2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Continual Driving Policy Optimization with Closed-Loop Individualized Curricula

Haoyi Niu[1†], Yizhou Xu[1†], Xingjian Jiang[1], Jianming Hu[1✉]

***Abstract*— The safety of autonomous vehicles (AV) has been a long-standing top concern, stemming from the absence of rare and safety-critical scenarios in the long-tail naturalistic driving distribution. To tackle this challenge, a surge of research in scenario-based autonomous driving has emerged, with a focus on generating high-risk driving scenarios and applying them to conduct safety-critical testing of AV models. However, limited work has been explored on the reuse of these extensive scenarios to iteratively improve AV models. Moreover, it remains intractable and challenging to filter through gigantic scenario libraries collected from other AV models with distinct behaviors, attempting to extract transferable information for current AV improvement. Therefore, we develop a continual driving policy optimization framework featuring Closed-Loop Individualized Curricula (CLIC), which we factorize into a set of standardized sub-modules for flexible implementation choices: *AV Evaluation*, *Scenario Selection*, and *AV Training*. CLIC frames AV Evaluation as a collision prediction task, where it estimates the chance of AV failures in these scenarios at each iteration. Subsequently, by re-sampling from historical scenarios based on these failure probabilities, CLIC tailors individualized curricula for downstream training, aligning them with the evaluated capability of AV. Accordingly, CLIC not only maximizes the utilization of the vast pre-collected scenario library for closed-loop driving policy optimization but also facilitates AV improvement by individualizing its training with more challenging cases out of those poorly organized scenarios. Experimental results clearly indicate that CLIC surpasses other curriculum-based training strategies, showing substantial improvement in managing risky scenarios, while still maintaining proficiency in handling simpler cases.**

I. Introduction

With remarkable advancements in deep learning (DL) and deep reinforcement learning (DRL), autonomous driving has gained substantial interest from academia, industry, and the public. However, the deployment of autonomous vehicles (AV) in the real world has been significantly impeded by safety concerns. The primary crux sources from the distribution of naturalistic driving data (NDD), which exhibits a long-tailed pattern [1], leading to a severe data imbalance characterized by a scarcity of safety-critical scenarios. Naturally, solely relying on NDD would require training and testing AV for billions of miles [2] to ensure safety. Thereby, a repertoire of studies has emphasized the need to generate safety-critical scenarios [3]–[12] to address the data imbalance issue. This has led to the emergence of an exciting avenue coined as scenario-based autonomous driving [13], [14] that harbors several advantages against autonomous driving under naturalistic situations. Logged scenarios offer substantial reproducibility, controllability and flexibility for re-organization, avoiding repetitive computations during replay and allowing re-sampling of valuable scenarios at will. These advantageous properties hold great promise for accelerating the testing phase [15]–[21] introduced with advanced re-sampling techniques over safety-critical scenarios.

However, limited research has explored the potential of utilizing the rich and diverse historical scenarios for closed-loop training of AV, rather than just for testing purposes. The main challenge is twofold: (1) *Scenario Transferability*: Pre-collected scenarios can vary significantly across different driving patterns, including those generated by overly timid and considerably aggressive AV models or human drivers. AV models trained directly on such scenarios may not always yield improvement due to the significant distribution inconsistency [22], [23]. (2) *Scenario Adaptation*: Unlike the fixed AV model used during the testing phase, closed-loop AV policy optimization inevitably involves dynamical improvement of the iterated AV model. Moreover, driving scenarios are widely acknowledged for their diversity in difficulty levels, so the AV model needs to be fed with carefully selected scenarios that align with their capabilities at each iteration [23]. This highlights the need to offer individualized curricula comprising more challenging scenarios and less boring ones, tailored to fulfill the requirements of efficient and effective training at current stage.

To tackle these challenges, we introduce a novel framework of **Continual Driving Policy Optimization with Closed-Loop Individualized Curricula (CLIC)**, which we divide into three standardized sub-modules for flexible implementation choices: (1) **AV Evaluation**: At each iteration, we begin by exposing the AV model to a subset of scenarios to assess the current AV capability. (2) **Scenario Selection**: Next, we aim to estimate whether the AV model collides with others in each scenario. To achieve this, we employ a discriminator network trained to predict collision probabilities based on the assessed outcomes, which we term "difficulty predictor". Subsequently, we leverage the predicted labels indicating risk levels to reweight sampling within the scenario library. This provides AV with individualized curricula that align with its current capability. (3) **AV Training**: The AV model is then trained using the scenarios obtained through individualized re-sampling. Overall, CLIC adheres

†Work done with equal contribution. {nhy22, xyz20}@mails.tsinghua.edu.cn
[1]Department of Automation, Tsinghua University. Correspondence to: Jianming Hu. hujm@mail.tsinghua.edu.cn
Work supported by National Natural Science Foundation of China under Grant No. 62333015 and Beijing Natural Science Foundation L231014.
Source code and supplementary materials are available at https://sites.google.com/view/icra2024clic.

to the principles of continual and curriculum learning by commencing training with simpler samples and progressively introducing more challenging ones, achieving scenario transfer and adaptation out of those diverse yet poorly organized historical scenario libraries for closed-loop AV optimization. The algorithmic design also helps safeguard against catastrophic forgetting [24], [25] of AV model. Against several competing curriculum-based baselines, experimental results demonstrate that CLIC effectively optimizes AV models to handle more challenging scenarios while minimizing any degradation in performance for easier cases.

II. Related Work

A. Scenario-Based Autonomous Driving

Scenario-based autonomous driving has recently become a popular paradigm for training and testing autonomous driving models. Currently, the focus of research in this area mainly lies in scenario generation and tesing AV model, particularly generating extreme scenarios which are rare in NDD by training RL models to control background vehicle (BV). Adaptive Stress Testing [10], [26]–[28] adopts Monte Carlo tree search and DRL to generate extreme scenarios. Feng et al. [5]–[8] propose a unified framework for adaptive testing scenario library generation. Bayesian optimization and DRL are employed in different cases. AdvSim [3] and KING [29] use a constructed adversarial cost function to explore and generate safety-critical scenarios. (Re)^{2}H2O [12] efficiently generates varied adversarial scenarios by combining NDD with simulation data through hybrid offline-and-online RL.

The scenarios generated from the above methods can be stored by saving the state information of all vehicles at each time step as a static scenario library, which is stable and easy to reproduce. However, according to our research, there is currently a lack of work that utilizes existing large-scale static scenario libraries for training AV models, thus failing to establish an industrial closed-loop of scenario generation, AV training and testing. Our goal is to complete the missing step in this closed loop, achieving an integrated approach to train and test scenario-based AV models.

B. Curriculum Learning

Curriculum learning (CL) [30] is a training strategy involving reweighting the training data and designing a series of tasks or examples in increasing difficulty order. By gradually exposing the model to more challenging instances, CL enables better generalization performance, preventing the model from becoming trapped in local optima. In addition to its original definition [30], CL has been extended to a series of similar or expanded algorithms, such as self-paced learning (SPL) [31]–[34] and teacher-guided learning [35]–[39]. These approaches have found wide applications in both DL [40]–[43] and RL [44]–[47] domains. In addition, Prioritized Experience Replay (PER) [48] is another method in RL that involves data reweighting. Instead of reweighting each training data, it focuses on reweighting individual transitions, assigning higher priority to transitions that are deemed more important or informative for learning.

The application of CL in AV training is currently not very common, and most of them align with predefined CL [49], and distinguish curricula by discrete factors such as weather and road topology structure [50], providing future BV trajectories of different lengths [51], changing the target driving distance [52] and the quantity of BV [53], [54]. In addition, [55] utilize the value function V from RL to measure the learning potential of different tasks, and divide tasks based on the distance range from the starting point to the intersection. Most of the aforementioned works define the curriculum as discrete stages and predefine when to switch between them, which often fall short in terms of automaticity, adaptability and flexibility; and usually train the model with random traffic flow. Although [51] also employs scenario-based training, it is not applicable to our large-scale scenario library with varying levels of difficulty.

To overcome the aforementioned limitations and better adapt to our objectives and the existing scenario library, we employ the definition of **Data-level Generalized CL** [49]: Curriculum learning refers to the reweighting of the target training set distribution in T training steps. We aim to assign different weights to each scenario in the scenario library automatically, and then perform weighted sampling to obtain training scenarios as individualized curriculum. By continually updating these weights, we can achieve changes in the difficulty distribution of the static scenario library, thus enabling continual optimization of driving policy.

III. Methodology

A. Problem Formulation

1) Scenario Formulation: Our scenarios are based on driving straight on the highway and assume that within our area of interest, there is only one AV (V^0) and N BVs ($V^1, \ldots, V^N$). For each vehicle at moment t, we use its lateral and longitudinal position (x, y), velocity v and heading angle θ as the state vector, and the changes in AV's velocity and heading angle between two moments as the action vector: $\mathbf{s}_t^i = [x_t^i, y_t^i, v_t^i, \theta_t^i]$, $\mathbf{a}_t^0 = [\Delta v_t^0, \Delta\theta_t^0]$. In this way, the complete state and action vectors are $\mathbf{s}_t = [\mathbf{s}_t^0, \mathbf{s}_t^1, \ldots, \mathbf{s}_t^N]^{\mathrm{T}}, \mathbf{a}_t = [\mathbf{a}_t^0]^{\mathrm{T}}$, while $\mathbf{s}_t^{\mathrm{BV}} = [\mathbf{s}_t^1, \ldots, \mathbf{s}_t^N]^{\mathrm{T}}$ only represents the state of all BVs.

A scenario d is defined as a finite sequence of traffic scenes consisting of successive H frames, where each frame contains the state of all BVs, and the initial state of all traffic participants (including the AV and all BVs) is given at the initial time: $d = [\mathbf{s}_0, \mathbf{s}_1^{\mathrm{BV}}, \ldots, \mathbf{s}_H^{\mathrm{BV}}]$. It should be emphasized that this definition of scenarios is AV-agnostic, getting rid of the influence of other AV models with distinct behaviors used to collect those scenarios. Therefore, they can be reapplied to different AV models for training.

Due to the limited modeling precision at the trajectory level in the SUMO simulator [56] we use, we choose to manually calculate the kinematic state transitions of AV:

$$\begin{aligned} x_{t+\Delta t}^0 &= x_t^0 + v_t^0 \cos\theta_t^0 \Delta t,\ y_{t+\Delta t}^0 = y_t^0 + v_t^0 \sin\theta_t^0 \Delta t \\ v_{t+\Delta t}^0 &= v_t^0 + \Delta v_t^0,\ \theta_{t+\Delta t}^0 = \theta_t^0 + \Delta\theta_t^0 \end{aligned} \tag{1}$$

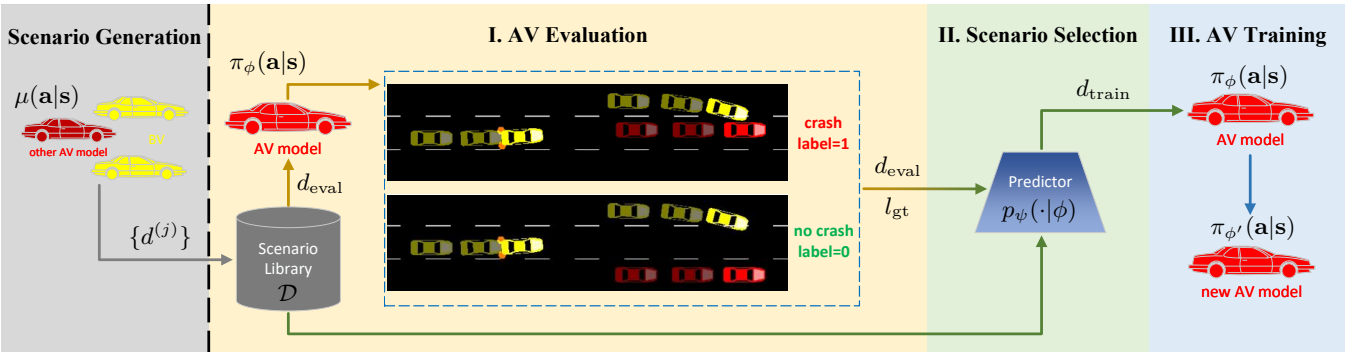

Fig. 1: Overall Algorithmic Architecture

while the state transition of BVs is directly provided by the recorded scenario data.

2) Training AV with RL: Consider the decision of AV in a traffic environment as a Markov Decision Process (MDP) defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P, \rho, \gamma)$ [57], [58], where $\mathcal{S}$, $\mathcal{A}$ and P are the state space, action space and transition probability as outlined in Section III-A.1, and ρ and γ are the initial state distribution and discount factor. Reward function $r_t = r(\mathbf{s}_t, \mathbf{a}_t) \in \mathcal{R}$ is defined as $r_t = r_{acc} + r_{vel} + r_{yaw} + r_{lane}$, which includes the following items (the values of each coefficient ρ can be found in supplementary material):

- *Accident*: Punish AV for accidents, including collisions with BV and driving off roads: $r_{acc} = -\rho_{acc} \cdot \mathbb{I}_{\{\text{AV} \in \mathcal{C}\}}$, where $\mathbb{I}$ is the indicative function, $\mathcal{C}$ represents the set of vehicles that have an accident within the current time step.
- *Velocity*: Encourage AV to drive faster within the velocity range $[v_{min}, v_{max}]$: $r_{vel} = \rho_{vel} \cdot \frac{v_t^0 - (v_{max}+v_{min})/2}{(v_{max}-v_{min})/2}$, where v_t^0 is the current velocity of AV.
- *Heading direction*: Instruct AV to drive smoothly and drive along the road direction: $r_{yaw} = -\rho_{yaw} \cdot |\theta_t^0|$, where θ_t^0 is the current heading angle of AV.
- *Lane selection*: Encourage AV to drive on the "best" lane, where the distance between AV and the nearest BV in front of AV on the same lane is the maximum: $r_{lane} = \rho_{lane} \cdot \mathbb{I}_{\{\text{AV on the best lane}\}}$.

Although our scenario library is static, the transition distribution used for training varies as the AV policy changes, because the AV state information is not included in the scenario library but is instead populated through online rollouts to align with the current AV policy. Thus, we choose the online RL algorithm Soft Actor-Critic (SAC) [59], [60] to solve the MDP problem. The objective function of SAC includes an entropy term $\mathcal{H}$ of the AV policy distribution:

$$J_\pi(\phi) = \sum_{t=0}^{H} \mathbb{E}_{\mathbf{s}_0 \sim \rho, \mathbf{a}_t \sim \pi_\phi(\cdot|\mathbf{s}_t), \mathbf{s}_{t+1} \sim P(\cdot|\mathbf{s}_t, \mathbf{a}_t)} \left[\gamma^t r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi_\phi(\cdot|\mathbf{s}_t))\right] \tag{2}$$

where π_ϕ is the AV policy parameterized by ϕ, and α is the temperature hyperparameter in the algorithm. This enables the exploration of more possible actions while maximizing the expected reward discounted by γ, which enhances the robustness of the model and is also more consistent with the application patterns in real-world traffic scenarios.

Algorithm 1 Universal Algorithmic Framework

1: **Initialize**: AV policy $\pi_\phi(\mathbf{a}|\mathbf{s})$
2: **Input**: static scenario library $\mathcal{D}$
3: **for** each iteration **do**
4: $\quad$ `info` $\leftarrow$ EVALUATEAV$(\pi_\phi(\mathbf{a}|\mathbf{s}), \mathcal{D})$
5: $\quad d_{\text{train}} \leftarrow$ SELECTSCENARIO$(\mathcal{D}$, `info`$)$
6: $\quad \pi_\phi(\mathbf{a}|\mathbf{s}) \leftarrow$ TRAINAV$(\pi_\phi(\mathbf{a}|\mathbf{s}), d_{\text{train}})$

B. Algorithmic Framework

We refine our approach into a universal modular framework (Algorithm 1), and provide our specific definitions for each module based on this.

In our universal algorithmic framework, each module is undefined and expandable, which allows us to focus on the relationships between modules and the flow of data, while ignoring the implementation details of each module. It also facilitates the adoption of different specific implementations within our framework in order to propose new methods.

Our universal algorithmic framework consists of three main stages: AV Evaluation, Scenario Selection, and AV Training. In the **AV Evaluation** stage, we evaluate the current AV model to obtain an estimate of the current capabilities of the AV model, denoted as `info` in Algorithm 1. The `info` here can take any form of evaluation information, depending on how the evaluation process and its corresponding output are defined. In the **Scenario Selection** stage, training scenarios are selected from the scenario library based on a certain criterion, in order to explore the boundaries of the AV model's capabilities and achieve the maximum improvement in AV performance. In the **AV Training** stage, any RL algorithm can be used to improve the AV policy in the training scenarios. These three stages are executed sequentially and stop after several iterations.

C. Algorithmic Implementation

1) AV Evaluation: In this stage, we aim to evaluate the capabilities of the current AV model $\pi_\phi(\mathbf{a}|\mathbf{s})$ using a subset of scenarios d_{eval} from the scenario library. Considering that the time cost of this stage should not be too high and the unbiasedness of the scenario distribution should be maintained, we randomly sample m scenarios from the scenario library $\mathcal{D}$ with a total of M scenarios, and sequentially rollout them in the environment to interact with the current AV model.

Algorithm 2 EVALUATEAV

Initialize: evaluation scenario number m
Input: current AV policy $\pi_\phi(\mathbf{a}|\mathbf{s})$, scenario library $\mathcal{D}$
$d_{\text{eval}} \leftarrow \text{SAMPLE}(\mathcal{D}, m)$
for $d^{(j)}$ in d_{eval} **do**
  $\mathbf{s} \leftarrow d^{(j)}_{\mathbf{s}_0}$; done $\leftarrow$ False
  while not done **do**
    $\mathbf{a} \leftarrow \arg\max_{\mathbf{a}} \pi_\phi(\mathbf{a}|\mathbf{s})$
    $\mathbf{s}', r,$ done $\leftarrow \text{STEP}(\mathbf{a})$; $\mathbf{s} \leftarrow \mathbf{s}'$
  $l^{(j)}_{\text{gt}} \leftarrow \mathbb{I}_{\{\text{AV}\in\mathcal{C}\}}$
return l_{gt}

Algorithm 3 SELECTSCENARIO

Initialize: difficulty predictor $p_\psi(\cdot|\phi)$, learning rate α, training scenario number n
Input: scenario library $\mathcal{D}$, evaluation scenarios d_{eval}, labels of evaluation scenarios l_{gt}
for each epoch **do**
  $l_{\text{pred}} \leftarrow p_\psi(d_{\text{eval}}|\phi)$
  $\psi \leftarrow \psi - \alpha\nabla_\psi \mathcal{L}_{\text{pred}}(\psi)$ ▷ Equation 3
$l_{\text{all}} \leftarrow p_\psi(\mathcal{D}|\phi)$
$d_{\text{train}} \leftarrow \text{WEIGHTEDSAMPLE}(\mathcal{D}, n, l_{\text{all}})$ ▷ Equation 4
return d_{train}

Then we return the results of the evaluation, which are a set of labels l_{gt} indicating whether an accident has occurred in AV. Detailed algorithmic steps are shown in Algorithm 2.

2) Scenario Selection: In order to continually optimize driving policy and enhance the ability of AV to handle extreme scenarios, it is important to focus training on more challenging scenarios while including a small number of less difficult scenarios to prevent forgetting. Thus, associating the difficulty of each scenario with its sampling weight is an intuitive approach. However, currently there is no perfect benchmark to assess scenario difficulty, meanwhile scenario difficulty is also a rather individualized metric, which is related to the AV itself. Thus we make use of the results from each evaluation stage as the information for the current AV model. We treat the scenarios used for evaluation d_{eval} and their collision labels l_{gt} as the ground truth, and then employ a supervised learning approach to train a difficulty predictor model $p_\psi(d|\phi)$, where d can be any scenario that meets the definition in Section III-A.1. We utilize a three-layer multi-layer perceptrons (MLP) [61] parameterized by ψ as the predictor $p_\psi(\cdot|\phi)$, and flatten the entire scenario data d into a one-dimensional vector as input. Binary cross-entropy loss (BCE Loss) is used as the loss function:

$$\mathcal{L}_{\text{pred}}(\psi) = -\mathbb{E}_{d^{(j)}\sim d_{\text{eval}}}\left[l^{(j)}_{\text{gt}}\cdot\log(l^{(j)}_{\text{pred}})+(1-l^{(j)}_{\text{gt}})\cdot\log(1-l^{(j)}_{\text{pred}})\right] \tag{3}$$

where $l^{(j)}_{\text{pred}} = p_\psi(d^{(j)}|\phi)$ is the predicted label value of the j-th scenario $d^{(j)}$. The number of training epochs is determined through experiments to ensure that the predictor neither underfits nor overfits the training data. The predictor is then applied to the entire scenario library $\mathcal{D}$ to obtain the predicted label values l_{all} for each scenario. These predicted label values are then used as weights for weighted sampling, resulting in a batch of training scenarios d_{train}. Specifically, the probability of sampling the j-th scenario $d^{(j)}$ is:

$$P(d^{(j)} \in d_{\text{train}}) = \frac{l^{(j)}_{\text{all}}}{\sum_{i=1}^{M} l^{(j)}_{\text{all}}} = \frac{p_\psi(d^{(j)}|\phi)}{\sum_{i=1}^{M} p_\psi(d^{(j)}|\phi)} \tag{4}$$

Algorithm 2 and Algorithm 3 together form the process of individualized curriculum design.

3) AV Training: In this stage, we employ online RL algorithm SAC to train the AV model. At each epoch, we sequentially rollout each scenario in d_{train} and update the parameters ϕ of the AV model $\pi_\phi(\mathbf{a}|\mathbf{s})$ based on SAC.

By incorporating the specific implementation of the aforementioned module into the universal algorithmic framework, we obtain an overall architecture illustrated in Figure 1, which includes the three modules mentioned above. The scenario generation stage for other AV model $\mu(\mathbf{a}|\mathbf{s})$ on the left of the figure has already been extensively studied and is therefore outside the scope of this algorithm. In each training iteration, a batch of scenarios d_{eval} is randomly sampled from the static scenario library $\mathcal{D}$ to evaluate the current AV model. These scenarios are rolled out in the environment and interacts with AV to obtain collision labels l_{gt}. These scenarios and labels are then used to train the difficulty predictor $p_\psi(\cdot|\phi)$, which, after training, predicts labels for all scenarios in $\mathcal{D}$. These predicted labels l_{all} are used as weights to perform weighted sampling on all scenarios, resulting in the training scenarios d_{train}. Finally, the AV model $\pi_\phi(\mathbf{a}|\mathbf{s})$ is trained using an online RL algorithm on these training scenarios, completing a full training iteration.

IV. EXPERIMENTS

We provide evidence for the superiority of CLIC in training AV models by addressing the following questions:

- Can CLIC train safer AV models than other baselines? (Section IV-B.1)
- Is CLIC strengthening the AV model while increasing the difficulty of training scenarios selected by the predictor throughout the training process? (Section IV-B.2)
- How does CLIC reweight the data distribution of the whole scenario library according to the current AV capability? (Section IV-B.3)
- Can CLIC select training scenarios specifically tailored to the defects of a particular vehicle? (Section IV-B.4)

A. Experiment Settings

1) Dataset: All traffic scenarios used in our experiments are generated by (Re)^{2}H2O [12]. Refer to supplementary material for more details about the dataset.

2) Baselines: We select five baselines for comparison:

- ***SAC w/ rand***: Simply sample a batch of training scenarios randomly from the scenario library.
- ***SAC w/ rand+fail***: In each training iteration, half of the training scenarios are randomly sampled from the scenario library, while another half consists of scenarios that failed in the corresponding AV Evaluation stage.
- ***SAC w/ fail***: All training scenarios are chosen from those that fail in AV Evaluation stage.

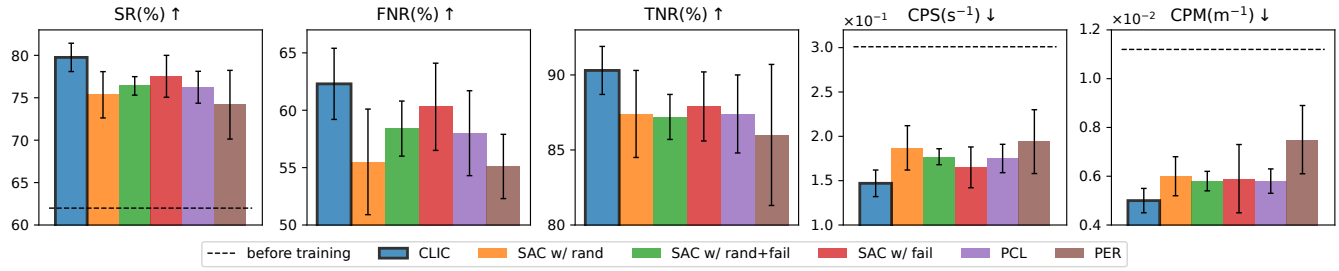

Fig. 2: Comparison of various metrics for CLIC and baselines.

- ***PCL***: Following the idea of predefined curriculum learning (PCL) [49], we divide the training process into four phases of the curriculum based on the number of BV (1∼4). The training set for each phase adds scenarios with a higher number of BV based on the previous phase.
- ***PER***: Apply PER [48] into RL based on *SAC w/ rand*.

3) Evaluation Metrics: To evaluate the capabilities and training effectiveness of the AV model in extreme scenarios, we employed the following metrics[1]:

- **SR** (%): Success rate, referring to the success rate of the AV model in all scenarios after the final iteration of training. It reflects the overall safety performance of the AV model in various extreme scenarios.
- **Confusion Matrix**: To distinguish between the difficulty levels of different scenarios and to highlight the effectiveness of the AV model in high-difficulty scenarios, we also borrow the concept of confusion matrix in supervised learning: we define the label of successful scenarios as 0, and that of failed scenarios as 1. The test results before training are treated as ground truth, while those after training are treated as predicted values. Specifically, we compute the following two metrics:
  - **FNR** (%) $= \frac{\mathrm{FN}}{\mathrm{TP+FN}}$: The proportion of successful scenarios after training that are failed before.
  - **TNR** (%) $= \frac{\mathrm{TN}}{\mathrm{TN+FP}}$: The proportion of successful scenarios after training that are successful before.
- **CPS** (s^{-1}) and **CPM** (m^{-1}): As defined in [12], CPS is Average Collision Frequency Per Second and CPM is Average Collision Frequency Per Meter: CPS $= \frac{N_{acc}}{T_{total}}$, CPM $= \frac{N_{acc}}{D_{total}}$, where N_{acc} is the number of scenarios in which AV has an accident during testing, T_{total} and D_{total} are the total testing time and the total driving distance traveled by AV along the road direction.

B. Experimental Results

1) Comparison Experiment: We conduct tests on all scenarios in the scenario library to compare different baselines on the aforementioned metrics, and present the results from 5 random seeds as shown in Figure 2. CLIC outperforms all baselines in all metrics, achieving the highest SR, FNR, TNR and the lowest CPS and CPM. This indicates that our training method not only enables the AV model to learn more challenging scenarios it couldn't handle before but also minimizes the risk of forgetting previously learned scenarios to the utmost. Although *SAC w/ fail* focuses on training failure scenarios and achieves relatively good results on FNR, it is also easy to forget simple scenarios, leading to a little bad performance on TNR. Furthermore, *SAC w/ rand* and *PER* perform the worst, further illustrating the importance of a well-designed curriculum compared to blindly training on randomly sampled scenarios from the scenario library. We also included the SR, CPS, and CPM of the AV model with random initialization before training in the figure. It can be observed that even for an untrained AV model, its SR is already quite high (62%). Therefore, using random sampling of scenarios for training like *SAC w/ rand* and *PER* would inevitably lead to inefficient training, wasting time on scenarios that the AV model does not require training on.

2) Matrix Experiment: In order to verify the changes in the capabilities of the AV model and the difficulty of scenarios selected by the predictor throughout the training process, we conduct the following matrix experiments: we save the AV model and predictor model at each training iteration, and sequentially use each predictor to select scenarios and test them with each AV model. This process yield a $T \times T$ matrix of SR, where T represents the total number of training iterations. For instance, the element "*AV iteration* = 3, *predictor iteration* = 4" represents the testing SR of the AV model trained after three iterations when tested on the scenarios selected by the predictor in the fourth iteration. As shown in Figure 3, the experimental results demonstrate the following characteristics:

- The AV models with a higher number of training iterations achieved higher SR among the scenarios selected by each specific predictor, indicating that the capability of the AV model is indeed continually increasing.
- For each specific AV model, there is a decreasing trend in SR among the scenarios selected by subsequent predictors, indicating that the difficulty of the scenarios selected by the predictor is indeed increasing.
- For the AV model in the first training iteration, it exhibits a higher SR on scenarios selected by some of subsequent predictors. This is primarily due to the lower capability of the AV model at this stage, rendering the difficulty level of predictions by the predictors insignificant. This observation further emphasizes the importance of individualized curriculum design in CLIC.

3) Analyses on Scenario Reweighting: To showcase the changes in the distribution of scenario weights before and after training, we tracked the changes in predicted label

[1]We expect higher values for SR, FNR and TNR, and lower values for CPS and CPM, which indicates that the AV model is safer.

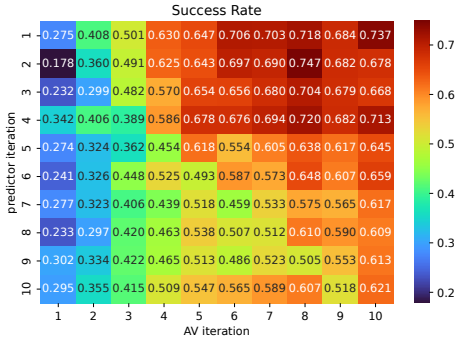

Fig. 3: Results of the matrix experiment.

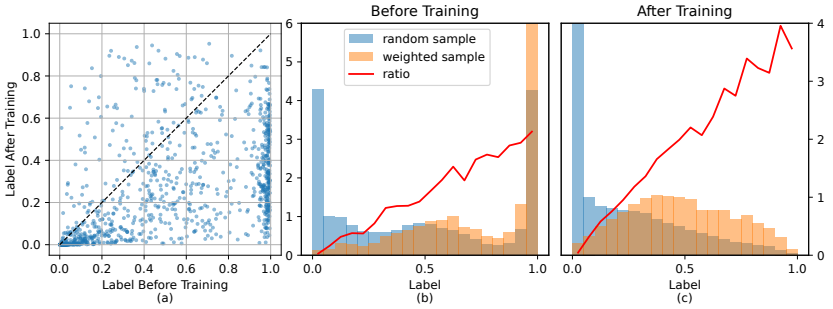

Fig. 4: **(a)** Changes of partial individual scenario labels before and after training. **(b)(c)** Label distribution before and after training.

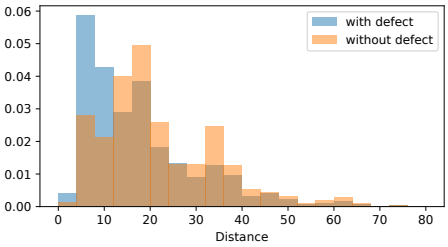

Fig. 5: Distribution of distances between the AV and the BVs on its left front side.

values for individual scenarios before and after training. These changes are presented as a scatter plot (Figure 4(a)). The scenarios in the figure are concentrated in two regions: First, scenarios with initially small label values tend to be concentrated around 0 after training; Second, scenarios with label values close to 1 before training show varying degrees of decrease in label values after training. Collectively, the majority of scenarios fall below the diagonal line $y = x$, indicating that these scenarios indeed become simpler for the AV model after training.

Moreover, to further illustrate the overall scenario reweighting, we also plot histograms of the label distributions obtained through random sampling and weighted sampling before and after training, as shown in Figure 4(b)(c). It can be observed that before training, the predicted labels are concentrated at both ends of the interval, while the rest of the distribution appears relatively uniform. However, with weighted sampling, the labels are concentrated near 1, indicating the selection of harder scenarios for the current AV model. After training, as the AV model becomes sufficiently powerful, the predicted labels near 1 significantly decrease, and more labels concentrate near 0. Nonetheless, in order to select relatively hard training scenarios, weighted sampling still acquires a considerable number of scenarios with larger labels. To provide a more intuitive demonstration of the weighted sampling, we also plot the ratio curve of the normalized frequencies corresponding to the two sampling methods. As expected, the ratio curve maintains a proportional relationship for each sampling instance, which is consistent with the setting of Equation 4.

4) Analyses on Individualization: To demonstrate that CLIC can indeed generate individualized curricula that identify specific defects in AV model and select more targeted training scenarios accordingly, we intentionally disable a trained AV model to get less aware of BVs on the left front side, then let this disabled AV go through CLIC pipeline and compare the selected scenarios with the ones selected for the AV model without defects. We analyze the distribution of distances between the AV and the BVs on its left front side in these scenarios illustrated in Figure 5. As expected, these BVs are positioned closer to the AV, increasing the risk of collisions. Furthermore, CLIC selected a significantly higher proportion (**27.87%**) of scenarios for AVs with defect where BVs are positioned on the left front side, surpassing the selected proportion (**17.19%**) of the AV model with no defect by a substantial margin. These clearly echo our intuition of providing individualized curricula that target specific defects in training AV.

V. Conclusion and Future Work

In this paper, we develop a scenario-based continual driving policy optimization framework with Closed-Loop Individualized Curricula (CLIC) technique, composed of three sub-modules for flexible implementation choices: AV Evaluation, Scenario Selection and AV Training. CLIC approaches AV Evaluation as a difficulty prediction task by training a discriminator on AV collision labels to estimate the potential failure probability of AV within corresponding scenarios. With the prediction results from the discriminator, CLIC reweights scenarios to select individualized curricula for AV training that incorporate less easy cases with more challenging ones according to current AV capability. To summarize, CLIC not only fully exploits the vast historical scenario library for closed-loop AV training rather than just for AV testing, but also facilitates AV improvement by individualizing its training on more helpful scenarios that match its current capability out of the diverse yet poorly organized library. Through extensive experimental analyses, CLIC outperforms other competing baselines, especially in handling safety-critical scenarios, while minimizing the degradation of performance in regular scenarios. For future work, we are interested in exploring scenario libraries with more complex road topologies and other types of traffic participants, as well as investigating theoretical guarantees for AV improvement within these individualized curricula. We also believe that CLIC is not only applicable to the field of autonomous driving policy optimization, but also a general method for complex tasks and continual learning in robotics.

REFERENCES

[1] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[2] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016. 1

[3] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, "Advsim: Generating safety-critical scenarios for self-driving vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9909–9918. 1, 2

[4] W. Ding, B. Chen, M. Xu, and D. Zhao, "Learning to collide: An adaptive safety-critical scenarios generating method," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2243–2250. 1

[5] S. Feng, Y. Feng, C. Yu, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part i: Methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1573–1582, 2020. 1, 2

[6] S. Feng, Y. Feng, H. Sun, S. Bao, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part ii: Case studies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5635–5647, 2020. 1, 2

[7] S. Feng, Y. Feng, H. Sun, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles: An adaptive framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1213–1222, 2020. 1, 2

[8] H. Sun, S. Feng, X. Yan, and H. X. Liu, "Corner case generation and analysis for safety assessment of autonomous vehicles," *Transportation research record*, vol. 2675, no. 11, pp. 587–600, 2021. 1, 2

[9] H. Niu, J. Hu, Z. Cui, and Y. Zhang, "Dr2l: Surfacing corner cases to robustify autonomous driving via domain randomization reinforcement learning," in *Proceedings of the 5th International Conference on Computer Science and Application Engineering*, 2021, pp. 1–8. 1

[10] R. Lee, O. J. Mengshoel, A. Saksena, R. W. Gardner, D. Genin, J. Silbermann, M. Owen, and M. J. Kochenderfer, "Adaptive stress testing: Finding likely failure events with reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 69, pp. 1165–1201, 2020. 1, 2

[11] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 305–17 315. 1

[12] H. Niu, K. Ren, Y. Xu, Z. Yang, Y. Lin, Y. Zhang, and J. Hu, "(re)2h2o: Autonomous driving scenario generation via reversely regularized hybrid offline-and-online reinforcement learning," in *2023 IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–8. 1, 2, 4, 5

[13] D. Nalic, T. Mihalj, M. Bäumler, M. Lehmann, A. Eichberger, and S. Bernsteiner, "Scenario based testing of automated driving systems: A literature survey," in *FISITA web Congress*, vol. 10, 2020. 1

[14] X. Li, "A scenario-based development framework for autonomous driving," *arXiv preprint arXiv:2011.01439*, 2020. 1

[15] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023. 1

[16] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 3, pp. 595–607, 2016. 1

[17] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2017. 1

[18] Z. Huang, H. Lam, D. J. LeBlanc, and D. Zhao, "Accelerated evaluation of automated vehicles using piecewise mixture models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2845–2855, 2017. 1

[19] Z. Zhong, Y. Tang, Y. Zhou, V. d. O. Neves, Y. Liu, and B. Ray, "A survey on scenario-based testing for automated driving systems in high-fidelity simulation," *arXiv preprint arXiv:2112.00964*, 2021. 1

[20] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8. 1

[21] J. Yang, H. Sun, H. He, Y. Zhang, H. X. Liu, and S. Feng, "Adaptive safety evaluation for connected and automated vehicles with sparse control variates," *IEEE Transactions on Intelligent Transportation Systems*, 2023. 1

[22] Z. Cao, S. Xu, H. Peng, D. Yang, and R. Zidek, "Confidence-aware reinforcement learning for self-driving cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7419–7430, 2021. 1

[23] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, 2023. 1

[24] M. Toneva, A. Sordoni, R. T. d. Combes, A. Trischler, Y. Bengio, and G. J. Gordon, "An empirical study of example forgetting during deep neural network learning," *arXiv preprint arXiv:1812.05159*, 2018. 2

[25] K. Khetarpal, M. Riemer, I. Rish, and D. Precup, "Towards continual reinforcement learning: A review and perspectives," *Journal of Artificial Intelligence Research*, vol. 75, pp. 1401–1476, 2022. 2

[26] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1–7. 2

[27] M. Koren and M. J. Kochenderfer, "Efficient autonomy validation in simulation with adaptive stress testing," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 4178–4183. 2

[28] A. Corso, P. Du, K. Driggs-Campbell, and M. J. Kochenderfer, "Adaptive stress testing with reward augmentation for autonomous vehicle validatio," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 163–168. 2

[29] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," in *European Conference on Computer Vision*. Springer, 2022, pp. 335–352. 2

[30] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48. 2

[31] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," *Advances in neural information processing systems*, vol. 23, 2010. 2

[32] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. Hauptmann, "Self-paced curriculum learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015. 2

[33] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2216–2226, 2018. 2

[34] P. Klink, C. D'Eramo, J. R. Peters, and J. Pajarinen, "Self-paced deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9216–9227, 2020. 2

[35] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, "Teacher–student curriculum learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3732–3740, 2019. 2

[36] R. Portelas, C. Colas, K. Hofmann, and P.-Y. Oudeyer, "Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments," in *Conference on Robot Learning*. PMLR, 2020, pp. 835–853. 2

[37] C. Romac, R. Portelas, K. Hofmann, and P.-Y. Oudeyer, "Teachmyagent: a benchmark for automatic curriculum learning in deep rl," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9052–9063. 2

[38] Y. Schraner, "Teacher-student curriculum learning for reinforcement learning," *arXiv preprint arXiv:2210.17368*, 2022. 2

[39] I. Shenfeld, Z.-W. Hong, A. Tamar, and P. Agrawal, "Tgrl: Teacher guided reinforcement learning algorithm for pomdps," in *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. 2

[40] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, "Automated curriculum learning for neural networks," in *international conference on machine learning*. Pmlr, 2017, pp. 1311–1320. 2

[41] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International conference on machine learning*. PMLR, 2018, pp. 2304–2313. 2

[42] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," in *International conference on machine learning*. PMLR, 2019, pp. 2535–2544. 2

[43] T.-H. Kim and J. Choi, "Screenernet: Learning self-paced curriculum for deep neural networks," *arXiv preprint arXiv:1801.00904*, 2018. 2

[44] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," in *Conference on robot learning*. PMLR, 2017, pp. 482–495. 2

[45] C. Florensa, D. Held, X. Geng, and P. Abbeel, "Automatic goal generation for reinforcement learning agents," in *International conference on machine learning*. PMLR, 2018, pp. 1515–1528. 2

[46] P. Klink, C. D'Eramo, J. Peters, and J. Pajarinen, "Boosted curriculum reinforcement learning," in *International Conference on Learning Representations*, 2021. 2

[47] Y. Cai, C. Zhang, H. Zhao, L. Zhao, and J. Bian, "Curriculum offline reinforcement learning," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 1221–1229. 2

[48] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015. 2, 5

[49] X. Wang, Y. Chen, and W. Zhu, "A survey on curriculum learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021. 2, 5

[50] A. Ozturk, M. B. Gunel, R. Dagdanov, M. E. Vural, F. Yurdakul, M. Dal, and N. K. Ure, "Investigating value of curriculum reinforcement learning in autonomous driving under diverse road and weather conditions," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 358–363. 2

[51] S. Khaitan and J. M. Dolan, "State dropout-based curriculum reinforcement learning for self-driving at unsignalized intersections," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 219–12 224. 2

[52] P. Agarwal, P. De Beaucorps, and R. De Charette, "Sparse curriculum reinforcement learning for end-to-end driving," *arXiv preprint arXiv:2103.09189*, 2021. 2

[53] L. Anzalone, S. Barra, and M. Nappi, "Reinforced curriculum learning for autonomous driving in carla," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3318–3322. 2

[54] L. Anzalone, P. Barra, S. Barra, A. Castiglione, and M. Nappi, "An end-to-end curriculum learning approach for autonomous driving scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19 817–19 826, 2022. 2

[55] Z. Qiao, K. Muelling, J. M. Dolan, P. Palanisamy, and P. Mudalige, "Automatically generated curriculum based reinforcement learning for autonomous vehicles in urban environment," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1233–1238. 2

[56] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011. 2

[57] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135. 3

[58] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 3

[59] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018. 3

[60] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870. 3

[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016. 4