

# Learning the References of Online Model Predictive Control for Urban Self-Driving

Yubin Wang, Zengqi Peng, Hakim Ghazzai, and Jun Ma

**Abstract**—In this work, we propose a novel learning-based online model predictive control (MPC) framework for motion synthesis of self-driving vehicles. In this framework, the decision variables are generated as instantaneous references to modulate the cost functions of online MPC, where the constraints of collision avoidance and drivable surface boundaries are latently represented in the soft form. Hence, the embodied maneuvers of the ego vehicle are empowered to adapt to complex and dynamic traffic environments, even with unmodeled uncertainties of other traffic participants. Furthermore, we implement a deep reinforcement learning (DRL) framework for policy search to cast the step actions as the decision variables, where the practical and lightweight observations are considered as the input features of the policy network. The proposed approach is implemented in the high-fidelity simulator involving compound-complex urban driving scenarios, and the results demonstrate that the proposed development manifests remarkable adaptiveness to complex and dynamic traffic environments with a success rate of 85%. Also, its advantages in terms of safety, maneuverability, and robustness are illustrated.

## I. INTRODUCTION

Safe and efficient driving strategies are incredibly essential for the wider adoption of self-driving technology in urban and residential scenarios. Nevertheless, the relationship between the strong safety guarantee and high driving efficiency is a trade-off, as conservative maneuvers could sacrifice efficiency while aggressive driving strategies lack the safety guarantee. Hence, it is crucial to develop an advanced motion synthesis strategy for self-driving vehicles to ensure the satisfaction of collision avoidance constraints, perform agile maneuvers, and exhibit robustness to uncertainties regarding the unpredictable behaviors of other traffic participants.

Optimization-based methods, particularly the model predictive control (MPC), have been widely studied due to their capability to optimize the feasible trajectory respecting various constraints. Essentially, the urban self-driving problem can be formulated as an optimization problem with nonlinear vehicle dynamics and various constraints of other types [1], [2]. However, the performance attained by such methods is degraded due to the conservative motions and undesirable driving behaviors, especially in traffic environments with high complexity and dynamicity. On the other hand, as a representative reinforcement learning (RL) method, deep reinforcement learning (DRL) aims to learn a neural network policy that can map high-dimensional raw observation features directly

Yubin Wang, Zengqi Peng, and Jun Ma are with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China (email: ywang575@connect.hkust-gz.edu.cn; zpeng940@connect.hkust-gz.edu.cn; jun.ma@ust.hk)

Hakim Ghazzai is with the Division of Computer, Electrical and Mathematical Science and Engineering, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia (email: Hakim.Ghazzai@kaust.edu.sa)

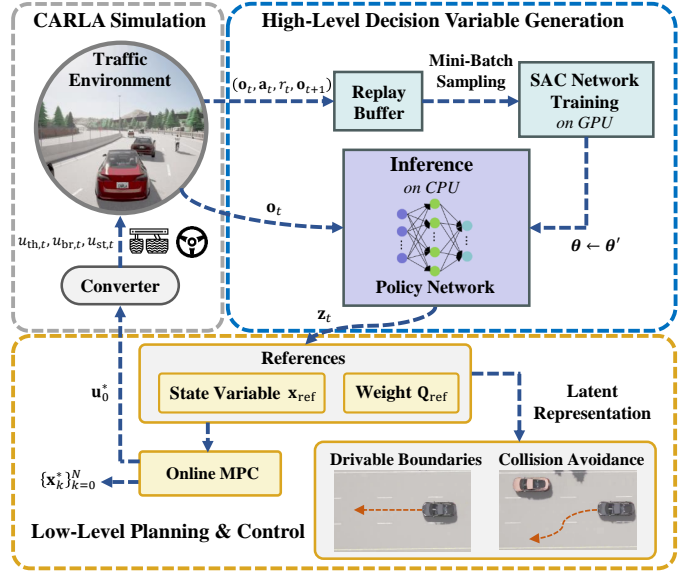


Fig. 1. Overview of the proposed framework for urban self-driving. A policy network is trained to produce instantaneous decision variables for low-level online MPC, whose cost functions are modulated to latently represent the constraints of collision avoidance and drivable surface boundaries.

to control commands [3], [4]. Apparently, such RL-based methods demonstrate the advantages of forgoing the need for dynamics modeling and online optimization [3]. However, the instability and low generalizability of the learned policy are the commonly encountered issues that hinder their applications. Furthermore, a promising way that facilitates effective and reliable driving strategies lies in the design of a hybrid framework that bridges MPC and DRL, in which the neural policy produces decision variables as references to formulate cost terms of the optimization problem in the MPC scheme. Thus, it facilitates improved maneuverability of the vehicle, and also ensures the feasibility as well as the reliability of the generated trajectories.

In this paper, we propose a novel learning-based online MPC framework for urban self-driving, where the constraints of collision avoidance and drivable road surface boundaries are latently represented by modulating costs with references in the form of decision variables. Furthermore, the policy search for the real-time generation of decision variables is cast to a DRL problem, where the policy network produces the step actions as decision variables for the MPC. Here, the soft actor-critic (SAC) algorithm is employed in the proposed framework to update the policy, where training data is uniformly sampled from the replay buffer. The overview of the proposed framework is depicted in Fig. 1. The main contributions of this work are summarized in three folds as follows:

- A novel learning-based MPC framework exhibiting high adaptiveness to dynamicity, complexity, and uncertainties of urban traffic environments is proposed, where the nonlinear and nonconvex constraints in terms of collision avoidance and drivable road boundaries are transformed into a latent term by modulating the cost functions with instantaneous references in the form of decision variables.
- The real-time generation of decision variables is formulated as a sequential decision-making problem, where an RL agent is designed to take the step actions as decision variables with input features of practical and lightweight observations.
- The proposed framework is implemented and evaluated in complex urban driving scenarios with the use of a high-fidelity simulator, where the results demonstrate the advantages in terms of maneuverability of self-driving vehicles, robustness to traffic uncertainties, and the superiority in driving efficiency and safety guarantee over baselines.

## II. RELATED WORKS

Learning-based MPC for motion synthesis is an emerging technique in the area of self-driving and other types of robotic applications, where appropriate learning techniques can be incorporated to model or parameterize the critical factors of MPC. A typical approach in learning-based MPC aims to model complex systems with classical machine learning or deep learning techniques. Gaussian process (GP) can be utilized to represent the model error to improve the simple nominal vehicle dynamics model online [5], or approximate the nonlinearities of the dynamics in the presence of uncertainty [6], where the solution quality of MPC is improved and the computational burden is alleviated. Furthermore, an approach is proposed in [7], which utilizes GP to model aerodynamic effects and incorporates it into an MPC framework for accurate and high-speed trajectory tracking. On the other hand, deep neural networks can also be applied for accurate predictions of the system dynamics, where knowledge-based neural ordinary differential equations are adopted to account for residual and uncertain dynamics for quadrotors in [8]. Also, a dynamics model represented by large-scale and complex neural networks is integrated within an MPC pipeline for high-speed and aggressive quadrotor maneuvers in [9]. Additionally, in [10], a deep neural network architecture is utilized to represent the variant dynamics model for robotic tasks in an online MPC framework, where the MPC is empowered to adaptively tune the dynamics for variant tasks.

The other paradigm of learning-based MPC is to utilize RL techniques to formulate the cost functions of MPC. In [11], the global value function learning is exploited to approximate the terminal cost of MPC, which allows for better policy quality beyond local solutions within a reduced planning horizon. Similarly, in [12], value learning is utilized to approximate the stage and terminal costs of MPC from scratch without human intervention, even with sparse or binary objectives. Furthermore, RL can also be integrated into the MPC framework to learn the high-level decision variables for cost formulation.

In [13] and [14], a high-level policy with the representation of Gaussian distribution is proposed, with which the traversal time of flying through a swinging gate for a quadrotor is determined. Furthermore, SE(3) decision variables are learned as state references of the MPC in [15], with which the quadrotor can traverse a moving and rotating gate. In [16], the augmented decision variables are introduced to parameterize the cost functions of high-level MPC for the task of chance-aware lane change in dense traffic environments. Despite the success of producing desired decision variables for MPC in [13]–[16], the high-level policy is trained by episodic RL through evaluating the quality of whole trajectories. Essentially, the delayed feedback and the difficulty in credit assignment degrade the performance of episodic RL in long-term decision-making. This is because it is challenging to determine which actions should be responsible for the final outcomes, and the agent cannot adapt its behavior at specific steps based on cumulative and delayed reward signals. Therefore, formulating the generation of decision variables as an episodic RL problem could potentially hinder the maneuverability and safety of vehicles in highly dynamic and complex traffic environments. To address this issue, we aim to incorporate the step-based RL for policy search, where the actions evolve with intermediate reward signals. Furthermore, we formulate the real-time generation of decision variables as a sequential decision-making problem. In this sense, the proposed framework is able to demonstrate improved adaptiveness to the high complexity and dynamicity of traffic environments in urban self-driving tasks.

## III. PRELIMINARIES AND PROBLEM STATEMENT

### A. Vehicle Model and Coordinate Transformation

In this work, the bicycle model is adopted to describe the vehicle’s kinematics. The state vector of the vehicle in the global coordinate  $\mathcal{W}_g$  is defined as  $\mathbf{X} = [X \ Y \ \Psi \ V]^\top$ , where  $X$  and  $Y$  denote the X-coordinate and Y-coordinate position of the center of the vehicle,  $\Psi$  is the heading angle, and  $V$  is the speed. Also, we integrate the control inputs into a vector as  $\mathbf{u} = [a \ \delta]^\top$ , where  $a$  and  $\delta$  are the acceleration and steering angle. Subsequently, by modeling the vehicle as a rectangle, the nonlinear kinematic model of the vehicle in continuous time is given by:

$$\dot{\mathbf{X}} = f(\mathbf{X}, \mathbf{u}) = \begin{bmatrix} V \cos(\Psi + \delta) \\ V \sin(\Psi + \delta) \\ \frac{2V}{L} \sin \delta \\ a \end{bmatrix}, \quad (1)$$

where  $L$  is the inter-axle distance of the vehicle.

For the convenience of formulating the optimization problem in general urban self-driving scenarios, we exploit a transformation of the vehicle state vector  $\mathbf{X}$  from the global coordinate  $\mathcal{W}_g$  into the road centerline reference coordinate  $\mathcal{W}_{\text{ref}}$ . It is assumed that the two-dimensional centerline of the road  $\mathcal{P}_{\text{ref}}$  is detected and the length of centerline  $|\mathcal{P}_{\text{ref}}|$  is parameterized by the longitudinal distance  $\lambda$  from its start, the point on the centerline can be defined as  $(X^{\mathcal{P}_{\text{ref}}}(\lambda), Y^{\mathcal{P}_{\text{ref}}}(\lambda))$ , where  $\lambda \in [0, |\mathcal{P}_{\text{ref}}|]$ . The tangential and normal vectors of the

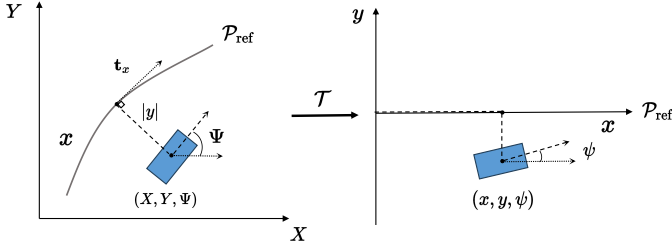


Fig. 2. Illustration of the pose transformation  $\mathcal{T}$  from the global coordinate frame  $\mathcal{W}_g$  to the centerline reference coordinate frame  $\mathcal{W}_{\text{ref}}$ .

centerline in the global coordinate  $\mathcal{W}_g$ , denoted by  $\mathbf{t}_\lambda$  and  $\mathbf{n}_\lambda$ , can be written as:

$$\mathbf{t}_\lambda = \begin{bmatrix} \frac{\partial X^{\mathcal{P}_{\text{ref}}}(\lambda)}{\partial Y^{\mathcal{P}_{\text{ref}}}(\lambda)} \\ \frac{\partial \lambda}{\partial \lambda} \end{bmatrix}, \quad \mathbf{n}_\lambda = \begin{bmatrix} -\frac{\partial Y^{\mathcal{P}_{\text{ref}}}(\lambda)}{\partial X^{\mathcal{P}_{\text{ref}}}(\lambda)} \\ \frac{\partial \lambda}{\partial \lambda} \end{bmatrix}. \quad (2)$$

According to [1], we can define an invertible transformation  $\mathcal{T}$  that maps the pose of the vehicle from the global coordinate  $\mathcal{W}_g$  to centerline reference coordinate  $\mathcal{W}_{\text{ref}}$  (as shown in Fig. 2), which is given by:

$$(x, y, \psi) = \mathcal{T}(X, Y, \Psi), \quad (3)$$

where  $x$ ,  $y$ , and  $\psi$  are the X-coordinate position, Y-coordinate position, and heading angle of the vehicle in the road centerline reference coordinate  $\mathcal{W}_{\text{ref}}$ . Specifically, we have

$$x = \underset{\lambda}{\operatorname{argmin}} (X - X^{\mathcal{P}_{\text{ref}}}(\lambda))^2 + (Y - Y^{\mathcal{P}_{\text{ref}}}(\lambda))^2, \quad (4)$$

$$y = \frac{1}{\|\mathbf{n}_x\|} \mathbf{n}_x^\top \cdot \begin{bmatrix} X - X^{\mathcal{P}_{\text{ref}}}(x) \\ Y - Y^{\mathcal{P}_{\text{ref}}}(x) \end{bmatrix}, \quad (5)$$

$$\psi = \Psi - \arctan \left( \frac{\partial Y^{\mathcal{P}_{\text{ref}}}(\lambda)}{\partial X^{\mathcal{P}_{\text{ref}}}(\lambda)} \Big|_{\lambda=x} \right). \quad (6)$$

Moreover, as  $\mathcal{T}$  is a spatial transformation, the speed keeps invariant, i.e.,  $v = V$ . Thus, the vehicle state vector is redefined as  $\mathbf{x} = [x \ y \ \psi \ v]^\top$  in the centerline reference coordinate  $\mathcal{W}_{\text{ref}}$ .

### B. Problem Statement of Urban Self-Driving

Urban self-driving is a compound task consisting of overtaking, lane change, and collision avoidance in residential scenarios where other non-interactive traffic participants (such as other vehicles, motorcyclists, cyclists, and pedestrians) are considered on the road. It is pertinent to note that the trajectories of other traffic participants are with unmodeled uncertainties and their behaviors are even unpredictable. With this generalized urban driving setting, the objective of this work is to plan in the centerline reference coordinate system  $\mathcal{W}_{\text{ref}}$  for the self-driving task. This process involves the generation of a sequence of control commands for the ego vehicle to reach the destination. Additionally, it also requires increasing driving efficiency while decreasing the risk of collisions with road boundaries and other traffic participants.

Mathematically, with the given sampling time  $d_t$  and vehicle model  $f$ , a sequence of vehicle states  $\mathbf{x}_k, \forall k \in [0, 1, \dots, N]$  and control commands  $\mathbf{u}_k, \forall k \in [0, 1, \dots, N-1]$  are discretized over a prediction horizon  $N$ . Let  $\mathbf{x}_g$  denote the vehicle

terminal state of the task, the control objective is to generate the optimal state trajectory  $\boldsymbol{\xi}^* = \{\mathbf{x}_k^*\}_{k=0}^N$  towards  $\mathbf{x}_g$  and a sequence of optimal control command  $\boldsymbol{\zeta}^* = \{\mathbf{u}_k^*\}_{k=0}^N$ , while increasing the average driving speed  $\bar{v}$  and decreasing the probability of collisions with on-road static or dynamic obstacles  $p_{\text{coll}}(\mathbf{x}_k^*)$ .

## IV. ONLINE MODEL PREDICTIVE CONTROL WITH INSTANTANEOUS REFERENCES

### A. Online MPC Formulation

The urban self-driving task is formulated as a nonlinear optimization problem under the MPC scheme over the prediction horizon  $N$ . We take the target-oriented stage cost  $J_{x_k} = \|\mathbf{x}_k - \mathbf{x}_g\|_{\mathbf{Q}_x}^2$ , terminal cost  $J_{x_N} = \|\mathbf{x}_N - \mathbf{x}_g\|_{\mathbf{Q}_x}^2$ , energy consumption cost  $J_{u_k} = \|\mathbf{u}_k\|_{\mathbf{Q}_u}^2$ , driving comfort cost  $J_{\Delta u_k} = \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_{\mathbf{Q}_{\Delta u}}^2$ , and intermediate reference cost  $J_{\text{ref},k}$  into consideration, where  $\mathbf{Q}_x$ ,  $\mathbf{Q}_u$ , and  $\mathbf{Q}_{\Delta u}$  are time-invariant positive semi-definite diagonal matrices. It is highlighted that the MPC is reformulated at each decision step  $t$ , as the cost functions are modulated with the instantaneous  $J_{\text{ref},k}$ . Therefore, we integrate all cost terms, which lead to the following nonlinear optimization problem to be reformulated and solved online:

$$\begin{aligned} \min_{\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}} \quad & J_{x_N} + \sum_{k=0}^{N-1} (J_{x_k} + J_{u_k} + J_{\Delta u_k} + J_{\text{ref},k}) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{x}_k + f(\mathbf{x}_k, \mathbf{u}_k) d_t, \\ & v_{\min} \leq v_k \leq v_{\max}, \\ & a_{\min} \leq a_k \leq a_{\max}, \\ & -\delta_{\max} \leq \delta_k \leq \delta_{\max}, \end{aligned} \quad (7)$$

where the speed of the ego vehicle is limited by  $v_{\min}$  and  $v_{\max}$  according to the traffic rule and the control commands are constrained by  $a_{\min}$ ,  $a_{\max}$ , and  $-\delta_{\max}$ ,  $\delta_{\max}$  considering the physical limits of vehicle model, respectively.

### B. Latent Constraint Representation

The constraints in terms of collision avoidance with other traffic participants and drivable surface boundaries are removed in our MPC formulation. In this work, we design an instantaneous and self-tuned cost term to replace these constraints, in order to reserve and even boost their functionalities. Specifically, the constraints of collision avoidance and drivable surface boundaries can be latently represented in the soft form as follows:

$$J_{\text{ref},k} = \|\mathbf{x}_k - \mathbf{x}_{\text{ref}}\|_{\mathbf{Q}_{\text{ref}}}^2, \quad (8)$$

where

$$\mathbf{x}_{\text{ref}} = [x_{\text{ref}} \ y_{\text{ref}} \ \psi_{\text{ref}} \ v_{\text{ref}}]^\top \quad (9)$$

is the intermediate full-state variable, and

$$\mathbf{Q}_{\text{ref}} = \text{diag}([Q_{x_{\text{ref}}}, Q_{y_{\text{ref}}}, Q_{\psi_{\text{ref}}}, Q_{v_{\text{ref}}}]]) \quad (10)$$

is the time-varying positive semi-definite diagonal weighting matrix to define the relative importance of designed reference cost among all costs.

Inspired by the soft constraints in MPC, the functionality of the constraints of collision avoidance and drivable surface boundaries can be achieved by formulating a collision-free state variable  $\mathbf{x}_{\text{ref}}$  within the road boundaries and appropriate  $\mathbf{Q}_{\text{ref}}$  in  $J_{\text{ref},k}$ . Thus, effective management of the constraints of collision avoidance with other traffic participants and drivable surface boundaries can be attained by designing appropriate  $\mathbf{x}_{\text{ref}}$  and  $\mathbf{Q}_{\text{ref}}$ , ensuring the rationality and safety of future state trajectories. In other words, the self-driving vehicle can bypass the surrounding obstacles through tracking well-designed  $\mathbf{x}_{\text{ref}}$  rather than directly solving the optimization problem with aforementioned constraints in hard or soft form.

Compared to the conventional formulation, the latent representation of these constraints reduces the complexity of the computation when complex and dynamic traffic environments are encountered. Therefore, it allows for the generation of feasible and agile motions, resulting in high driving efficiency and a strong safety guarantee. Ideally, the robustness of our MPC towards the uncertainties of traffic environments is also can be improved since the instantaneous  $\mathbf{x}_{\text{ref}}$  and  $\mathbf{Q}_{\text{ref}}$  (as inputs of MPC) are well-designed according to the surrounding traffic environment. Overall, through the design of appropriate  $\mathbf{x}_{\text{ref}}$  and  $\mathbf{Q}_{\text{ref}}$ , the self-driving vehicle can flexibly adjust the driving strategy for safe and efficient self-driving in highly complex and dynamic environments while exhibiting strong robustness to traffic uncertainties.

### C. Real-Time Decision Variable Generation

In this work, we integrate  $\mathbf{x}_{\text{ref}}$  and  $\mathbf{Q}_{\text{ref}}$  into a decision vector  $\mathbf{z}$  as references to formulate the MPC, which is defined as follows:

$$\mathbf{z} = \left[ \mathbf{x}_{\text{ref}}^\top \quad \text{vec}(\mathbf{Q}_{\text{ref}})^\top \right]^\top \in \mathbb{R}^8. \quad (11)$$

Let  $f_{\text{MPC}}$  denote the mapping function of MPC. It is noted that various optimal state trajectories  $\boldsymbol{\xi}^* = \{\mathbf{x}_k^*\}_{k=0}^N$  can be generated by feeding MPC with different decision vector  $\mathbf{z}$ , where

$$\boldsymbol{\xi}^*(\mathbf{z}) = f_{\text{MPC}}(\mathbf{z}). \quad (12)$$

Therefore, it is imperative to obtain the desired decision variables as instantaneous references to formulate the MPC. Inspired by [13]–[16], the high-level policy  $\pi$  can directly map the features of observation from the traffic environments into decision variables. Hence, we can incorporate a step-based DRL technique to learn the optimal high-level policy  $\pi^*$ , such that the decision variables are automatically determined in real time.

## V. LEARNING THE DECISION VARIABLES VIA DRL

### A. Partial Observation

In the sequel, we use the notations with subscript  $t$  to represent their respective values at decision step  $t$ . With  $T$  as the total number of simulation time steps, the generation of decision variables through the inference of high-level policy  $\pi$  is a sequential decision-making problem in essence, which can be recorded as:

$$\mathcal{Z} = (\mathbf{z}_0^\top, \mathbf{z}_1^\top, \dots, \mathbf{z}_{T-1}^\top). \quad (13)$$

Table I  
OBSERVATION SPACE ( $\mathbb{R}^{4+n}$ )

$\tilde{x}_t$	Distance to terminal state of X-position in $\mathcal{W}_{\text{ref}}$	$\mathbb{R}$
$y_t$	Y-position in $\mathcal{W}_{\text{ref}}$	$\mathbb{R}$
$\psi_t$	Heading angle in $\mathcal{W}_{\text{ref}}$	$\mathbb{R}$
$v_t$	Speed in $\mathcal{W}_{\text{ref}}$	$\mathbb{R}$
$\mathbf{d}_t$	2D Lidar distance measurements ( $-90^\circ, 90^\circ, 50\text{m}$ )	$\mathbb{R}^n$

Table II  
VALUE RANGES OF DECISION VARIABLES

Variable	Interval	Variable	Interval
$x_{\text{ref},t}$	$[-20, 20]$	$Q_{x_{\text{ref},t}}$	$[0, 20]$
$y_{\text{ref},t}$	$[-10, 10]$	$Q_{y_{\text{ref},t}}$	$[0, 20]$
$\psi_{\text{ref},t}$	$[-\pi/2, \pi/2]$	$Q_{\psi_{\text{ref},t}}$	$[0, 20]$
$v_{\text{ref},t}$	$[-10, 20]$	$Q_{v_{\text{ref},t}}$	$[0, 20]$

Considering the online MPC as the low-level planner, at each decision step  $t$ , the decision variables are determined according to the features of observation from the traffic environments, as the key factors of references to modulate the cost functions of MPC. In this work, the partial observation of the RL agent at each decision step  $t$  is  $\mathbf{o}_t = (\tilde{x}_t, y_t, \psi_t, v_t, \mathbf{d}_t)$ , and it is defined in Table I. In this table,  $\tilde{x}_t$  is the distance of the X-coordinate position between current vehicle state  $\mathbf{x}_t$  and terminal state  $\mathbf{x}_g$  in centerline reference coordinate  $\mathcal{W}_{\text{ref}}$ .  $y_t$ ,  $\psi_t$ , and  $v_t$  are the Y-coordinate position, heading angle, and speed of  $\mathbf{x}_t$  in  $\mathcal{W}_{\text{ref}}$ , respectively.

Essentially, we utilize a practical and lightweight perception method which makes use of a  $n$ -line 2D Lidar to perceive the surrounding traffic environments around the ego vehicle. Furthermore, our perception method is independent of external sensor fusion and data processing approaches. Moreover, we apply  $z$ -score normalization to whitening the observation features for stable and efficient training.

### B. Action

At each decision step  $t$ , given the agent's observation  $\mathbf{o}_t$ , the policy network outputs continuous action  $\mathbf{a}_t$  as the decision vector  $\mathbf{z}_t$ , which is denoted as:

$$\mathbf{z}_t = \mathbf{a}_t = \pi(\mathbf{o}_t). \quad (14)$$

The action space of RL is set to be continuous, whose lower and upper bounds are given in Table II. It is noted that  $x_{\text{ref},t}$  is considered as the deviation of the current X-coordinate position, i.e.,  $\tilde{x}_{\text{ref},t} \leftarrow x_{\text{ref},t} + x_t$ , and  $\mathbf{Q}_{\text{ref},t}$  takes the proportion form of  $\mathbf{Q}_x$ , i.e.,  $\tilde{\mathbf{Q}}_{\text{ref},t} \leftarrow \mathbf{Q}_{\text{ref},t} \odot \mathbf{Q}_x$ , where  $\odot$  is the Hadamard product for element-wise multiplication. With the generated step actions  $\mathbf{a}_t$  as the decision vector  $\mathbf{z}_t$ , the corresponding MPC is modulated to generate a sequence of optimal state trajectories  $\boldsymbol{\xi}^*(\mathbf{z}_t) = \{\mathbf{x}_k^*\}_{k=0}^N$  and control command  $\boldsymbol{\zeta}^*(\mathbf{z}_t) = \{\mathbf{u}_k^*\}_{k=0}^N$ .

Then, the first tuple  $\mathbf{u}_0^*$  in  $\{\mathbf{u}_k^*\}_{k=0}^N$  is converted to control signals of throttle  $u_{\text{th},t}$ , brake  $u_{\text{br},t}$ , and steering angle  $u_{\text{st},t}$  by a command converter  $f_c$  to transit the vehicle state, as:

$$(u_{\text{th},t}, u_{\text{br},t}, u_{\text{st},t}) = f_c(\mathbf{u}_t), \quad (15)$$

---

**Algorithm 1:** Online Motion Synthesis with MPC Through Policy Inference

---

**Input:**  $f_{\text{MPC}}, \mathbf{x}_g$   
**Output:**  $u_{\text{th},t}, u_{\text{br},t}, u_{\text{st},t}$

- 1 Sample  $\mathbf{o}_t$ ;
- 2 Normalize  $\mathbf{o}_t$ ;
- 3  $\mathbf{z}_t = \mathbf{a}_t = \pi^*(\mathbf{o}_t)$ ;
- 4  $J_{\text{ref},k} = \|\mathbf{x}_k - \mathbf{x}_{\text{ref}}\|_{\mathbf{Q}_{\text{ref}}}^2$ ;
- 5 Solve (7) online to get  $\xi^*(\mathbf{z}_t)$  and  $\zeta^*(\mathbf{z}_t)$ ;
- 6  $(u_{\text{th},t}, u_{\text{br},t}, u_{\text{st},t}) = f_c(\mathbf{u}_t)$ ;

---

where

$$(u_{\text{th},t}, u_{\text{br},t}) = \begin{cases} (\min(a_t/3, 1), 0), & a_t \geq 0 \\ (0, \max(-a_t/8, -1)), & a_t < 0 \end{cases},$$

$$u_{\text{st},t} = \text{clip}(\delta_t, -1, 1).$$

Note that  $\text{clip}(\cdot)$  is a clip function to prevent values from exceeding the prescribed threshold  $[-1, 1]$ .

The motion synthesis procedures are executed according to Algorithm 1.

### C. Reward

To encourage safe and efficient self-driving behavior, after taking each decision  $\mathbf{z}_t = \mathbf{a}_t$ , the RL agent receives a reward signal  $r_t$  with the reward function:

$$r_t(\mathbf{a}_t | \mathbf{o}_t) = r_{\text{forward}} + r_{\text{speed}} + r_{\text{coll}} + r_{\text{road}} + r_{\text{steer}} + r_{\text{time}}. \quad (16)$$

More details of the reward are shown in Table III, where  $y_{\text{bound}}$  is the Y-coordinate position of the left or right road boundary in  $\mathcal{W}_{\text{ref}}$ .

Table III  
REWARD FUNCTION

$r_{\text{forward}}$	Step forward distance	$\bar{x}_{t-1} - \bar{x}_t$
$r_{\text{speed}}$	Average speed if reaching destination	$\bar{v}$
$r_{\text{coll}}$	Penalty if collision occurs	-100
$r_{\text{road}}$	Deviation distance if out of road	$- y_t - y_{\text{bound}} $
$r_{\text{steer}}$	Steering cost	$- \delta $
$r_{\text{time}}$	Punishment if out of time	-100

### D. Policy Training

The corresponding transition  $(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1})$  including observations, actions, and rewards are stored into the fixed-size first-in, first-out replay buffer  $\mathcal{D}$  for offline training. We utilize the off-policy SAC algorithm [17], [18] to learn the optimal policy  $\pi^*$  that maximizes the return together with its entropy:

$$\pi^* = \operatorname{argmax}_{\pi} \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t)} \left[ \sum_{t=0}^T \gamma^t (r_t + \alpha \mathcal{H}(\pi(\cdot | \mathbf{o}_t))) \right], \quad (17)$$

where  $\gamma$  is the discount factor,  $\mathcal{H}$  denotes the entropy, and  $\alpha$  is the temperature parameter that tunes the importance of the entropy term versus the return.

---

**Algorithm 2:** Policy Training with Off-Policy RL

---

**Input:**  $f_{\text{MPC}}$   
**Output:**  $\pi_{\theta}^*$

- 1 Initialize  $\theta$  and empty  $\mathcal{D}$ ;
- 2 Reset the environment to get  $\mathbf{x}_g$  and  $\mathbf{o}_t$ ;
- 3 **while not terminated do**
- 4     Normalize  $\mathbf{o}_t$ ;
- 5     **if random exploration then**
- 6         Randomly sample  $\mathbf{z}_t = \mathbf{a}_t = \text{random}()$ ;
- 7     **else**
- 8          $\mathbf{z}_t = \mathbf{a}_t = \pi_{\theta}(\mathbf{o}_t)$ ;
- 9     **end**
- 10    Solve (7) online to obtain  $\xi^*(\mathbf{z}_t)$  and  $\zeta^*(\mathbf{z}_t)$ ;
- 11     $(u_{\text{th},t}, u_{\text{br},t}, u_{\text{st},t}) = f_c(\mathbf{u}_t)$ ;
- 12    Sample  $r_t, \mathbf{o}_{t+1}$ ;
- 13    Store  $(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1})$  in  $\mathcal{D}$ ;
- 14    **if update policy then**
- 15         Update  $\pi_{\theta}$  with SAC;
- 16 **end**

---

The soft state value of SAC is calculated as follows:

$$V(\mathbf{o}_t) = \mathbb{E}_{\mathbf{a}_t} [Q(\mathbf{o}_t, \mathbf{a}_t)] - \alpha \log(\pi(\mathbf{a}_t | \mathbf{o}_t)), \quad (18)$$

where  $Q(\mathbf{o}_t, \mathbf{a}_t)$  is soft state-action value function. A critic network is trained to approximate  $Q_{\phi}(\mathbf{o}_t, \mathbf{a}_t)$ , where  $\phi$  denotes the parameters of the critic network, and the critic loss is computed as:

$$J_Q(\phi) = \mathbb{E}_{\mathbf{o}_t} \left[ \frac{1}{2} (Q_{\phi}(\mathbf{o}_t, \mathbf{a}_t) - (r_t + \gamma \mathbb{E}_{\mathbf{o}_{t+1}} [V(\mathbf{o}_{t+1})]))^2 \right]. \quad (19)$$

Furthermore, the policy loss is obtained as:

$$J_{\pi}(\theta) = \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t)} [\alpha \log(\pi_{\theta}(\mathbf{a}_t | \mathbf{o}_t)) - Q_{\phi}(\mathbf{o}_t, \mathbf{a}_t)], \quad (20)$$

where  $\theta$  is the set of parameters of the policy network. Note that the temperature parameter  $\alpha$  is auto-tuned during training and the temperature loss is counted as:

$$J(\alpha) = \mathbb{E}_{\mathbf{a}_t} [-\alpha (\log \pi_t(\mathbf{a}_t | \mathbf{o}_t) - \alpha \bar{\mathcal{H}})], \quad (21)$$

where  $\bar{\mathcal{H}}$  is the target entropy.

To avoid the critic learning oscillation and subsequent performance deterioration, the conditions of terminating the episode are to be distinguished before entering the replay buffer, i.e., we set the value to 0 if and only if a collision occurs. Moreover, the disparity in magnitude between the reward values can lead to a sudden change in the Q-function, which hampers the learning process. To address this issue, we execute a reward shaping step whenever an outlier reward is given, which significantly improves the training effectiveness. In this work, we take the form of reward shaping as:

$$r_t = \begin{cases} -5, & r_t \leq -5, \\ r_t, & \text{otherwise.} \end{cases} \quad (22)$$

The policy training process is summarized in Algorithm 2.

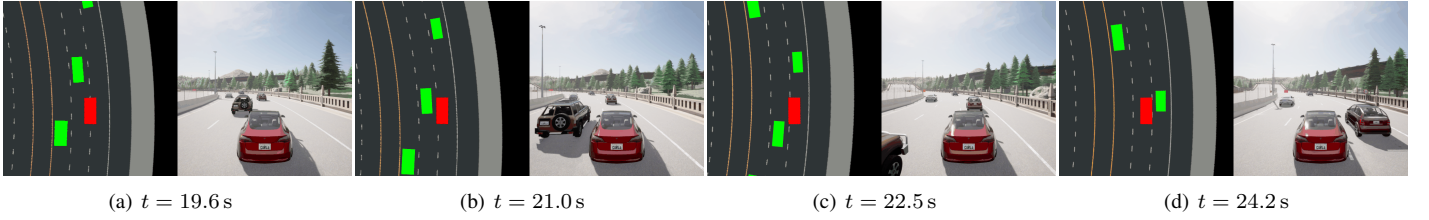


Fig. 3. Key frames of a trail with our framework in lane change and overtaking. The left side of each subfigure is a bird-view image where the red rectangle is the ego vehicle and the green rectangles represent other traffic participants, while the right side of each subfigure is a third-person view attached to the ego vehicle. It is noted that the rectangles on the bird-view images are the inflated bounding box of traffic participants.

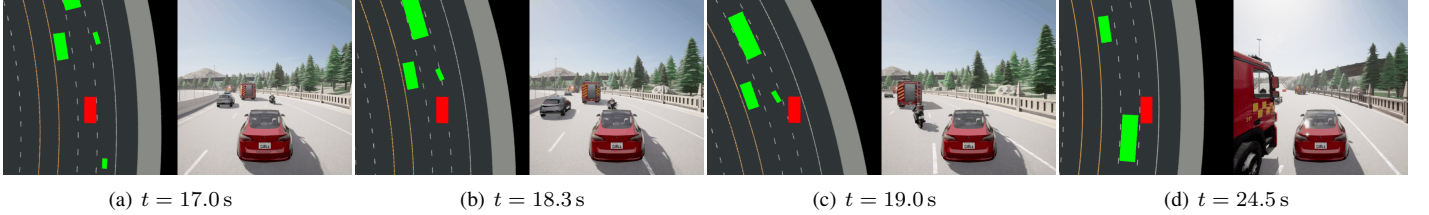


Fig. 4. Key frames of a trail with our framework in complex and dynamic traffic environments.

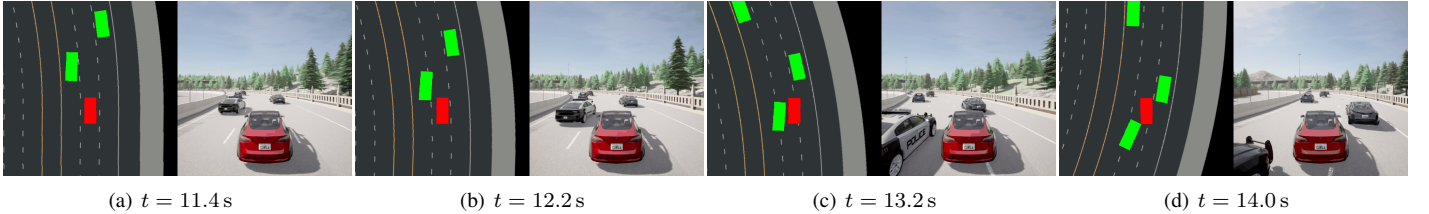


Fig. 5. Key frames of a trail with our framework towards the driving behavior uncertainties of other traffic participants.

## VI. EXPERIMENTS

### A. Implementation Setup

The MPC problem is solved using CasADi [19] with the IPOPT option via the single-shooting method. The weighting matrices  $\mathbf{Q}_x$ ,  $\mathbf{Q}_u$ , and  $\mathbf{Q}_{\Delta_u}$  are set to  $\text{diag}([100, 100, 100, 10])$ ,  $\text{diag}([1, 1])$ , and  $\text{diag}([0.1, 0.1])$ . We take the length of prediction horizon  $N$  as 5.0s and discrete sampling time  $d_t$  as 0.1s. The lower and upper bounds of speed, acceleration, and steering angle are set to  $v_{\min} = 0.0 \text{ m/s}$ ,  $v_{\max} = 10.0 \text{ m/s}$ ,  $a_{\min} = -9.0 \text{ m/s}^2$ ,  $a_{\max} = 4.5 \text{ m/s}^2$ ,  $\delta_{\min} = -0.75 \text{ rad}$ , and  $\delta_{\max} = 0.75 \text{ rad}$ , respectively. We construct the policy and critic networks in PyTorch [20], each with 2 hidden layers with 256 LeakyReLU nodes. The networks are trained with Adam optimizer [21] with a learning rate  $3 \times 10^{-4}$ . The discount factor is set to  $\gamma = 0.99$ . Training starts after the replay buffer collects more than 2500 randomly explored step data.

The long outer ring road with 3 lanes in Town05 of the high-fidelity simulator CARLA [22] is the testbed for SAC model training and performance evaluation<sup>1</sup>. In this traffic environment, we set the Tesla Model 3 as the self-driving vehicle and consider 9 other traffic participants with various types, e.g., vans, cars, motorcyclists, and cyclists. All other traffic participants are non-interactive and in auto-pilot mode.

<sup>1</sup>The supplementary videos for the simulations are accessible at <https://youtu.be/bbw-YqMoilA>.

Here, we take a 73-line 2D Lidar for perception, i.e.,  $n = 73$ , and do not rely on the external module to predict the future trajectories of other participants.

### B. Driving Performance

**Safety:** In our framework, the safety of urban driving can be guaranteed, as the MPC-based maneuvers can generate collision-free trajectories for lane change and overtaking if the high-level policy is well-trained to produce proper decision variables. A trail is shown in Fig. 3 to manifest the effectiveness of our framework for motion synthesis in collision avoidance and safety guarantee for urban self-driving. At  $t = 19.6 \text{ s}$ , the ego vehicle intends to overtake two front vehicles in the middle and right lanes. Intuitively, human drivers are accustomed to overtaking the front vehicles by lane change with enough safe space margin. It is observed that at  $t = 21.0 \text{ s}$ , the ego vehicle starts to enter the right lane to overtake the front vehicle on the middle lane with enough safe space margin; then at  $t = 22.5 \text{ s}$ , the ego vehicle overtakes the first vehicle and attempts to return to the middle lane for preparation of overtaking the next vehicle which is located on the right lane; and finally at  $t = 24.2 \text{ s}$ , the ego vehicle prepares for overtaking the second vehicle with acceptable safe distance. It is equally worth noting that the constraints of road boundaries are strictly obeyed in this trial, even if they are not directly adopted to the optimization problem. Hence, it can

be concluded that the functionality of collision avoidance and safety guarantee is well retained in our framework for motion synthesis.

**Maneuverability:** By latently representing the constraints of collision avoidance and drivable surface boundaries with instantaneous references, our MPC framework leads to high maneuverability in complex and dynamic traffic environments. A trail is displayed to illustrate the superiority in terms of agility as shown in Fig. 4. It is noted that the current traffic is a complex dilemma at  $t = 17.0$  s, as all lanes are occupied with traffic participants in driving. The motorcyclist on the right lane intends to change into the middle lane, leaving the right lane temporarily empty. In this case, the only way to escape from this dilemma is to traverse the temporarily empty lane to enter the open road for high driving efficiency; otherwise, the ego vehicle would sacrifice driving efficiency to follow the front vehicles. It is considered that such a challenging dilemma is a feasible testbed to evaluate the maneuverability of our framework due to its high complexity and dynamicity. At  $t = 19.0$  s, the ego vehicle turns right to occupy the temporarily empty lane and then accelerates to overtake the surrounding vehicles. Finally, the ego vehicle enters the open road and escapes from the traffic dilemma at  $t = 24.5$  s. Therefore, the superiority in maneuverability of our framework for motion synthesis in complex and dynamic environments is clearly demonstrated.

**Robustness:** Unmodeled uncertainties of other traffic participants' behaviors are challenging problems to tackle because they degrade the performance of maneuvers in safety and feasibility guarantees and could even lead to the failure of safe motion synthesis. As illustrated in Fig. 5, the ego vehicle attempts to overtake the two front vehicles by traversing the middle lane at  $t = 11.4$  s. However, we can find that the front vehicle on the left lane intends to enter the middle lane to block the on-taking route of our ego vehicle at  $t = 12.2$  s. Then, the ego vehicle urgently dodges and attempts to bypass the left vehicle with enough safe distance margin at  $t = 13.2$  s. Ultimately, the emergence of the potential collision caused by unpredictable lane-change behavior of the front vehicle is released  $t = 14.0$  s. Hence, our framework manifests the strong robustness to uncertain behaviors of other traffic participants.

### C. Comparison Analysis

We compare our proposed framework with the following representative baseline methods:

- **Vanilla-RL:** The MPC is removed from the proposed framework, where the policy network directly encodes the input features to actions as control commands [3].
- **Hard-MPC:** The hard constraints of collision avoidance with other traffic participants and drivable surface boundaries are implemented according to [1], [2].
- **Soft-MPC:** The aforementioned constraints are further formulated in the soft form, i.e., the penalty terms are added in the cost functions to limit the amount of constraint violation [23].

It is pertinent to note that our framework is independent of an external module for trajectory prediction while the

Table IV  
SUCCESS, COLLISION, TIME-OUT RATE AS WELL AS THE AVERAGE DRIVING SPEED OF DIFFERENT METHODS FOR URBAN DRIVING

Approaches	Succ. (%)	Coll. (%)	Time. (%)	Aver. speed (m/s)
<b>Ours</b>	<b>85</b>	<b>15</b>	<b>0</b>	<b>8.58</b>
Vanilla-RL	74	26	0	8.25
Hard-MPC	64	24	12	5.32
Soft-MPC	75	25	0	5.97

conventional methods (Hard-MPC and Soft-MPC) require the predicted positions of surrounding participants to implement the constraint of collision avoidance. In this work, a naive trajectory prediction method is adopted for the conventional methods, where the participants are assumed to keep the current speed to drive forward along the current heading angle. Besides, full current state observability is assumed for conventional methods. Furthermore, we can also find that conventional methods encounter challenges in our implemented traffic environments due to the high complexity and dynamicity of our trials. Apparently, generating safe and reasonable driving behaviors under these conditions seems to be rather difficult. Hence, the complexity of the traffic environment is reduced by decreasing the number of other traffic participants from 9 to 6. Moreover, to facilitate the constraint formulation of collision avoidance, we uniform all traffic participants to the Tesla Model 3. Furthermore, a 37-line 2D Lidar is adopted for perception.

We run 100 trials of different methods in the simplified traffic environment and record their performance in terms of success, collision and time-out rate, and also the average driving speed as shown in Table IV. Here, an instance is recorded as a success if reaching the destination without collision and within the specified episode length; otherwise, it is recorded as a collision if collided or recorded as a time-out if out of time. In this table, it is observed that our method reaches the highest success rate of 85% and the lowest collision rate of 15% with a time-out rate of 0%, while obtaining the fastest average driving speed of 8.58 m/s compared to the baselines. It is clearly demonstrated that our proposed framework manifests superiority in terms of safety and driving efficiency. The strong safety guarantee can be elaborated as the functionality of collision avoidance is ensured and the robustness to traffic uncertainties is enhanced. The high driving efficiency can be expounded by the improved maneuverability, which is supported by the fact that transforming the aforementioned constraints into the latent form significantly reduces the difficulty in computing an efficiency-friendly solution of MPC under such a complex and dynamic environment.

Vanilla-RL is expected to show high maneuverability and driving efficiency if the reward function is well-designed due to its capability of directly mapping the input features into control commands. Through our trials, it manifests acceptable driving efficiency with an average driving speed of 8.25 m/s, but the safety guarantee is not as strong as ours due to the collision rate of 26%. This indicates that the black-box characteristic of the learned policy hampers the stability of such a strategy for motion synthesis. Furthermore, this is also

a part of our intentions to bridge RL and MPC to improve the stability and safety of learned policy with the applications of urban self-driving.

It is noted from this table that Hard-MPC and Soft-MPC suffer from driving efficiency with the average speed of 5.32 m/s and 5.97 m/s because the solutions to the MPC problem are too conservative for maneuvers to generate agile driving behaviors for high driving efficiency (i.e., overtaking).

Specifically, due to the heavy computational burden of solving the nonlinear, nonconvex, and constrained MPC online, the time-delay of control commands of Hard-MPC seriously hinders the real-time driving performance. Intuitively, unacceptable time-delay can lead to the failure of collision avoidance and violation of the limitation of road boundaries. Hence, the safety of Hard-MPC is worse than expected with a collision rate of 24%. The other disadvantage of Hard-MPC in terms of the time-out rate of 12% can be interpreted by the predicament that occurs when the initial state of MPC is out of the target state set, i.e., the vehicle will be stuck in this predicament once out of the road.

Furthermore, an additional key factor that impairs the performance of Soft-MPC is the oscillation and divergence of solutions when the optimization becomes overly complex to solve. It is inferred that the solution quality of MPC is degraded under the cost settings with the soft form of the aforementioned constraints rather than our proposed latent form. Therefore, it is explanatory that Soft-MPC has a collision rate of 25%.

Based on the above comparative analysis, we can conclude that our proposed method demonstrates superiority in terms of driving efficiency and safety guarantee, even when taking into account the practical and lightweight observability and the absence of any additional trajectory prediction module.

## VII. CONCLUSIONS

In this paper, we propose a novel learning-based online MPC framework to address self-driving tasks in complex and dynamic urban scenarios. By modulating the cost functions with instantaneous references in the form of decision variables, the constraints in terms of collision avoidance and drivable road surface boundaries are transformed into the latent form. Then, the policy search for real-time generation of desired references is formulated as a DRL problem, where the step actions are cast as the decision variables. Through a series of experiments in a high-fidelity simulator, our framework is shown to manifest improved maneuverability and enhanced robustness to traffic uncertainties, and the results also demonstrate the superiority of the proposed method in terms of driving efficiency and safety guarantee over other baselines. Our future work is to further enhance the scalability and generalizability of our method when deploying to other complex and dynamic self-driving scenarios. Hardware experimental validation is also part of our future interests.

## REFERENCES

- [1] F. Eiras, M. Hawasly, S. V. Albrecht, and S. Ramamoorthy, "A two-stage optimization-based motion planner for safe urban driving," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 822–834, 2021.
- [2] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, "Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2994–3008, 2017.
- [3] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Dürri, "Superhuman performance in Gran Turismo Sport using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4257–4264, 2021.
- [4] Y. Song, H. Lin, E. Kaufmann, P. Dürri, and D. Scaramuzza, "Autonomous overtaking in gran turismo sport using curriculum reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 9403–9409.
- [5] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.
- [6] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.
- [7] G. Torrente, E. Kaufmann, P. Föhn, and D. Scaramuzza, "Data-driven mpc for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3769–3776, 2021.
- [8] K. Y. Chee, T. Z. Jiahao, and M. A. Hsieh, "KNODE-MPC: A knowledge-based data-driven predictive control framework for aerial robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2819–2826, 2022.
- [9] T. Salzmann, E. Kaufmann, J. Arrizabalaga, M. Pavone, D. Scaramuzza, and M. Ryll, "Real-time neural MPC: Deep learning model predictive control for quadrotors and agile robotic platforms," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2397–2404, 2023.
- [10] I. Lenz, R. A. Knepper, and A. Saxena, "DeepMPC: Learning deep latent features for model predictive control." in *Robotics: Science and Systems*, vol. 10. Rome, Italy, 2015, p. 25.
- [11] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan online, learn offline: Efficient learning and exploration via model-based control," *arXiv preprint arXiv:1811.01848*, 2018.
- [12] N. Karnchanachari, M. I. Valls, D. Hoeller, and M. Hutter, "Practical reinforcement learning for MPC: Learning from sparse objectives in under an hour on a real robot," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 211–224.
- [13] Y. Song and D. Scaramuzza, "Learning high-level policies for model predictive control," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7629–7636.
- [14] —, "Policy search for model predictive control with application to agile drone flight," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2114–2130, 2022.
- [15] Y. Wang, B. Wang, S. Zhang, H. W. Sia, and L. Zhao, "Learning agile flight maneuvers: Deep se (3) motion planning and control for quadrotors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1680–1686.
- [16] Y. Wang, Y. Li, H. Ghazzai, Y. Massoud, and J. Ma, "Chance-aware lane change with high-level model predictive control through curriculum reinforcement learning," *arXiv preprint arXiv:2303.03723*, 2023.
- [17] 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.
- [18] 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.
- [19] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on Robot Learning*. PMLR, 2017, pp. 1–16.
- [23] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, 2020.