios. In this paper, we present a novel CAscade DEep REinforcement learning framework, CADRE, to achieve model-free vision-based autonomous urban driving. In CADRE, to derive representative latent features from raw observations, we first offline train a Co-attention Perception Module (CoPM) that leverages the co-attention mechanism to learn the inter-relationships between the visual and control information from a pre-collected driving dataset. Cascaded by the frozen CoPM, we then present an efficient distributed proximal policy optimization framework to online learn the driving policy under the guidance of particularly designed reward functions. We perform a comprehensive empirical study with the CARLA *NoCrash* benchmark as well as specific obstacle avoidance scenarios in autonomous urban driving tasks. The experimental results well justify the effectiveness of CADRE and its superiority over the state-of-the-art by a wide margin.

## Introduction

Autonomous driving (Dickmanns 2002; Leonard et al. 2008; Chen et al. 2015; Bojarski et al. 2016; Codevilla et al. 2018) has been widely studied over the last few decades and attracted increasing interest due to its huge potential to change the way people travel, while still remaining many technical barriers that must be conquered. Urban driving is one of the most challenging problems due to the high complexity and dynamics of the urban environment (e.g., surrounding vehicles and moving pedestrians). Given the start and target position, the goal of autonomous urban driving is to successfully complete the route within limited time and meet the requirements of pre-defined conditions, such as no collisions. Under complex urban environment and traffic conditions, it is hard for classic rule-based control approaches to handle all corner cases and avoid violations during driving.

<sup>\*</sup>These authors contributed equally.

<sup>†</sup>Corresponding author.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Recently, significant progress has been made for Imitation Learning (IL) (Pomerleau 1991; Ng, Russell et al. 2000), which enables an agent to mimic the behaviors of human experts. For vision-based autonomous driving, benefiting from IL, quite a few works focus on developing an end-to-end behavior cloning system (Codevilla et al. 2018; Sauer, Savinov, and Geiger 2018; Codevilla et al. 2019), which directly learns a mapping from the raw observations to the control actions, e.g., steer, brake, and throttle. However, as mentioned in Codevilla et al. (2019), there are several key limitations in IL for vision-based autonomous driving:

- 1) The collected dataset from human experts is biased since most of the data are simple behaviors like following the lane. Consequently, the diversity of the data decreases as the dataset grows, which reduces the generalization ability of the agent.
- 2) IL methods suffer from the *distribution shift*, i.e., the training data from experts and the testing observations in the environment are not independent and identically distributed (Bottou 2010).

Deep Reinforcement Learning (DRL) is another promising technique for model-free control and has achieved remarkable success in various complex tasks such as video games and robot control (Van Hasselt, Guez, and Silver 2016; Fujimoto, Hoof, and Meger 2018; Haarnoja et al. 2018). With the guidance of reward signals, DRL methods can alleviate the distribution shift problem through continuously interacting with the environment and effectively correcting current control policies. Some prior works (Dosovitskiy et al. 2017; Kendall et al. 2019; Sallab et al. 2017) have presented DRL-based autonomous driving frameworks, which, however, can only achieve high performance in simple scenarios, like steering control in an empty town, but fail in dense traffic scenarios. This may be due to two aspects: 1) The urban environment is very complex, especially in dense traffic containing many vehicles and pedestrians, which requires stronger perception and generalization ability of the DRL agent; 2) Unlike other control tasks (e.g., video games), the DRL agent is hard to evaluate its derived actions during the long-term and complicated behaviors only through a simple sparse reward signal.

In this work, we present a novel CAscade DEep REin-

forcement learning framework, CADRE, to realize model-free vision-based autonomous urban driving. CADRE consists of two cascade learning stages: 1) By leveraging the co-attention mechanism (Lu et al. 2016) to model the inter-relationships between visual and control information, an offline multi-task learning stage aims to train a Co-attention Perception Module (CoPM) to derive representative latent features from current raw observations. 2) With the help of the frozen CoPM and a carefully designed reward function, an online policy learning stage aims to learn optimal driving policies using distributed Proximal Policy Optimization (PPO) (Schulman et al. 2017). To the best of our knowledge, we are the first to present a successful online DRL agent on urban driving, particularly with dense traffic handling. We summarize our main contributions in the following: 1) We present a novel cascade deep reinforcement learning framework, enabling vision-based autonomous driving in complicated urban environments. 2) We propose a co-attention perception module (CoPM), which captures the inter-relationships between visual and control information, to better support the environment perception. 3) A sequential model is leveraged to capture the temporal correlations among the frame sequence, and further contributes to the decision-making for long-term driving. 4) The effectiveness and superiority of our framework are well justified by extensive experiments on *CARLA NoCrash* benchmark and specific obstacle avoidance scenarios in the CARLA simulator (Dosovitskiy et al. 2017).

## Related Work

**Vision-based Urban Autonomous Driving.** Vision-based urban driving requires an autonomous agent to finish the routes successfully mainly based on visual information. The earliest vision-based autonomous vehicles date back to Dickmanns and Zapp (1987); Pomerleau (1988). As one of the most commonly used method to train an autonomous agent, Imitation learning (IL) has been applied in many previous works (Pomerleau 1988; Bojarski et al. 2016; Codevilla et al. 2018; Sauer, Savinov, and Geiger 2018; Codevilla et al. 2019; Chen et al. 2020; Kim et al. 2020) by mimicking expert demonstrations. ALVINN (Pomerleau 1988) and DAVE-2 (Bojarski et al. 2016) learned a lane following policy which can not take a specific turn and need the driver to take over. More recently, CIL (Codevilla et al. 2018) trained an imitation learning agent conditioned on high-level commands, making the vehicle respond to navigational commands. The following work CILRS (Codevilla et al. 2019) proposed the *NoCrash* benchmark discussing the key limitations and further improved CIL by using a deeper residual architecture and a speed regularization. ChauffeurNet (Bansal, Krizhevsky, and Ogale 2019) leveraged synthesized data and additional imitation losses for dense urban driving and complex scenarios. DA-RB+ (Prakash et al. 2020) proposed an on-policy data aggregation and sampling techniques in the context of dense urban driving. Recently, Zhang et al. (2021) trained an IL agent with the supervisions from an RL coach and BEV image ground-truths. In this work, we use behavior cloning tasks to help the representative feature extraction from raw observations for sub-

sequent DRL-based agent rather than controlling the vehicle directly.

Deep Reinforcement Learning (DRL) methods try to learn a robust policy via trial and error. CIRL (Liang et al. 2018) fine-tuned a DDPG (Lillicrap et al. 2016) agent based on an imitation agent. Kendall et al. (2019) was the first work that applied a DRL agent on the real car. Having a similar goal, (Osiński et al. 2020) transferred a DRL agent trained in CARLA simulator (Dosovitskiy et al. 2017) to a real-world vehicle. There are several recent works (Jaafrā et al. 2019; Chen, Yuan, and Tomizuka 2019; Toromanoff, Wirbel, and Moutarde 2020) close to CADRE. Specifically, Chen, Yuan, and Tomizuka (2019) adapted and compared state-of-the-art model-free deep RL algorithms DDQN (Van Hasselt, Guez, and Silver 2016), TD3 (Fujimoto, Hoof, and Meger 2018) and SAC (Haarnoja et al. 2018) to the complex autonomous urban driving. IARL (Toromanoff, Wirbel, and Moutarde 2020) was the first DRL agent that can handle complex urban environment including the traffic light detection at intersections. After that, DeRL (Huang et al. 2021) improved the policy learning through predicting future transitions. Different from the above methods, we propose an on-policy DRL method with a different perception module to successfully handle dense urban traffic.

**Attention Mechanisms.** Attention mechanisms have been widely studied and achieved tremendous success in various tasks (e.g., classification, recognition and machine translation), as in Mnih et al. (2014); Jaderberg et al. (2015); Cao et al. (2015); Vaswani et al. (2017); Wang et al. (2017, 2018). Trained in an end-to-end way, attention can help the neural networks focus on a subset of inputs or latent features which is relatively important for the tasks. DANet (Fu et al. 2019) proposed the spatial and channel-wise self-attention to integrate local features and global dependencies for scene segmentation. Specifically, there is a growing number of methods using co-attention mechanisms in vision-and-language tasks like visual question answering (Lu et al. 2016; Nguyen and Okatani 2018; Wu et al. 2018). Co-attention siamese network (COSNet) (Lu et al. 2019) tried to mine the global correlations and scene context among video frames for unsupervised video object segmentation. These above methods suggest that co-attention is a good way to capture the relationships between different modalities. There are several works (Chen et al. 2017; Kim and Canny 2017; Cultrera et al. 2020) using the attention mechanisms in autonomous driving, but they did not use the co-attention between different modalities and mainly focused on the visual attention to improve the interpretability of the models. Different from the existing methods, we introduce control information from behavior cloning tasks and use co-attention to build the correlations between the visual and control information, which can help the subsequent training of the DRL-based agent.

## Methodology

In this section, we describe the cascade deep reinforcement learning framework, CADRE, as shown in Figure 1. Specifically, CADRE consists of two core components: 1)

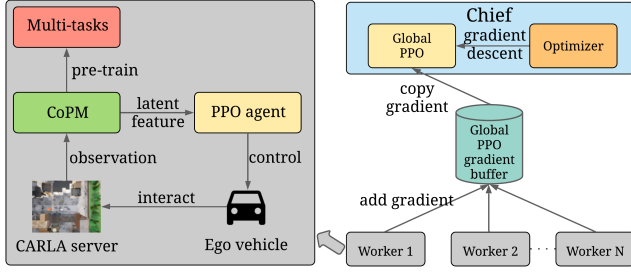


Figure 1: Overview of the CASCADE Deep REinforcement learning framework (CADRE).

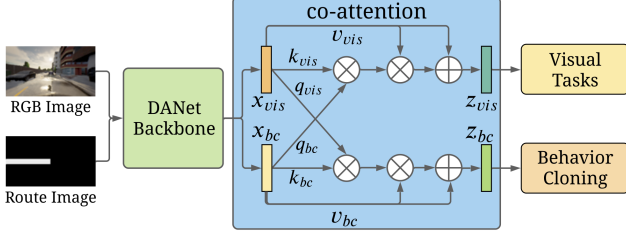


Figure 2: Overview of the Co-attention Perception Module (CoPM). Note that  $\otimes$  represents the multiplication operation and  $\oplus$  represents the addition operation.

Co-attention Perception Module (CoPM) leverages the co-attention mechanism to model the inter-relationships between the visual and control information for multi-task learning and provides representative latent features to the PPO agent. 2) Distributed Proximal Policy Optimization (PPO) agent is trained to make decision for the ego vehicle based on the latent features from CoPM using the proposed reward function.

At timestamp  $t$ , CADRE receives the observation  $\mathbf{o} = \langle \mathbf{c}, \mathbf{m}, \mathbf{w} \rangle$  from CARLA (Dosovitskiy et al. 2017), where  $\mathbf{c} \in \mathbb{R}^{3 \times H \times W}$  is an RGB image of height  $H$  and width  $W$  from the front-view camera,  $\mathbf{m}$  is a low-dimensional vector referred to as measurements containing information of the ego vehicle (e.g., position, orientation, and speed), and  $\mathbf{w} = \{w_1, \dots, w_K\}$  is a waypoint sequence given by the built-in route planner that the vehicle would follow. During the inference, given current observation  $\mathbf{o}$ , the pre-trained Co-attention Perception Module (CoPM) extracts representative latent features  $\mathbf{z}$  for the PPO agent. Afterward, the PPO agent makes a decision on the steer, throttle, and brake values  $\mathbf{a} = [a^{st}, a^{th}, a^{br}]$  for the ego vehicle.

## Offline Multi-task Learning for Perception

**Observation Preprocessing.** Given the current observation  $\mathbf{o} = \langle \mathbf{c}, \mathbf{m}, \mathbf{w} \rangle$ , one issue is that the camera image  $\mathbf{c}$ , measurement vector  $\mathbf{m}$  and waypoint sequence  $\mathbf{w}$  have different structures. To fuse the raw observations, we propose to preprocess measurements  $\mathbf{m}$  and waypoint sequence  $\mathbf{w}$  to a route image  $\mathbf{g}$  with the same size as the RGB image  $\mathbf{c}$ . Fixing the ego vehicle at the center position of the route image, we calculate the coordinates of each waypoint and connect

each pair of two adjacent waypoints by a line. Afterward, we concatenate the route image  $\mathbf{g}$  and camera image  $\mathbf{c}$  in channel and pass it into the CoPM.

**Co-attention Perception Module.** As shown in Figure 2, the CoPM, trained for the multitask learning (Caruana 1997) in an offline supervised way, aims to provide representative features  $\mathbf{z}$  of raw complex observations to the PPO agent. Taking DANet (Fu et al. 2019) as the backbone, CoPM extracts features containing both the spatial and channel interdependencies from the observations by leveraging the position attention and channel attention. Here, to gain as much visual and decision-making information as possible, CoPM has a vision branch for the visual tasks as well as a behaviour cloning branch for the control tasks, which differs from Toromanoff, Wirbel, and Moutarde (2020) predicting implicit affordance (Sauer, Savinov, and Geiger 2018). However, in most scenarios of autonomous driving, it is not necessary to focus on all global contexts of the observations (e.g., what is in the middle of the road is usually more important than the scenes on both sides). Thus, we should pay more attention to the information that influence the vehicle decision-making. Towards handling the above limitation, it is crucial to capture the inter-relationships between the visual and control information and thus we propose the co-attention bridging the vision and behaviour cloning branches.

Given the input features  $\mathbf{x}_{vis}$  and  $\mathbf{x}_{bc}$  from DANet backbone for vision and behaviour cloning branches, we obtain the keys, queries and values  $\mathbf{k}_{vis}, \mathbf{k}_{bc}, \mathbf{q}_{vis}, \mathbf{q}_{bc}, \mathbf{v}_{vis}, \mathbf{v}_{bc} \in \mathbb{R}^{d_{att} \times 1}$  respectively through non-linear blocks by

$$\mathbf{k} = \sigma(\mathbf{W}_k \mathbf{x} + \mathbf{b}_k), \quad (1)$$

$$\mathbf{q} = \sigma(\mathbf{W}_q \mathbf{x} + \mathbf{b}_q), \quad (2)$$

$$\mathbf{v} = \sigma(\mathbf{W}_v \mathbf{x} + \mathbf{b}_v), \quad (3)$$

where  $\mathbf{k} \in \{\mathbf{k}_{vis}, \mathbf{k}_{bc}\}$ ,  $\mathbf{q} \in \{\mathbf{q}_{vis}, \mathbf{q}_{bc}\}$ ,  $\mathbf{v} \in \{\mathbf{v}_{vis}, \mathbf{v}_{bc}\}$ ,  $\sigma$  is the activation function, and  $\mathbf{W}, \mathbf{b}$  are learnable parameters of the non-linear blocks. Then we calculate the scaled dot-product attention (Vaswani et al. 2017) for both branches by

$$\mathbf{a}_{vis} = \frac{\mathbf{q}_{bc} \mathbf{k}_{vis}^T}{\sqrt{d_{att}}}, \quad \mathbf{a}_{bc} = \frac{\mathbf{q}_{vis} \mathbf{k}_{bc}^T}{\sqrt{d_{att}}}, \quad (4)$$

$$\hat{\mathbf{a}}_{vis} = \text{softmax}(\mathbf{a}_{vis}), \quad \hat{\mathbf{a}}_{bc} = \text{softmax}(\mathbf{a}_{bc}). \quad (5)$$

Different from self-attention (Vaswani et al. 2017), we multiply  $\mathbf{k}_{vis}$  by  $\mathbf{q}_{bc}$  and  $\mathbf{k}_{bc}$  by  $\mathbf{q}_{vis}$ . For vision branch, the co-attention is calculated from  $\mathbf{q}_{bc}$  and  $\mathbf{k}_{vis}$ , which aims to focus on the visual information important for the decision making. For behaviour cloning branch, we use the symmetric attention queried by  $\mathbf{q}_{vis}$ , which aims to focus on decision-making information important for the visual tasks. To keep the original information for both branches, we add the values  $\mathbf{v}_{vis}, \mathbf{v}_{bc}$  to the attention-weighted sum and then concatenate them to latent feature  $\mathbf{z}$  for the PPO agent in the

second stage.

$$\mathbf{z}_{vis} = \mathbf{v}_{vis} + \sum_{i=1}^{d_z} \hat{\mathbf{a}}_{vis,i} \mathbf{v}_{vis,i}, \quad (6)$$

$$\mathbf{z}_{bc} = \mathbf{v}_{bc} + \sum_{i=1}^{d_z} \hat{\mathbf{a}}_{bc,i} \mathbf{v}_{bc,i}, \quad (7)$$

$$\mathbf{z} = [\mathbf{z}_{vis}, \mathbf{z}_{bc}]. \quad (8)$$

The vision branch consists of three heads  $H_{rou}$ ,  $H_{cam}$  and  $H_{lig}$  for route image reconstruction, semantic segmentation for camera image and light states classification respectively, to mine the semantic context and extract the visual information in observation  $\mathbf{o}$ . We use mean square error (MSE) loss for the reconstruction, and cross-entropy loss for the semantic segmentation and classification tasks:

$$\mathcal{L}_{rou} = \|\mathbf{y}_{rou} - H_{rou}(\mathbf{z}_{vis})\|_2, \quad (9)$$

$$\mathcal{L}_{cam} = - \sum_{c=1}^{C_{cam}} \mathbf{y}_{cam} \log H_{cam}(\mathbf{z}_{vis}), \quad (10)$$

$$\mathcal{L}_{lig} = - \sum_{c=1}^{C_{lig}} \mathbf{y}_{lig} \log H_{lig}(\mathbf{z}_{vis}), \quad (11)$$

where  $\mathbf{y}_{rou}$ ,  $\mathbf{y}_{cam}$  and  $\mathbf{y}_{lig}$  are the ground-truths,  $C_{cam}$  and  $C_{lig}$  are the number of categories for camera image semantic segmentation and light state classification.

The behaviour cloning branch consists of two heads  $H_{ste}$  and  $H_{thro}$  for the regression of the steer and throttle. We use MSE loss for both regression tasks:

$$\mathcal{L}_{ste} = \|\mathbf{y}_{ste} - H_{ste}(\mathbf{z}_{bc})\|_2, \quad (12)$$

$$\mathcal{L}_{thro} = \|\mathbf{y}_{thro} - H_{thro}(\mathbf{z}_{bc})\|_2, \quad (13)$$

where  $\mathbf{y}_{ste}$ ,  $\mathbf{y}_{thro}$  are the ground-truths for steer and throttle. Combining all the above terms together, we train the CoPM by performing multi-task learning and minimizing the full objective  $\mathcal{L}_{full}$  below:

$$\mathcal{L}_{full} = \lambda_{rou} \mathcal{L}_{rou} + \lambda_{cam} \mathcal{L}_{cam} + \lambda_{lig} \mathcal{L}_{lig} + \lambda_{ste} \mathcal{L}_{ste} + \lambda_{thro} \mathcal{L}_{thro}, \quad (14)$$

where  $\lambda_{rou}$ ,  $\lambda_{cam}$ ,  $\lambda_{lig}$ ,  $\lambda_{ste}$ ,  $\lambda_{thro}$  are the weights for each loss and are set to 0.5, 1.0, 0.1, 0.1, 0.1, respectively, in our implementation.

**Data Diversity Augmentation.** To train the CoPM, we pre-collect a dataset in the CARLA simulator. But we find that the CoPM pre-trained on the dataset collected using the built-in autopilot can not generalize well on observations that the PPO agent meets during the second stage. Because the car controlled by autopilot can always perform lane following and turning perfectly and know when to brake to avoid a collision, while the PPO agent usually deviates from the route or hits objects before it converges to the optimal driving policy. In order to collect imperfect samples and enrich the diversity of samples, we propose to add random noise to the decision made by autopilot with a probability of 0.7:

$$a^{st} := a^{st} + \lambda_{st}(2 \cdot noise - 1), \quad (15)$$

$$a^{th} = 0.75, \text{ if } a^{th} < 0.3, \quad (16)$$

where  $a^{st}$  and  $a^{th}$  are the steer and throttle returned by the autopilot,  $\lambda_{st}$  is set to 10 in our implementation and  $noise$  is a variable from uniform distribution  $U(0, 1)$ . Note that  $a^{st}$  will be clipped to  $[-1, 1]$  once it is out of range.

## Distributed Proximal Policy Optimization for Control

Given the latent features  $\mathbf{z}$  at time timestamp  $t$  from the frozen CoPM, to emphasize the measurement vector  $\mathbf{m}$  containing the steer, throttle and brake values at timestamp  $t-1$ , ego vehicle's linear velocity  $v$ , route deviation degree  $\theta$  and route deviation distance  $d$  at timestamp  $t$ , which are important for making decision, we explicitly concatenate  $\mathbf{m}$  and  $\mathbf{z}$  to  $\hat{\mathbf{z}}$  as the state for the PPO agent. To help the agent learn a good policy (e.g, decelerate quickly to avoid collision), given the continuous action space  $\mathbf{a} = [a^{st}, a^{th}, a^{br}]$  for the steer, throttle, and brake, we discretize them respectively. In CARLA, a vehicle's basic movement is controlled by three different values, i.e., steer value  $a^{st} \in [-1.0, 1.0]$ , throttle value  $a^{th} \in [0.0, 1.0]$  and brake value  $a^{br} \in [0.0, 1.0]$ . Throttle value will be automatically set to 0 if brake value is larger than 0. For the steer space, we discretize it into 33 bins from -1 to 1 evenly, where -1 corresponds to fully turning left and 1 corresponds to fully turning right. For the throttle and brake space, we merge them together and discretize it into 3 different actions: *accelerate* (throttle is 0.6 and brake is 0), *move forward* (throttle is 0 and brake is 0), and *decelerate* (throttle is 0 and brake is 1).

**Temporal Dependencies.** Through the frozen CoPM, the latent feature  $\hat{\mathbf{z}}$  contains the inter-relationships between the visual and behavior cloning information and current measurements from multi-sensors on the ego vehicle. However, we have only fully exploited the information in the current observation, which is not enough since driving is a long-term behavior. To capture the temporal dependencies among consecutive frames, we leverage a commonly used sequential model LSTM (Hochreiter and Schmidhuber 1997) to integrate information from eight consecutive frames:

$$\hat{\mathbf{z}}_t := LSTM(\hat{\mathbf{z}}_{t-7:t}), \quad (17)$$

where the subscript  $t$  represents the timestamp.

**Reward Shaping.** Our reward formulation includes two parts: 1) The agent obtains a *sparse reward* when a pre-defined event is triggered; 2) The agent obtains a *dense reward* at every timestamp. For the sparse reward, we define four types of bad events giving a penalty reward to the agent: 1) collision static objects, 2) collision vehicles or pedestrians, 3) vehicle blocked, and 4) route deviation. We also define a good event giving a bonus reward: complete the route successfully. Note that once an event happens, the current episode ends immediately and a new episode begins. Although the sparse reward is critical, it can not help the model evaluate the local action due to its sparsity. Therefore, to address the sparse reward problem, we define the dense rewards consisting of a deviation degree reward  $r^\theta$ , a deviation distance reward  $r^d$  and a velocity reward  $r^v$ . The deviation degree reward  $r^\theta$  is calculated as follows:

$$r^\theta = \max(0, 1 - \frac{\theta}{\theta_{max}}), \quad (18)$$

where  $\theta$  is the route deviation degree and  $\theta_{max}$  is the maximum threshold set to  $90^\circ$ . It indicates that  $r^\theta$  decreases with the increasing of  $\theta$ . If  $\theta$  is larger than the maximum allowed value, then the agent will get a minimum reward of 0. The deviation distance reward  $r^d$  is calculated as follows:

$$r^d = \max(0, 1 - \frac{d}{d_{max}}), \quad (19)$$

where  $d$  is the route deviation distance and  $d_{max}$  is the maximum threshold set to 2.5m. It indicates that  $r^d$  decreases with the increasing of  $d$ . If  $d$  is larger than the maximum allowed value, then the agent will get a minimum reward of 0. The velocity reward  $r^v$  is calculated as follows:

$$r^v = \begin{cases} \min(1, \frac{v}{v_{min}}), & \text{if } v < v_{target}, \\ \max(0, 1 - \frac{v - v_{target}}{v_{max} - v_{target}}), & \text{otherwise.} \end{cases} \quad (20)$$

The velocity reward indicates that the ego vehicle’s speed should follow a target speed range to achieve an optimal speed control. When there are no dynamic objects (i.e., vehicles and pedestrians) detected, we set  $v_{min}, v_{max}$  as the minimum and the maximum suggested velocity according to the traffic rules respectively, and  $v_{target}$  as the target velocity defined by  $v_{target} = (v_{min} + v_{max})/2$ . When an obstacle exists ahead, we set  $v_{min}$  as  $v$ ,  $v_{target}$  as the distance value to the obstacle,  $v_{max}$  as the maximum detection distance of the obstacle sensor. This setting indicates that the ego vehicle should shift down when it approaches the obstacle.

**Distributed Training.** Following Schulman et al. (2017), the clipped surrogate objective of the PPO agent is

$$\mathcal{L}_{clip} = \mathbb{E}_t[\min(r_t A_t, \text{clip}(r_t, 1 - \epsilon, 1 + \epsilon) A_t)], \quad (21)$$

where  $r_t = \frac{\pi(\mathbf{a}_t|\hat{\mathbf{z}}_t)}{\pi_{old}(\mathbf{a}_t|\hat{\mathbf{z}}_t)}$ ,  $\hat{\mathbf{z}}_t$  is the latent features from the LSTM,  $\mathbf{a}_t$  is the predicted action,  $A_t$  is the advantage function (i.e., the accumulated discounted rewards minus the state value  $V(\hat{\mathbf{z}}_t)$  in PPO), and  $r_t$  is the probability ratio with respect to old policy  $\pi_{old}$  and updated policy  $\pi$ . The *clip* operation removes the incentive for moving  $r_t$  out of interval  $[1 - \epsilon, 1 + \epsilon]$ , which makes the training process more stable. The goal of the second stage for CADRE is to minimize  $\mathcal{L}_{clip}$  in order to derive an optimal driving policy  $\pi$ . As shown in Figure 1, we use a chief-worker distributed architecture to achieve distributed training. Each worker contains a model and a CARLA server at local, where the PPO agent collects driving experience and computes the gradients independently. Once the global PPO gradient buffer gathers all the local gradients from each worker, the global PPO in the chief process performs a gradient descent and copy the model parameter to each local worker. Afterward, a new training iteration begins.

## Experiments

We conducted all the experiments on the latest CARLA simulator (Dosovitskiy et al. 2017) version 0.9.10. Through the detailed experiments, we aim to answer the following questions: 1) How does CADRE perform in urban driving compared with other baselines, especially on dense traffic condition? 2) For the perception subtask, is it beneficial to the

training of the DRL-based agent by introducing the control information and the corresponding inter-relationships? 3) For the control subtask, what are the strengths and weaknesses of on-policy DRL methods? What insight does CADRE provide to alleviate the weaknesses?

## Experimental Setup

**Simulator Environment.** CARLA (Dosovitskiy et al. 2017) contains high-fidelity maps with many professionally designed static objects (e.g., buildings and traffic signs), as well as dynamic objects (e.g., dozens of types of vehicles and pedestrians with different appearances). Overall it provides a convenient and realistic environment for experiments in autonomous driving.

**Evaluation Metrics.** In all of the following experiments, we use two metrics referred as *average route completion ratio* and *success rate* (Codevilla et al. 2019) to evaluate the training process and testing results respectively. Route completion is the ratio of the successful distance travelled by the vehicle to the total length of the route. The *average route completion ratio* is computed by averaging the route completion ratio results of all training routes in the training process. As defined in Codevilla et al. (2019), the success rate is 1 for each route if the ego vehicle completes the required route in a limited time range with no collisions, otherwise is 0.

**NoCrash Benchmark.** *NoCrash* benchmark is proposed in Codevilla et al. (2019) to evaluate the autonomous driving policy under various urban conditions. Basically, it contains three different traffic conditions with increasing difficulty levels as *empty* (no dynamic objects), *regular* (medium number of pedestrians and vehicles), and *dense* (a large number of pedestrians and vehicles). Besides, it defines 6 kinds of weathers and 25 routes in Town01 for training and Town02 for testing. We train our autonomous agent in the training town and evaluate the performance under the testing weathers and testing town respectively.

**Obstacle Avoidance Scenarios for Evaluation.** *NoCrash* benchmark evaluates the agent on a long route, during which every scenario is formed with the random behaviors of the vehicles and pedestrians, such as pedestrians crossing the road, which is unreproducible in some sense. Furthermore, there are 27 kinds of vehicles and 26 kinds of pedestrians in CARLA 0.9.10. The differences in the appearance of different objects (e.g., ordinary cars and large trucks) can cause the agent to act differently, which is not taken into account in *NoCrash* benchmark.

Therefore, to reduce the randomness and to evaluate the obstacle avoidance performance and inertia problem (Codevilla et al. 2019) more accurately, we designed a set of short routes with obstacle avoidance scenarios with each kind of vehicle and pedestrian in Town01 and Town02 (i.e., there are 27 and 26 kinds of routes for vehicle and pedestrian avoidance). Note that all the routes are only for evaluation. Same as in *NoCrash* benchmark, we calculate the success rates across all routes.

**Vehicle avoidance scenarios.** When the ego vehicle arrives at the trigger point, a stationary tool vehicle will be generated 20 meters ahead for a period of time in order to

block the ego vehicle. The ego vehicle has to stop in time and continue driving after the tool vehicle disappears.

*Pedestrian avoidance scenarios.* When the ego vehicle arrives at the trigger point, a pedestrian will be generated 20 meters ahead at the sidewalk in order to cross the road and block the ego vehicle. The ego vehicle has to stop in time and continue driving after the pedestrian crosses the road.

**Dataset for CoPM.** In the first stage, for training the CoPM, we collected a large and diverse dataset following the 25 training routes under three conditions defined in *NoCrash* benchmark from CARLA, using the built-in autopilot with additional random noise as mentioned before. The dataset has 315,545 samples, each containing a  $256 \times 144$  monocular RGB image, a waypoint sequence, vehicle measurements and the ground truths for the visual and behavior cloning tasks.

**Route and Scenario Augmentation in Training.** The predefined routes and traffic scenarios in *NoCrash* is not sufficient for DRL policy training. Therefore, we use a route and scenario augmentation method to boost a better driving policy. First, we cut the 25 long training routes into 112 short routes where there are even number of routes under each command. After that, we predefine the vehicle and pedestrian avoidance scenarios along these short routes in order to obtain a good driving policy. Last, we resume the ego vehicle from the failure location in each location rather than choosing a random start point to increase the number of samples which are meaningful for policy learning. It takes totally 3.2M training samples for CADRE to converge at a stable performance, which is enormously fewer than the 20M training samples required by the off policy DRL method IARL (Toromanoff, Wirbel, and Moutarde 2020). Apparently, it is an important strength of on policy method that it needs less training samples for converge.

## Results on *NoCrash* Benchmark

We compared CADRE with the state-of-the-art model-free reinforcement learning method IARL (Toromanoff, Wirbel, and Moutarde 2020) and imitation learning method LBC (Chen et al. 2020) on CARLA 0.9.10. Results are taken from Chen, Koltun, and Krähenbühl (2021). Note that on CARLA 0.9.10, IARL released a strong model trained on all towns and all weathers. In that case, this model does not have held-out testing weathers and thus we can only get the results under the training weathers.

Table 1 shows the quantitative results in terms of the success rate on *NoCrash* benchmark. For answering the first question, we observe that, on version 0.9.10, CADRE consistently outperformed LBC under test weathers on all traffic conditions. Especially, CADRE achieved 82 and 61 success rates in training and testing town respectively under dense traffic and training weathers, with 19 and 28 success rates higher than the performance achieved by LBC. It demonstrates that CADRE can drive the vehicle carefully and keep the high performance on the dense traffic condition. Even driving on the testing unseen weather, CADRE can keep the high success rates on all conditions via a good policy, while LBC dropped its performance dramatically. It is worth noting that the behavior cloning branch (3rd column in Table 1)

Task	Town	Weather	IARL	LBC	CoPM	Ours
Empty			85	89	62	<b>95</b>
Regular	Train	Train	86	87	63	<b>92</b>
Dense			63	75	70	<b>82</b>
Empty			–	60	62	<b>94</b>
Regular	Train	Test	–	60	66	<b>86</b>
Dense			–	54	72	<b>76</b>
Empty			77	86	86	<b>92</b>
Regular	Test	Train	66	<b>79</b>	70	78
Dense			33	53	39	<b>61</b>
Empty			–	36	44	<b>78</b>
Regular	Test	Test	–	36	44	<b>72</b>
Dense			–	12	30	<b>52</b>

Table 1: Results on *NoCrash* benchmark.

	Vehicle avoidance	Pedestrian avoidance
LBC	55/81	73/78
IARL	69/81	57/78
Ours	81/81	76/78

Table 2: Results on obstacle avoidance scenarios.

in the CoPM can also achieve comparable performance on all conditions. We also provide more detailed reference results for performance comparison between CADRE and the state-of-the-art on different versions of CARLA in the supplemental materials.

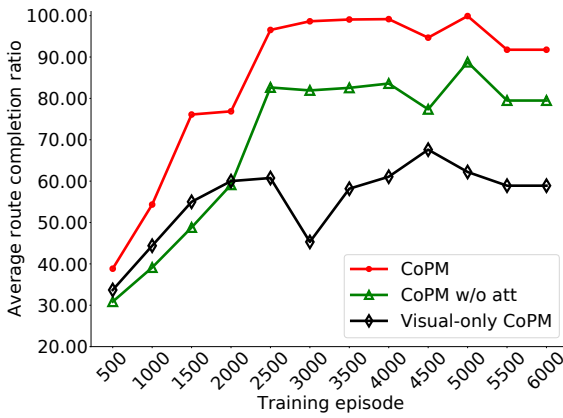
## Results on Obstacle Avoidance Scenarios

Table 2 shows the quantitative results in terms of the success rate on our designed routes with specified obstacle avoidance scenarios. Same as on the *NoCrash* benchmark, all the experiments are built on CARLA 0.9.10. We repeated the evaluation 3 times for each scenario (i.e., 81 times for vehicle avoidance and 78 times for pedestrian avoidance in total). In both vehicle and pedestrian avoidance scenarios, CADRE consistently outperformed IARL and LBC, which indicated that CADRE has a better perception ability for the obstacles.

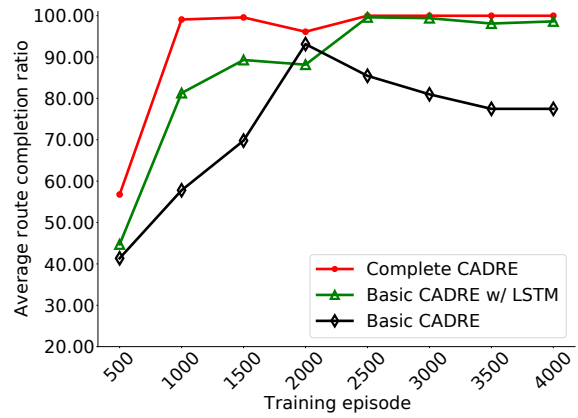
## Ablation Study

To better understand the merits of CADRE, we conducted detailed ablation studies on the CoPM and the PPO agent in Figures 3a and 3b respectively.

**Ablation Study on the CoPM.** To answer the second question and verify the effectiveness of the co-attention mechanism and the control information, we compared the complete CoPM to two simpler baselines: 1) CoPM without co-attention (CoPM w/o att) and 2) Visual-only CoPM that is trained only for visual tasks thus without co-attention. Figure 3a shows the quantitative results on the training process of the PPO agent in terms of the average route completion ratio. We can observe that the CoPM (red line) achieved a higher average route completion ratio than the CoPM w/o att (green line) and the visual-only CoPM (black line), which



(a) Training curves of the ablation study on the CoPM.



(b) Training curves of the ablation study on the PPO agent.

Figure 3: Ablation study for the CADRE.

	Vehicle avoidance	Pedestrian avoidance
Visual-only CoPM	75/81	39/78
CoPM w/o att	76/81	68/78
CoPM	81/81	76/78

Table 3: Ablation study on the CoPM on obstacle avoidance scenarios.

	Vehicle avoidance	Pedestrian avoidance
Basic CADRE	51/81	51/78
Basic CADRE w/ LSTM	71/81	61/78
Complete CADRE	81/81	76/78

Table 4: Ablation study on the PPO on obstacle avoidance scenarios.

indicated that CoPM can build the inter-relationships between the visual and control information, and extract more representative latent features to help the learning of the PPO agent. In addition, Table 3 shows the quantitative results about the CoPM in terms of the success rate on obstacle avoidance scenarios. The CoPM without co-attention outperformed the visual-only CoPM by relatively large margins, showing the control information from the behavior cloning task is beneficial to the learning of the PPO agent.

**Ablation Study on the PPO Agent.** To answer the third question, We built an ablation study to illustrate the contribution of each component in the DRL policy model and demonstrate the training process with *average route completion ratio* metric. We use three settings of CADRE as follows: 1) Basic CADRE: consisting of the CoPM with a vanilla PPO policy model; 2) Basic CADRE w/ LSTM: adding the LSTM module to capture the temporal correlations among frame sequence; 3) Complete CADRE: the basic CADRE w/ LSTM in a distributed training manner with 4 workers. As shown in Figure 3b, compared with the basic CADRE, CADRE w/ LSTM converged faster and obtained

a better result in *average route completion ratio*, which indicates that the temporal correlations among the consecutive frames are beneficial to the learning. Clearly, complete CADRE converged faster than all of the baselines and eventually stabilized to an *average route completion ratio* nearly to 100%.

It should be noticed that there is a performance drop for all these methods in episode 1500 – 2000. Complete CADRE quickly recovers from the performance drop while the basic CADRE remains at a *average route completion ratio* at 78%. This result indicates that it is possible for an on-policy method to crash into a local optimal performance and never recover. However, with the distributed training architecture and our carefully designed reward function, it can recover from the local optimal performance and remain stable.

Table 4 shows the quantitative results about the PPO agent in terms of the success rate on obstacle avoidance scenarios. We can clearly observe that both the temporal correlations from LSTM and distributed training contribute a lot to the performance improvement on obstacle avoidance scenarios.

## Conclusion

In this paper, we proposed a novel CAscade Deep REinforcement learning (DRL) framework, CADRE for vision-based autonomous urban driving. To reduce the complexity of the driving task, CADRE splits it into perception and control subtasks. In the first stage, we pre-train a Co-attention Perception Module (CoPM), which leverages the co-attention mechanism to build the inter-relationships between the visual and control information, to provide representative latent features from the raw observations to the subsequent agent. In the second stage, we use an on-policy algorithm, Proximal Policy Optimization (PPO), to train the agent in a distributed way. With the careful reward shaping and sequential model LSTM, the PPO agent learns a good policy, which can achieve high success rates in complex urban environment even on the dense traffic condition. Results on *NoCrash* benchmark and specific obstacle avoidance scenarios demonstrated the effectiveness of CADRE.

## Acknowledgments

This work was supported in part by Shanghai Pujiang Program and the National Research and Development Program of China (No. 2019YQ1700).

## References

- Bansal, M.; Krizhevsky, A.; and Ogale, A. 2019. Chauffeur-net: Learning to drive by imitating the best and synthesizing the worst. *Proc. Robotics: Science and Systems*.
- Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.
- Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, 177–186. Springer.
- Cao, C.; Liu, X.; Yang, Y.; Yu, Y.; Wang, J.; Wang, Z.; Huang, Y.; Wang, L.; Huang, C.; Xu, W.; et al. 2015. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, 2956–2964.
- Caruana, R. 1997. Multitask learning. *Machine learning*, 28(1): 41–75.
- Chen, C.; Seff, A.; Kornhauser, A.; and Xiao, J. 2015. Deep-driving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, 2722–2730.
- Chen, D.; Koltun, V.; and Krähenbühl, P. 2021. Learning to drive from a world on rails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15590–15599.
- Chen, D.; Zhou, B.; Koltun, V.; and Krähenbühl, P. 2020. Learning by cheating. In *Conference on Robot Learning*, 66–75. PMLR.
- Chen, J.; Yuan, B.; and Tomizuka, M. 2019. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2765–2771. IEEE.
- Chen, S.; Zhang, S.; Shang, J.; Chen, B.; and Zheng, N. 2017. Brain-inspired cognitive model with attention for self-driving cars. *IEEE Transactions on Cognitive and Developmental Systems*, 11(1): 13–25.
- Codevilla, F.; Miiller, M.; López, A.; Koltun, V.; and Dosovitskiy, A. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–9. IEEE.
- Codevilla, F.; Santana, E.; López, A. M.; and Gaidon, A. 2019. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9329–9338.
- Cultrera, L.; Seidenari, L.; Becattini, F.; Pala, P.; and Del Bimbo, A. 2020. Explaining autonomous driving by learning end-to-end visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 340–341.
- Dickmanns, E. D. 2002. The development of machine vision for road vehicles in the last decade. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 1, 268–281. IEEE.
- Dickmanns, E. D.; and Zapp, A. 1987. Autonomous high speed road vehicle guidance by computer vision. *IFAC Proceedings Volumes*, 20(5): 221–226.
- Dosovitskiy, A.; Ros, G.; Codevilla, F.; López, A. M.; and Koltun, V. 2017. CARLA: An Open Urban Driving Simulator. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, 1–16. PMLR.
- Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3146–3154.
- Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 1587–1596. PMLR.
- Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 1861–1870. PMLR.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.
- Huang, C.; Zhang, R.; Ouyang, M.; Wei, P.; Lin, J.; Su, J.; and Lin, L. 2021. Deductive Reinforcement Learning for Visual Autonomous Urban Driving Navigation. *IEEE Transactions on Neural Networks and Learning Systems*.
- Jaafra, Y.; Laurent, J. L.; Deruyver, A.; and Naceur, M. S. 2019. Seeking for robustness in reinforcement learning: application on Carla simulator.
- Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015. Spatial transformer networks. *Advances in neural information processing systems*, 28: 2017–2025.
- Kendall, A.; Hawke, J.; Janz, D.; Mazur, P.; Reda, D.; Allen, J.-M.; Lam, V.-D.; Bewley, A.; and Shah, A. 2019. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, 8248–8254. IEEE.
- Kim, I.; Lee, H.; Lee, J.; Lee, E.; and Kim, D. 2020. Multi-task Learning with Future States for Vision-based Autonomous Driving. In *Proceedings of the Asian Conference on Computer Vision*.
- Kim, J.; and Canny, J. 2017. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, 2942–2950.
- Leonard, J.; How, J.; Teller, S.; Berger, M.; Campbell, S.; Fiore, G.; Fletcher, L.; Frazzoli, E.; Huang, A.; Karaman, S.; et al. 2008. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10): 727–774.
- Liang, X.; Wang, T.; Yang, L.; and Xing, E. 2018. CirI: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 584–599.



- Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*.
- Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering. *Advances in neural information processing systems*, 29: 289–297.
- Lu, X.; Wang, W.; Ma, C.; Shen, J.; Shao, L.; and Porikli, F. 2019. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3623–3632.
- Mnih, V.; Heess, N.; Graves, A.; et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, 2204–2212.
- Ng, A. Y.; Russell, S. J.; et al. 2000. Algorithms for inverse reinforcement learning. In *ICML*, volume 1, 2.
- Nguyen, D.-K.; and Okatani, T. 2018. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6087–6096.
- Osiński, B.; Jakubowski, A.; Zięcina, P.; Miłoś, P.; Galias, C.; Homoceanu, S.; and Michalewski, H. 2020. Simulation-based reinforcement learning for real-world autonomous driving. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6411–6418. IEEE.
- Pomerleau, D. A. 1988. Alvin: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1.
- Pomerleau, D. A. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1): 88–97.
- Prakash, A.; Behl, A.; Ohn-Bar, E.; Chitta, K.; and Geiger, A. 2020. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11763–11773.
- Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19): 70–76.
- Sauer, A.; Savinov, N.; and Geiger, A. 2018. Conditional affordance learning for driving in urban environments. In *Conference on Robot Learning*, 237–252. PMLR.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Toromanoff, M.; Wirbel, E.; and Moutarde, F. 2020. End-to-End Model-Free Reinforcement Learning for Urban Driving using Implicit Affordances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7153–7162.
- Van Hasselt, H.; Guez, A.; and Silver, D. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.
- Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; and Tang, X. 2017. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.
- Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.
- Wu, Q.; Wang, P.; Shen, C.; Reid, I.; and Van Den Hengel, A. 2018. Are you talking to me? reasoned visual dialog generation through adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6106–6115.
- Zhang, Z.; Liniger, A.; Dai, D.; Yu, F.; and Van Gool, L. 2021. End-to-End Urban Driving by Imitating a Reinforcement Learning Coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15222–15232.