

# Uncertainty-aware Deep Imitation Learning and Deployment for Autonomous Navigation through Crowded Intersections

Zeyu Zhu, Shuai Wang and Huijing Zhao

**Abstract**—Navigation through crowded intersections is a challenge for autonomous vehicles, where uncertainty arises from interaction with other road users, encountering new scenes and weathers, etc. Recent end-to-end autonomous control deep models learned from human drivers have shown promising driving performance, whereas they are not as transparent and safe as traditional rule-based systems. When facing situations that they are unfamiliar with or uncertain about, the deep models’ predictions could be unsafe and untrustworthy. Without the ability to identify these situations and issue warnings beforehand, cascading errors of deep models may result in catastrophes. Therefore, this work combines the strengths of both data-driven and traditional rule-based approaches to achieve better driving quality and safety. We propose a heterogeneity uncertainty quantification method based on imitation learning, where both data and model uncertainties of the lateral and longitudinal control tasks are quantified. We also propose a policy deployment strategy where a safety indicator is developed upon estimated uncertainty to bridge the data-driven performance layer and the rule-based fallback layer. We learned from human driving demonstrations and conducted extensive closed-loop tests. Results demonstrate the effectiveness and importance of the proposed uncertainty quantification method and policy deployment strategy.

## I. INTRODUCTION

Inspired by the amazing advances in deep learning, recent years have witnessed tremendous efforts in end-to-end autonomous driving [1]–[3]. A bunch of vision-based autonomous control methods based on deep imitation learning have been developed [4]–[8]. These approaches learn to map raw camera images to control signals such as acceleration and steering angles. To realize multiple driving maneuvers, conditional imitation learning [4] was proposed to predict control actions conditioned on high-level commands such as turn left, turn right, go straight, etc.

Safe and efficient navigation through crowded intersections has been one of the main challenges of autonomous urban driving [9]–[11]. In such tasks, the autonomous agent needs to exercise robust control to interact with other road users. The uncertainty introduced by other road users poses a significant threat to the reliability of autonomous driving models. Although end-to-end autonomous driving systems have shown promising outcomes compared to traditional modularized counterparts [12], [13], several limitations hinders the future development and application of them. First,

This work was supported by the National Natural Science Foundation of China under Grants U22A2061.

The authors are with the National Key Lab of General Artificial Intelligence, and also with the School of Intelligence Science and Technology, Peking University, Beijing 100871, China.

Correspondence: H.Zhao, zhaohj@pku.edu.cn, Z.Zhu, zhuzeyu@pku.edu.cn.

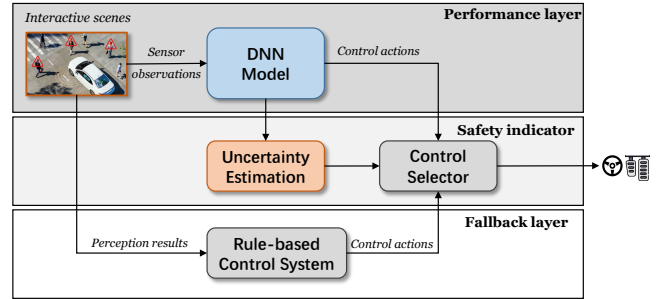


Fig. 1. We combine learning-based control models (performance layer) and a rule-based control system (fallback layer). A safety indicator is developed upon the uncertainty estimation of networks. The final control defaults back to a rule-based system for safety when the models are unreliable.

end-to-end methods do not typically provide the same level of transparency as in traditional modular systems, resulting in poor reliability and trustworthiness. Second, literature works have focused mainly on improving task performance, while quantification of uncertainties in either data (e.g., due to sensor noise) or model (e.g., due to distributional mismatch) has less been studied. Third, when a deep model is confronted with unfamiliar scenarios, such as traveling in a new town it hasn’t experienced in training, its performance usually drops dramatically. Countermeasures need to be taken by identifying the situations when the model is uncertain.

Uncertainty quantification is of significant importance for safety-critical applications like autonomous driving [14]. At intersection navigation, when encountering a pedestrian, a human driver can choose to yield or pass through longitudinal control, and meanwhile steer through lateral control. Such behavior can be modeled implicitly through imitation learning of human driving data. However, lateral and longitudinal controls have uncertainties of different nature (refer to Fig. 6). For example, scene features such as lane markings and obstacles are of different importance for lateral and longitudinal control tasks. In addition, the lateral and longitudinal controls are manipulated by human drivers’ feet and hands, respectively, and show different control vibration tolerances in expert demonstrations. There are literature works on DIL-based autonomous control policies that addressed intersection scenarios [4], [5], [15] and uncertainty-aware imitation learning methods have also been studied [16]–[20]. However, none of these works address the heterogeneity uncertainties in lateral and longitudinal controls, which cannot be ignored when learning from human driving data.

Demonstrated in Fig. 1, this work proposes an uncertainty-aware deep imitation learning method based on multi-task conditional imitation learning (MTCIL) [8]. We further com-

bine the deep model with a rule-based control system to enhance its safety and reliability under uncertain challenging interactive scenarios. Our contributions are as follows:

- A heterogeneity uncertainty quantification method is proposed for autonomous navigation at crowded intersections. Uncertainties at both the data and model levels are quantified in the lateral and longitudinal control tasks, and are learned from human driving data within the MTCIL framework.
- An online deployment strategy is proposed to enhance driving performance in challenging situations. By utilizing the quantified uncertainty as a safety indicator for control switch, the data-driven approach is combined with a traditional rule-based control system to achieve both high driving quality and safety.
- Extensive experiments are conducted demonstrating the superior performance of the proposed method. CARLA simulator is used for closed-loop test. Models are learned from human driving data collected in the simulator and extensively evaluated under various conditions.

## II. RELATED WORKS

### A. Deep Imitation Learning for Autonomous Driving Control

*Direct perception* methods [15], [21] leverage neural networks to extract intermediate representations, based on which control actions are generated. The choices of representations can be distances to the preceding vehicle and centerline [15]. One limitation is that system expertise is needed to design these representations, which is case-by-case and sub-optimal.

*End-to-end* methods [4], [5], [8], [22] directly train neural networks that map raw observations to control actions. One pioneering work is NVIDIA’s PilotNet [22] but it only adapts to simple lane following. To learn different driving skills, conditional imitation learning (CIL) [4] utilizes multi-branch network structures where the output is conditioned on high-level behavior commands. Subsequent methods reduce the neural networks’ perception burden through privileged supervision, such as road maps [6] and BEV images [7].

However, these methods focus on improving task performance and lack the ability to quantify uncertainty associated with either data or models. Directly deploying these models could be dangerous especially in safety-critical self-driving.

### B. Deep Learning Uncertainty Estimation

Uncertainty estimation in deep learning is significant [23], [24]. There are usually two kinds of deep learning uncertainty, i.e., aleatoric/data uncertainty and epistemic/model uncertainty [25]. The former one captures the noise inherent in data, which cannot be reduced even with unlimited data. The latter one accounts for uncertainty in model, which arises from data sparsity and distributional mismatch. Model uncertainty can be reduced given more training data.

Data uncertainty is usually modeled by placing a distribution over the model output and can be learned via heteroscedastic regression [26], [27]. Model uncertainty can be estimated via Bayesian [28] and non-Bayesian methods. Estimating the posterior of network weights is non-trivial

and approximations exist [29]–[31]. Dropout variational inference is a practical approach that can handle large complex models [30], [32]. There are also non-Bayesian methods such as ensembles [33], [34]. These methods are common in that they apply probabilistic reasoning on the network weights.

### C. Uncertainty-aware Autonomous Driving

In autonomous driving, uncertainty can arise from stochastic varying traffic environment, multi-modal driving behavior, partial observation and sensor noise, etc. Some methods [16]–[20] have been proposed to consider the deep learning uncertainty in autonomous driving. [16] proposed to deploy a visual-based driving policy from a stochastic and uncertainty-aware perspective. [18] quantified uncertainty with statistical guarantees based on Bayesian Neural Networks (BNN). However, their model deals with images of few pixels and simple stationary environments. [20] utilized epistemic uncertainty to detect and recover from distribution shifts.

However, none of these works address the heterogeneity uncertainties in lateral and longitudinal controls, which cannot be ignored when learning from human driving data.

## III. METHODOLOGY

### A. Problem Definition

To navigate through crowded urban intersections, an autonomous agent needs to interact with pedestrians on crosswalks and vehicles in its vicinity. Given real-time observations like front-view images, it adjusts the lateral and longitudinal control actions to complete a sequence of driving commands successfully, safely and efficiently.

This work learns from a demonstration dataset  $\mathcal{D}$  consisted of driving trajectories  $\{\mathcal{T}_n\}_{n=1}^N$ . One trajectory  $\mathcal{T}_n$  is a temporal series of tuples  $\{(o_n^t, c_n^t, a_n^t)\}_{t=1}^T$ , where  $o_n^t$ ,  $c_n^t$  and  $a_n^t$  represent observation, command and action, respectively. The observation at each time step includes the front-view RGB image  $I_n^t$  and ego speed  $v_n^t$ . The actions include the lateral steering angle  $a_{lat}$  and longitudinal acceleration  $a_{lon}$ . The commands are categorized into lateral and longitudinal ones accordingly, i.e.,  $c_{lat} \in \{\text{follow lane, turn left, turn right, go straight}\}$  and  $c_{lon} \in \{\text{decelerate, maintain, accelerate}\}$ . The goal is to learn a deep neural network control policy  $\pi_\theta$  parameterized by  $\theta$ , which maps observations and commands into actions  $a_{NN} = \pi_\theta(o, c)$ . A typical  $L_2$  imitation loss as follows is optimized to obtain the optimal parameters  $\theta^*$ .

$$\theta^* = \arg \min_{\theta} \sum_i \mathcal{L}(\pi_\theta(o_i, c_i), a_i) \quad (1)$$

To tackle the different natures of uncertainties in lateral and longitudinal controls, the multi-task conditional imitation learning (MTCIL) framework [8] is exploited, as shown in Fig. 2(a). To quantify the uncertainty dependent on input data, i.e. data uncertainty, the deep networks output a mean  $\mu$  and a variance  $\sigma^2$  of Gaussian distribution for both lateral and longitudinal control actions. These variance values  $\sigma^2$  reflecting data uncertainty are also used to weight tasks of lateral and longitudinal controls in multi-task learning. On the other hand, deep ensemble is used to quantify the

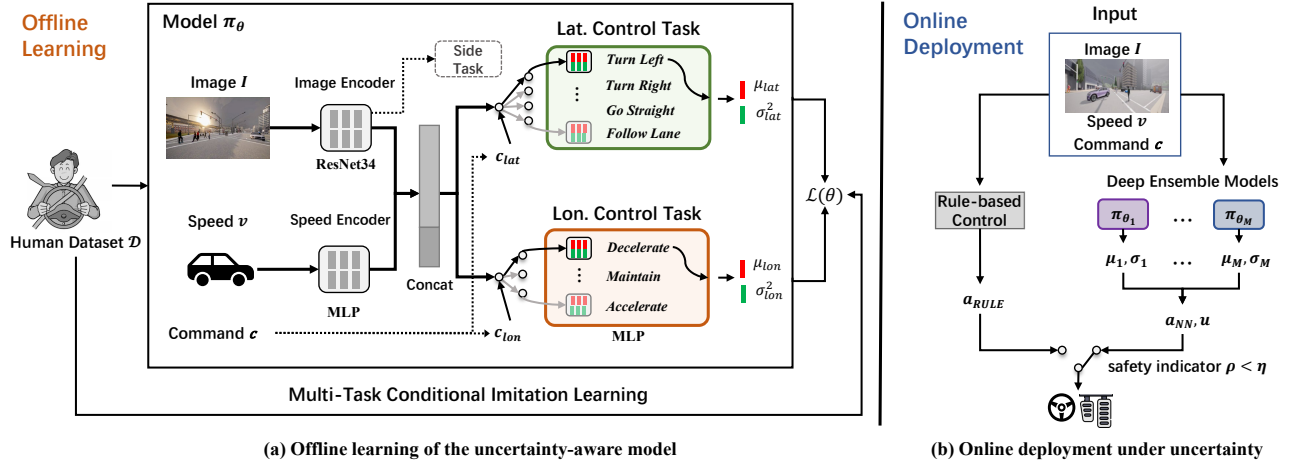


Fig. 2. (a) **Offline learning of the uncertainty-aware model** (cf. Section III-B). The density network outputs action means and variances that estimate data uncertainty. The model is trained via maximum likelihood within the MTCIL framework. (b) **Online deployment under uncertainty** (cf. Section III-D). When the ensemble yields uncertain (potentially unsafe) predictions at test time, the control is switched to a rule-based control system.

uncertainties from both input data and deep control models as shown in Fig. 2(b). In online deployment, the total uncertainty is used to identify situations when the deep policy model is uncertain, and subsequently the output of a rule-based control system is exploited to improve safety.

### B. Learning Data Uncertainty with Heteroscedastic Loss

Assume the demonstration dataset  $\mathcal{D} = \{(o_i, c_i, a_i)\}_{i=1}^{|\mathcal{D}|}$  is sampled from a true joint distribution  $\mathbb{P}_{tr}$ . The generation of data samples can be formulated as follows [23]:

$$a = \pi_{\theta}(o, c) + \epsilon, \quad (o, c) \sim \mathbb{P}_{tr}(o, c) \quad (2)$$

where  $\pi_{\theta}$  is a driving policy mapping input  $(o, c)$  to action  $a$  and  $\epsilon$  is the additive noise. A Gaussian noise is usually assumed for regression tasks, i.e.,  $a \sim \mathcal{N}(\pi_{\theta}(o, c), \sigma^2)$ . Since the noise  $\epsilon$  may vary across different observations and commands, this work adopts the heteroscedastic noise [32], where a learnable function  $\sigma(o, c)$  varying with input is used for variance estimation. To this end, we transform the original deterministic neural network into a probabilistic density network that can yield uncertainty estimates.

Our learning framework is shown in Fig. 2(a). The input front-view image  $I$  and ego speed value  $v$  are encoded using a ResNet34 [35] and a multi-layer perceptron (MLP) respectively. Features are concatenated and passed to two branches to perform lateral and longitudinal control tasks. Each task is accomplished by a conditional control module that contains multiple MLPs, one of which is selected each time according to the lateral and longitudinal command  $c = (c_{lat}, c_{lon})$ . To learn input-dependent data uncertainty, we use the probabilistic density network, where each control MLP outputs a mean action  $\mu$  and a variance  $\sigma^2$ :

$$\pi_{\theta}(o, c) = [\mu_{lat}, \sigma_{lat}^2, \mu_{lon}, \sigma_{lon}^2] \quad (3)$$

Given the dataset  $\mathcal{D} = \{(o_i, c_i, a_i)\}_{i=1}^{|\mathcal{D}|}$ , where  $a_i = (a_{i,lat}, a_{i,lon})$  is a tuple containing the steer for lateral control and acceleration for longitudinal control. Our imitation learning objective is to maximize the total likelihood:

$$\begin{aligned} \theta^* &= \arg \min_{\theta} \mathcal{L}(\theta) \\ \mathcal{L}(\theta) &= \sum_i -\log p(a_i | \pi_{\theta}(o_i, c_i)) \\ &= \sum_i \frac{1}{2} \sigma_{lat,i}^{-2} \|a_{lat,i} - \mu_{lat,i}\|^2 + \log \sigma_{lat,i} \\ &\quad + \frac{1}{2} \sigma_{lon,i}^{-2} \|a_{lon,i} - \mu_{lon,i}\|^2 + \log \sigma_{lon,i} \end{aligned} \quad (4)$$

Intuitively, the loss applies weight to different samples, relaxing the MSE error of those with higher data uncertainty, while the log terms serve as regularization.

### C. Deep Ensembles based Total Uncertainty Estimation

Deep ensembles [34], [36] is used to estimate the total uncertainty from both data and model. Given an ensemble of density networks with parameters  $\theta_1, \theta_2, \dots, \theta_M$ , it is expected that the ensembles predict consistent results for the well-learned (*confident*) scenes, while inconsistent predictions confronting rare, unexperienced or confusing (*uncertain*) situations. Model uncertainty can be assessed via measures of the disagreement between models (e.g., variance). This work exploits the law of total variance [37]:

$$\mathbb{V}[a|s] = \underbrace{\mathbb{V}_{p(\theta|\mathcal{D})} [\mathbb{E}_{p(a|\pi_{\theta}(s))} [a]]]}_{\text{model uncertainty}} + \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})} [\mathbb{V}_{p(a|\pi_{\theta}(s))} [a]]]}_{\text{expected data uncertainty}}$$

where  $s$  denotes  $(o, c)$  and the total variance of  $\mathbb{V}[a|s]$  is decomposed into the variance of expectations (*model uncertainty*) and the expectation of variances (*data uncertainty*).

This work constructs ensembles through the non-Bayesian *Bootstrap* approach [38]. Compared with Bayesian methods, the advantage is that the burden of selecting suitable model prior and nontrivial posterior variational approximation [39], [40] is removed. The entire dataset  $\mathcal{D}$  is randomly divided into subsets, i.e.,  $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \dots \cup \mathcal{D}_M$ , on which models are trained independently using the loss (Eqn. 4) to obtain the parameters  $\theta_1, \theta_2, \dots, \theta_M$ . Given a test input  $(o, c)$ , each ensemble  $\pi_{\theta_m}$  predicts mean actions for lateral

and longitudinal controls and their variances, i.e.,  $\mu_m = (\mu_{lat}, \mu_{lon})_m$  and  $\sigma_m^2 = (\sigma_{lat}^2, \sigma_{lon}^2)_m$ . The final ensemble action  $a_{NN}$  and total uncertainty  $u$  are estimated as follows:

$$a_{NN} = \frac{1}{M} \sum_m \mu_m \quad (5)$$

$$u = \underbrace{\frac{1}{M} \sum_m \mu_m^2 - \left( \frac{1}{M} \sum_m \mu_m \right)^2}_{\text{model uncertainty}} + \underbrace{\frac{1}{M} \sum_m \sigma_m^2}_{\text{data uncertainty}} \quad (6)$$

#### D. Policy Deployment under Uncertainty

During online policy deployment, the quantification of total uncertainty  $u$  is used to identify the situations when the model is uncertain, and let a traditional modular rule-based control system to take over. In order to avoid frequent switching between control models, a safety indicator  $\rho$  is developed by accumulating the discounted total uncertainties within a time window  $(t - T, t]$ , and the rule-based control action is exploited when  $\rho \geq \eta$ .

$$\rho_t = \sum_{k=0}^{T-1} \gamma^k u_{t-k} \quad (7)$$

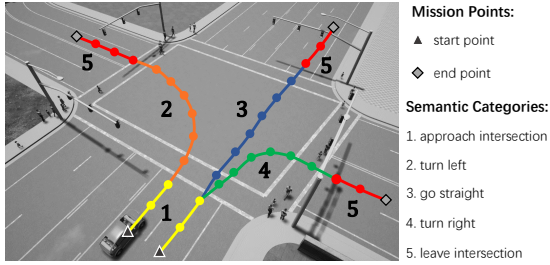


Fig. 3. Driving routes divided into sections of different semantic categories.

However, during driving the level of uncertainty may vary with region and driving behavior, and there are different variations in uncertainty for lateral and longitudinal controls. It is reflected in uncertainty maps in Fig. 6, where uncertainties in lateral and longitudinal controls during 50 closed-loop test episodes are recorded, and average uncertainties are estimated for each grid of the maps. It can be seen that the lateral uncertainty before entering the intersection is low, while that within the intersection is high. To this end, the driving routes are divided into sections of five semantic categories as shown in Fig. 3, namely *approach intersection*, *leave intersection*, *turn left*, *turn right* or *go straight through intersection*. Using the train dataset, the lateral and longitudinal uncertainty distributions on the road sections for each semantic category are calculated and shown in Fig. 4(a-b), where the horizontal and vertical axes denote uncertainty (log scale) and cumulative probability, respectively. Given a certain cumulative probability  $\lambda$  as shown in Fig. 4(c), the corresponding uncertainty value of each semantic category  $k$  is assigned to  $\eta_{lon,k}$  or  $\eta_{lat,k}$ , and sets of thresholds  $\eta_{lat} = \{\eta_{lat,k}\}_{k=1}^5$  and  $\eta_{lon} = \{\eta_{lon,k}\}_{k=1}^5$  are obtained for switch of lateral and longitudinal controls between the rule-based and the deep learning models on different road

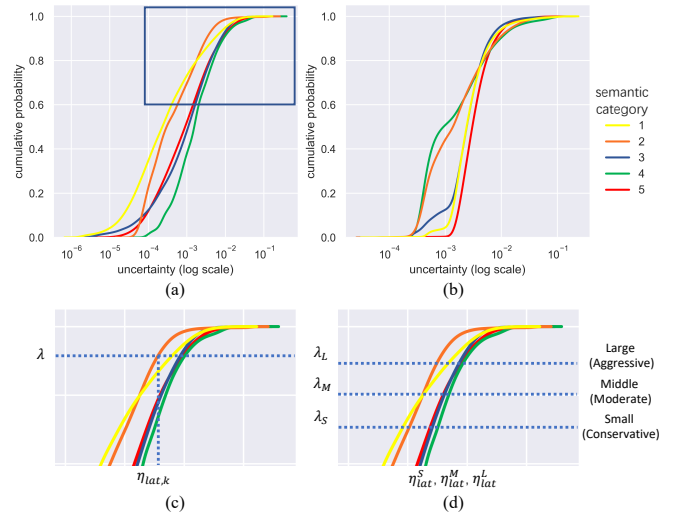


Fig. 4. (a-b) Cumulative probability of the total lateral (a) and longitudinal (b) uncertainty distributions belonging to different semantic categories. (c) determination of the threshold of semantic category  $k$  (d) multiple sets of thresholds to develop different driving styles

sections. On the other hand,  $\lambda$  can be varied to develop different driving behaviors. In this work,  $\lambda$  is defined at three different levels (c.f. Fig. 4(d)): Large, Middle, Small, yielding the control of aggressive, moderate and conservative styles. The deployment algorithm is outlined below.

#### Algorithm 1 Policy Deployment under Uncertainty

**Input:** models  $\{\pi_{\theta_m}\}_{m=1}^M$ , rule-based policy  $\pi^R$ , thresholds

$\eta_{lat}, \eta_{lon}$

**Output:** Control actions  $\{a_t\}$ , where  $a_t = (a_{lat,t}, a_{lon,t})$

```

1: while  $\neg$  Episode end do
2:   Get  $o_t, c_t$ 
3:   for  $m = 1$  to  $M$  do
4:      $\pi_{\theta_m}(o_t, c_t) = (\mu_{lat}, \sigma_{lat}^2, \mu_{lon}, \sigma_{lon}^2)_m$ 
5:   end for
6:   Compute  $u_{lat,t}, u_{lon,t}$  on Eqn. 6
7:   Compute  $a_{lat,t}, a_{lon,t}$  on Eqn. 5
8:   Compute  $\rho_{lat,t}, \rho_{lon,t}$  on Eqn. 7
9:   Get semantic category  $k$  of the current road section
10:  if  $\rho_{lat,t} \geq \eta_{lat,k}$  then
11:    Switch to rule policy,  $a_{lat,t} = \pi_{lat}^R(o_t, c_t)$ 
12:  end if
13:  if  $\rho_{lon,t} \geq \eta_{lon,k}$  then
14:    Switch to rule policy,  $a_{lon,t} = \pi_{lon}^R(o_t, c_t)$ 
15:  end if
16:  Control vehicle with  $a_{lat,t}, a_{lon,t}$ 
17:   $t = t + 1$ 
18: end while

```

## IV. EXPERIMENTAL RESULTS

### A. Benchmark and Dataset

**Benchmark:** This work uses the IntersectNav Benchmark [8], which supports closed-loop test on the realistic 3D simulator CARLA. Six intersections and 41 navigation routes navigating are available. To complete three kinds of tasks, i.e., turn left/right and go straight, the agent needs to interact with pedestrians and other vehicles. In each episode, 20-30

pedestrians and 6-15 environmental vehicles are generated in the intersection and controlled by CARLA built-in AI controllers. Challenging situations such as unprotected turn could happen. There are four types of episode outcomes: collision, lane invasion, timeout and success (complete the task without former three failures). Evaluations from the aspects of task completion and driving performance are conducted using the metrics defined in [8]. For the former, success/collision/timeout/lane invasion rates are estimated. For the latter, intense actions (the average number of over-controlled actions, the larger the value the more uncomfortable) and total steps (the average total steps per episode, the smaller the value the more efficient) are estimated.

*Dataset:* The demonstration dataset from [8] is used, which contains about 1.3K trajectories and 40 hours driving data from human drivers. Data from four intersections are used for training and validation while data from the other two are reserved for test. The training dataset includes six weather conditions such as *ClearNoon* and *CloudySunset*. During evaluation, eight new weather conditions such as *HardRainNoon* and *ClearNight* are also tested.

### B. Compared Methods and Implementation Details

Methods of different categories are chosen for comparison. *Rule-based baseline:*

- CARLA autopilot [41]: a modular rule-based control system, including collision check, motion planner and a PID controller. Given other agents' locations, potential collision hazards are identified to determine the vehicle target speed upon rules. Then the vehicle's movement is determined by PID to reach the target waypoints.

*Learning-based baselines:*

- CILRS [5]: the original method unaware of uncertainty.
- MTCIL [8]: aware of homoscedastic (input independent) data uncertainty only.
- MTCIL\_En, a deep ensemble [34] variant of MTCIL [8]
- CILRS\_Drop, a variant of our method that uses test-time dropout [30], [32] instead of deep ensemble [34].

*The proposed:*

- UACIL-D: our uncertainty-aware deep imitation learning model.
- UACIL-U: our uncertainty-aware deployment method that combines learning-based models with a rule-based system (CARLA autopilot) as fallback.

We train all models using Adam optimizer with an initial learning rate  $2e-4$ , which is divided by 10 if validation loss stops decreasing for more than 5 training epochs. We set the test-time dropout rate of CILRS\_Drop to 0.5. For ensemble models, we set the number of models  $M = 5$  following [34]. In Eqn. 7, we set  $T = 10$  and  $\gamma = 0.95$ . Three cumulative probabilities  $\lambda = 0.95, 0.92, 0.9$  are chosen to determine the uncertainty thresholds.

### C. Task Performance

The task completion results of evaluated methods are reported in Tab. I. The driving performance results of methods

TABLE I  
CLOSED-LOOP TEST TASK COMPLETION OF EVALUATED METHODS

Method	Succ. Rt.	Time. Rt.	Lane. Rt.	Colli. Rt.
<i>Train scene</i>				
Rule-based	95.0 ± 1.4	4.7 ± 0.8	0.0 ± 0.0	<b>0.3 ± 0.2</b>
CILRS	65.3 ± 1.8	9.4 ± 0.6	7.2 ± 1.6	18.1 ± 1.5
MTCIL	84.7 ± 0.8	3.7 ± 1.5	5.9 ± 0.4	5.7 ± 1.6
CILRS_Drop	85.2 ± 3.1	2.4 ± 1.5	3.0 ± 1.4	9.4 ± 2.6
MTCIL_En	94.9 ± 2.2	3.8 ± 2.0	0.0 ± 0.0	1.1 ± 0.6
UACIL-D	96.3 ± 1.6	2.6 ± 1.1	0.4 ± 0.3	0.7 ± 0.6
UACIL-U	<b>97.9 ± 1.7</b>	<b>1.7 ± 0.4</b>	<b>0.0 ± 0.0</b>	0.4 ± 0.2
<i>New scene</i>				
Rule-based	94.6 ± 2.6	5.0 ± 2.7	0.0 ± 0.0	<b>0.4 ± 0.3</b>
CILRS	52.8 ± 3.4	9.8 ± 4.2	9.7 ± 2.6	27.7 ± 3.2
MTCIL	80.3 ± 1.6	4.1 ± 1.6	8.5 ± 1.2	7.1 ± 0.6
CILRS_Drop	86.7 ± 4.2	3.3 ± 1.5	3.2 ± 1.9	6.8 ± 2.3
MTCIL_En	92.2 ± 2.0	3.8 ± 1.8	1.5 ± 1.0	2.5 ± 1.0
UACIL-D	94.0 ± 2.2	4.3 ± 2.0	0.0 ± 0.0	1.7 ± 1.0
UACIL-U	<b>96.1 ± 2.6</b>	<b>3.1 ± 2.1</b>	<b>0.0 ± 0.0</b>	0.8 ± 0.6

<sup>1</sup> Mean and standard deviation over five random seeds.

<sup>2</sup> **Succ. Rt.:** Success Rate, **Time. Rt.:** Timeout Rate, **Lane. Rt.:** Lane Invasion Rate, **Colli. Rt.:** Collision Rate.

<sup>3</sup> Our proposed methods are marked in grey. UACIL-D/U: Direct deployment/Uncertainty-aware deployment.

TABLE II  
CLOSED-LOOP TEST DRIVING PERFORMANCE OF EVALUATED METHODS

Method	Intense Actions	Total Steps
<i>Train scene</i>		
	#, ↓	#, ↓
Rule-based	241.962 ± 3.092	554.835 ± 11.229
MTCIL_En	56.810 ± 2.392	514.088 ± 15.268
UACIL-D	<b>55.717 ± 6.033</b>	<b>510.494 ± 12.113</b>
UACIL-U	64.501 ± 4.503	519.549 ± 13.971
<i>New scene</i>		
Rule-based	260.508 ± 13.571	597.779 ± 17.806
MTCIL_En	51.495 ± 0.834	558.013 ± 34.536
UACIL-D	<b>43.347 ± 1.763</b>	<b>536.465 ± 13.825</b>
UACIL-U	51.771 ± 1.690	546.126 ± 6.260

<sup>1</sup> Mean and standard deviation over five random seeds.

<sup>2</sup> **Intense actions:** Average times of the agents' steer  $|a_{lat}| > 0.4$  or acceleration  $|a_{lon}| > 0.9$ ; **Total steps:** Average time steps of closed-loop test episodes.

whose success rates are higher than 90% are further reported in Tab. II. In brief, the learning-based methods show superior driving performance than the rule-based method, where our UACIL-D is the best in terms of intense actions (comfort) and total steps (efficiency). The rule-based method performs well in task completion. By combining the learning-based with the rule-based, our UACIL-U has the best overall performance. The results are discussed in more detail below.

*Rule-based baseline:* The rule-based baseline has the lowest collision rates (best safety) since its parameters are tuned to be conservative and guarantee safety first (i.e., always leave large safe distances and yield to pedestrians). However, its driving performance is worse than learning-based methods (c.f. Tab. II), i.e., more intense actions and less efficiency.

*Learning-based baselines:* Among learning-based methods, ensemble methods MTCIL\_En and UACIL-D outperform other methods by far in success rates. One explanation is that ensemble models benefit from training on different subsets of data and gain diverse experiences. This avoids overfitting to most common samples. Besides, the ensemble models' aggregated action predictions are more robust to disturbances in observations, resulting in better

control stability. By treating each input sample differently in the learning process, UACIL-D achieves better driving performance compared to MTCIL\_En.

*The proposed:* UACIL-U outperforms UACIL-D by a reduced collision rate and an increased success rate. It also outperforms the rule-based system in driving performance (fewer intense actions and higher efficiency), which demonstrates the benefit of combining learned models with the rule-based system through uncertainty-aware deployment.

#### D. Uncertainty Quantification Results

Fig. 5 plots the lateral data and model uncertainty (cf. Eqn. 6) distribution on different datasets. In our case, the model uncertainty on train set can be diminished to a low level via enough training while data uncertainty can't. The reason lies in that data uncertainty is the inherent property of human control actions. Another finding is that model uncertainty significantly increases at the test set which is collected from intersections unexperienced during training.

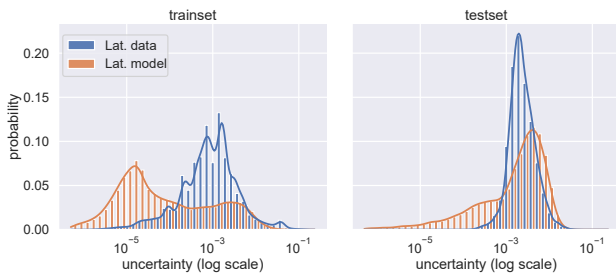


Fig. 5. Estimated data and model uncertainty on train and test datasets.

We further analyze the uncertainty quantification results of closed-loop test under various conditions (i.e., scenes, weather). The average uncertainty of 400 closed-loop test episodes in is listed in Tab. III. The new scene and new weather is the most challenging situation and it has the highest values of most types of uncertainties. Besides, lateral and longitudinal total uncertainties have different sensitivity to changes in scenes and weathers.

TABLE III

AVERAGE UNCERTAINTY IN TEST UNDER DIFFERENT CONDITIONS

Scene	Weather	Lat. Uc. $\times 10^{-3}$			Lon. Uc. $\times 10^{-3}$		
		Total	Model	Data	Total	Model	Data
train	train	2.219	0.748	1.471	3.268	1.891	1.377
new	train	4.622	<b>1.187</b>	3.435	4.237	<b>2.536</b>	1.701
train	new	3.263	1.047	2.216	3.979	2.182	1.797
new	new	<b>4.797</b>	1.080	<b>3.717</b>	<b>4.320</b>	2.219	<b>2.101</b>

<sup>1</sup> **Bold** numbers are the highest and underlined ones are the second highest.

#### E. Task-dependent Uncertainty Analysis

Our results find that uncertainties of the lateral and longitudinal control tasks show different properties (i.e., heterogeneity) and vary across spatial areas. The uncertainty distribution of one of the left turn tasks is shown in Fig. 6. The results are from 50 closed-loop test episodes. In the left turn task, the lateral total uncertainty is smaller than that of longitudinal control. This is consistent with the property of human driving data, i.e., the steering angle during left

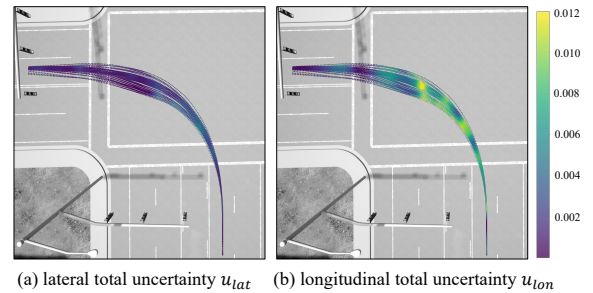


Fig. 6. Total uncertainty distribution visualization (top-down view).

turn is relatively stable while there are more fluctuations in acceleration. In Fig. 6(a), the lateral total uncertainty is relatively lower at straight lanes and higher in the intersection area, indicating more uncertainty during the course of turning compared to lane keeping. In Fig. 6(b), the longitudinal total uncertainty is larger in the intersection, especially at the waiting area in front of the crosswalks. This indicates higher uncertainty when the agent interacts with pedestrians, e.g., unsure about whether to yield to pedestrians or not.

A case study is presented in Fig. 7. The first and second rows show frames from different simulation episodes with low and high total uncertainty, respectively. We find low uncertainty usually corresponds with scenarios when the model is pretty sure while high uncertainty tends to correspond with uncertain rare scenarios. For example, the high uncertainty frames of three cases correspond to unprotected left turn, opposite vehicle running red light and right turn merge, respectively. All these situations are rare in the training data, showing that our method has the potential to identify situations different than training. Besides, the lateral and longitudinal total uncertainties have different quantities and vary across navigation tasks, demonstrating their heterogeneity.

#### F. Uncertainty-aware Policy Deployment Results

This section analyzes the performance gap between direct and uncertainty-aware policy deployment under more challenging conditions. Three sets of uncertainty thresholds  $\{\eta_{lat}, \eta_{lon}\}$  are chosen for comparison in Algorithm. 1 that correspond to **Small**, **Middle** and **Large** values. The higher the uncertainty threshold, the more aggressive the final deployment policy (i.e., more trust in the network's predictions). Vision-based end-to-end driving control models

TABLE IV

TAKE OVER RATIOS OF DIFFERENT THRESHOLDS AND CONDITIONS

Method	Threshold	Condition			
		TS&TW	NS&TW	TS&NW	NS&NW
UACIL-U(S)	Small	0.1751	0.3296	0.2189	0.3753
UACIL-U(M)	Medium	0.0806	0.1311	0.0878	0.1468
UACIL-U(L)	Large	0.0409	0.0648	0.0426	0.0695

<sup>1</sup> TS/NS: Train/New Scene, TW/NW: Train/New Weather.

are sensitive to changes in images (e.g., different illumination conditions). Demonstrated in Fig. 8, the direct deployment method UACIL-D's performance endures dramatic degradation (a decrease in success rate up to 10%) when encountering new scenes or weathers. Although uncertainty-aware deployment methods UACIL-U have similar success

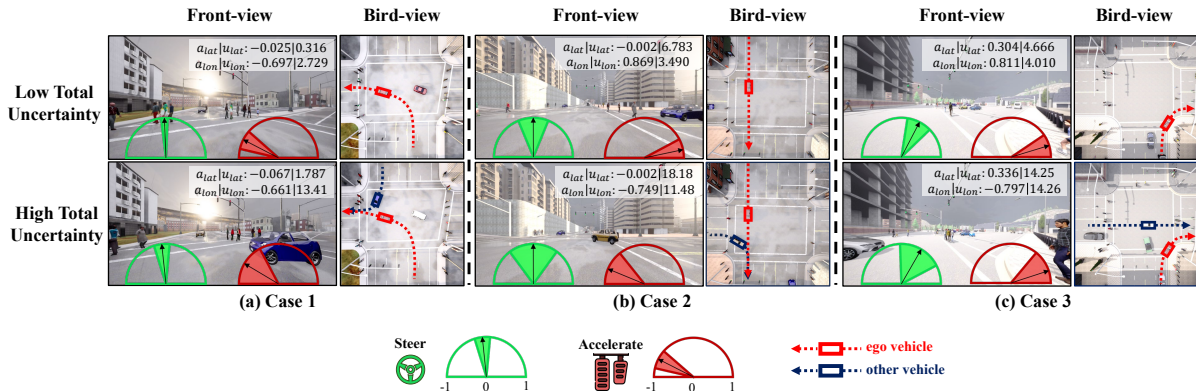


Fig. 7. Case study of simulation frames. The first row: low total uncertainty frames that correspond with **confident** scenarios, which the model is well-trained to handle. The second row: high total uncertainty frames that tend to correspond with **uncertain** and **rare** scenarios, which the model has hardly experienced during training. The arrow points to the prediction while the sector represents the uncertainty. Case 1 is from a train scene while Case 2&3 are from new scenes. The order of magnitude of uncertainty values is  $10^{-3}$ . **Note that the bird-view image is only for reference.**

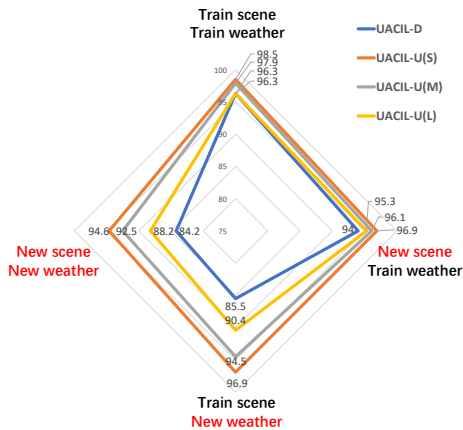


Fig. 8. Success rates of different deployment settings. UACIL-D/U: Direct/Uncertainty-aware deployment; S,M,L correspond to three sets of uncertainty thresholds: Small, Middle, Large.

rates as UACIL-D under train scene and train weather, they have better performance under new conditions. This demonstrates the necessity of considering the uncertainty during deployment and switching to the fallback rule-based system for reliability when the models are uncertain.

Nevertheless, let the rule-based method take over too many times could be inefficient. The take over ratios (proportion of frames when rule-based policy takes over control in an episode) are averaged over multiple simulation episodes and reported in Tab. IV. Although UACIL-U(S) achieves the highest success rate in Fig. 8, it has the largest take over ratio up to 37.5%. In our case, we find that UACIL-U(M) has the acceptable performance and take over ratios (less than 15% under new scenes and new weathers).

## V. CONCLUSION, INSIGHTS AND FUTURE WORKS

This work proposes an uncertainty-aware deep imitation learning and deployment method for autonomous navigation through crowded intersections. A heterogeneity uncertainty quantification method is developed, and an uncertainty-aware deployment strategy is proposed to switch control to a fallback system when deep models are uncertain. Extensive experiments are conducted, and the proposed method is

demonstrated of superior performance on both task completion and driving performance. Besides, we have the following insights and future works.

*Learning-based method v.s. rule-based control system.* Traditional rule-based control systems generate driving actions using hand-crafted rules and fine-tuned parameters. In dynamic intersection scenarios, their parameters are tuned to ensure safety first, leading to conservative driving behavior while sacrificing efficiency. In contrast, data-driven methods can represent the highly non-linear procedure of a human driver's decision-making. The deep model learned from human demonstrations have the potential to mimic human drivers' behavior, whereas it is not as transparent and safe as hand-crafted rules due to its black-box feature. Using uncertainty quantification as proposed in this work, an autonomous driving system can leverage deep models to achieve comfort and efficiency in most cases, while leaving safety to a rule-based system in corner cases.

*Is the quantified uncertainty reliable and trustable?* In most uncertainty quantification techniques, since the uncertainty is estimated based on the output of models, the reliability of the uncertainty is an open question, especially when facing new test data that has distributional shift from training set. In future work, uncertainty calibration methods will be leveraged to mitigate the problem.

*Is modeling the deep uncertainty enough?* The modeled uncertainty in this work comes from deep learning, i.e., from data and model, respectively. However, apart from the ego-perspective uncertainty, uncertainty arising from surrounding traffic systems should be modeled. For example, the pedestrians' and environment vehicles' behaviors are inherently stochastic, dynamic and uncertain. Planning under both kinds of uncertainty in a unified framework is left for future work.

## REFERENCES

- [1] S. Kuutti *et al.*, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2020.

- [2] Z. Zhu and H. Zhao, "A survey of deep rl and il for autonomous driving policy learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14043–14065, 2021.
- [3] A. O. Ly and M. Akhloufi, "Learning to drive by imitation: An overview of deep behavior cloning methods," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 195–209, 2020.
- [4] F. Codevilla *et al.*, "End-to-end driving via conditional imitation learning," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 4693–4700.
- [5] F. Codevilla *et al.*, "Exploring the limitations of behavior cloning for autonomous driving," in *IEEE International Conference on Computer Vision*, 2019, pp. 9329–9338.
- [6] D. Chen *et al.*, "Learning by cheating," in *Conference on Robot Learning*, 2020, pp. 66–75.
- [7] Z. Zhang *et al.*, "End-to-end urban driving by imitating a reinforcement learning coach," in *IEEE International Conference on Computer Vision*, 2021, pp. 15222–15232.
- [8] Z. Zhu and H. Zhao, "Learning autonomous control policy for intersection navigation with pedestrian interaction," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 5, pp. 3270–3284, 2023.
- [9] V. Sezer *et al.*, "Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent," in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 3578–3585.
- [10] D. Isele *et al.*, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2034–2039.
- [11] L. Wei *et al.*, "Autonomous driving strategies at intersections: Scenarios, state-of-the-art, and future outlooks," in *IEEE International Intelligent Transportation Systems Conference*, 2021, pp. 44–51.
- [12] D. González *et al.*, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2015.
- [13] B. Paden *et al.*, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [14] R. McAllister *et al.*, "Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning," in *International Joint Conference on Artificial Intelligence*, 2017.
- [15] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *Conference on Robot Learning*, 2018, pp. 237–252.
- [16] L. Tai *et al.*, "Visual-based autonomous driving deployment from a stochastic and uncertainty-aware perspective," in *IEEE International Conference on Intelligent Robots and Systems*, 2019, pp. 2622–2628.
- [17] Y. Cui *et al.*, "Uncertainty-aware data aggregation for deep imitation learning," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 761–767.
- [18] R. Micheltore *et al.*, "Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 7344–7350.
- [19] F. Nozarian, C. Müller, and P. Slusallek, "Uncertainty quantification and calibration of imitation learning policy in autonomous driving," in *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, 2020, pp. 146–162.
- [20] A. Filos *et al.*, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?" In *International Conference on Machine Learning*, 2020, pp. 3145–3153.
- [21] C. Chen *et al.*, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.
- [22] M. Bojarski *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.
- [23] A. Malinin, "Uncertainty estimation in deep learning with application to spoken language assessment," Ph.D. dissertation, University of Cambridge, 2019.
- [24] M. Abdar *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.
- [25] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, University of Cambridge, 2016.
- [26] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *IEEE International Conference on Neural Networks*, vol. 1, 1994, pp. 55–60.
- [27] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic gaussian process regression," in *International Conference on Machine Learning*, 2005, pp. 489–496.
- [28] D. J. MacKay, "A practical bayesian framework for back-propagation networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992.
- [29] A. Graves, "Practical variational inference for neural networks," *Advances in Neural Information Processing Systems*, vol. 24, 2011.
- [30] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.
- [31] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016, pp. 1050–1059.
- [32] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" In *Advances in Neural Information Processing Systems*, 2017, pp. 5574–5584.
- [33] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, vol. 1857, Springer, 2000, pp. 1–15.
- [34] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems 30*, 2017.
- [35] K. He *et al.*, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [36] K. Chua *et al.*, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [37] S. Depeweg *et al.*, "Decomposition of uncertainty for active learning and reliable reinforcement learning in stochastic systems," *Stat*, vol. 1050, p. 11, 2017.
- [38] I. Osband *et al.*, "Deep exploration via bootstrapped dqn," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [39] C. Blundell *et al.*, "Weight uncertainty in neural network," in *International Conference on Machine Learning*, 2015, pp. 1613–1622.
- [40] J. M. Hernández-Lobato and R. Adams, "Probabilistic back-propagation for scalable learning of bayesian neural networks," in *International Conference on Machine Learning*, 2015, pp. 1861–1869.
- [41] A. Dosovitskiy *et al.*, "Carla: An open urban driving simulator," in *Conference on robot learning*, 2017, pp. 1–16.