Chance-Aware Lane Change with High-Level Model Predictive Control Through Curriculum Reinforcement Learning

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

***Abstract*—Lane change in dense traffic typically requires the recognition of an appropriate opportunity for maneuvers, which remains a challenging problem in self-driving. In this work, we propose a chance-aware lane-change strategy with high-level model predictive control (MPC) through curriculum reinforcement learning (CRL). In our proposed framework, full-state references and regulatory factors concerning the relative importance of each cost term in the embodied MPC are generated by a neural policy. Furthermore, effective curricula are designed and integrated into an episodic reinforcement learning (RL) framework with policy transfer and enhancement, to improve the convergence speed and ensure a high-quality policy. The proposed framework is deployed and evaluated in numerical simulations of dense and dynamic traffic. It is noteworthy that, given a narrow chance, the proposed approach generates high-quality lane-change maneuvers such that the vehicle merges into the traffic flow with a high success rate of 96%. Finally, our framework is validated in the high-fidelity simulator under dense traffic, demonstrating satisfactory practicality and generalizability.**

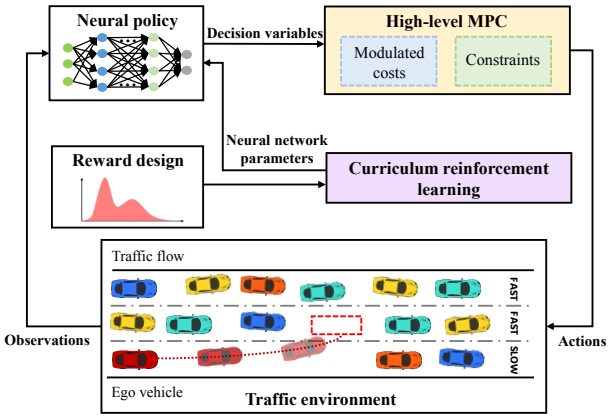

Figure 1. Overview of our proposed framework for chance-aware lane-change problems. The recognized dynamic chance for lane change is visually represented through the use of a red dashed rectangle.

I. Introduction

Chance-aware lane change is geared towards maneuvering the ego vehicle to track and occupy the recognized narrow and moving chance with an appropriate safe margin, such that the ego vehicle merges into the traffic flow. However, it is still an open and challenging problem on how to generate a satisfactory trajectory and feasible lane-change maneuvers. First, explicit prediction in terms of motions and states of traffic flow is demanded to infer the exact pose for the execution of lane change. Also, the time window needs to be identified dynamically for lane-change maneuvers during the planning horizon. Moreover, the inevitable existence of traffic uncertainties poses potential threats to ensure driving safety. To address these challenges, it is imperative to develop a motion planner for chance-aware lane change with strong adaptiveness towards such dense and dynamic traffic.

As one of the widely used optimization-based approaches, model predictive control (MPC) has gained wide popularity for its effectiveness in optimizing the trajectory while dealing with various constraints for self-driving [1]–[3]. However, in complex scenarios such as the aforementioned chance-aware lane-change problem, it is nearly impossible to properly set all the handcrafted factors of MPC, e.g., time-varying constraints corresponding to the dynamic identification of time windows for lane-change maneuvers. In this sense, the solution quality could be significantly degraded. On the other hand, reinforcement learning (RL) has shown promising results in generating effective and agile maneuvers for self-driving by learning a driving strategy through trial-and-error [4]–[7]. However, it is a common problem that pure RL-based methods suffer from instability and limited safety guarantees of control policies.

Learning-based MPC stands as a promising hybrid paradigm for devising control strategies, wherein critical elements of MPC can be acquired through classical machine learning models or deep neural networks. A typical routine for parameterizing the MPC with crucial factors relies on learning the system model. In [8], [9], the Gaussian processes are employed to enhance the system model to improve the solution quality of MPC and alleviate the computational burden. Moreover, the formulation of constraints within the MPC scheme can benefit from deep learning techniques. In [10], [11], recurrent neural networks (RNNs) and generative adversarial networks (GANs) are integrated into an MPC framework for generating lane-change maneuvers in dense traffic. RNNs and GANs are trained to predict the future motions and poses of interactive vehicles, contributing to the establishment of safety constraints within the MPC scheme.

Furthermore, the integration of MPC and RL has been explored for motion planning tasks due to its outstanding flexibility and adaptiveness to challenging problems. As one of the seminal works in this area, a Gaussian distribution is harnessed to model the high-level policy in [12], [13], with which the traversal time is defined as a sequence of decision variables, such that the MPC can be parameterized. Hence, the quadrotor accomplishes the task of flying through a swinging gate. However, the state references of MPC are fixed as the gate states, which could degrade the performance of agile

Yubin Wang, Yulin Li, 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; yline@connect.ust.hk; zpeng940@connect.hkust-gz.edu.cn; jun.ma@ust.hk)

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

Manuscript 3125 submitted to 2024 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2023.

flight. Additionally, in [14], SE(3) decision variables modeled by deep neural networks, are further designed as the references of MPC. Essentially, this approach manifests the effectiveness in traversing a moving and rotating gate. Nevertheless, the weight modulation under the MPC scheme is empirical, which hinders the generalization of the method to dense and dynamic environments with higher complexity.

In this paper, a novel learning-based MPC framework for chance-aware lane-change tasks is proposed, where the augmented decision variables are designed to parameterize the MPC. Specifically, we make use of a neural policy to learn full states as references of MPC and also their regulatory factors which can automatically determine the relative importance of references within the planning horizon. To deal with the reward sparsity issue and further improve the training efficiency, we incorporate curriculum reinforcement learning (CRL) with policy transfer and enhancement to learn the optimal policy progressively with ordered curricula. The contributions of this paper are listed as follows:

(1) We propose a novel learning-based MPC framework that incorporates full-state references and regulatory factors which can modulate the relative importance of each cost term within the cost functions. This facilitates the effective extraction and adjustment of all crucial information for optimizing the solution quality of MPC, leading to improved adaptiveness to dense and dynamic traffic.

(2) To improve the policy quality and avoid unstable learning, we present CRL with policy transfer and enhancement to learn a neural policy, which achieves faster convergence and higher reward compared to other baselines.

(3) The proposed approach is validated through numerical simulations under dense and dynamic traffic, where improved safety and effectiveness are demonstrated through comparative experiments. Furthermore, the practicality and generalizability are illustrated through experimental validations in the high-fidelity simulator.

II. Problem Statement

A. Vehicle Dynamic Model

The bicycle model in [15] is used in this work, where the state vector of the vehicle is defined as $\mathbf{x} = \begin{bmatrix} p_x & p_y & \varphi & v_x & v_y & \omega \end{bmatrix}^\top$, where p_x and p_y denote the X-coordinate and Y-coordinate position of the vehicle's center of mass, φ is the heading angle, v_x and v_y are the longitudinal speed and lateral speed, and ω represents the yaw angular velocity. Also, we integrate the actions into a vector as $\mathbf{u} = \begin{bmatrix} a & \delta \end{bmatrix}^\top$, where a and δ are the acceleration and steering angle. Subsequently, the nonlinear dynamic model f_{dyn} of the vehicle in discrete time is given by:

$$\begin{aligned}\mathbf{x}_{t+1} &= \mathbf{x}_t + f_{\text{dyn}}(\mathbf{x}_t, \mathbf{u}_t) d_t \\ &= \begin{bmatrix} p_{x,t} + (v_{x,t}\cos\varphi_t - v_{y,t}\sin\varphi_t) d_t \\ p_{y,t} + (v_{x,t}\sin\varphi_t + v_{y,t}\cos\varphi_t) d_t \\ \varphi_t + \omega_t d_t \\ v_{x,t} + a_t d_t \\ \frac{L_k\omega_t d_t - k_f\delta_t v_{x,t} d_t - m v_{x,t}^2 \omega_t d_t}{m v_{x,t} - (k_f + k_r) d_t} \\ \frac{I_z v_{x,t}\omega_t + L_k v_{y,t} d_t - l_f k_f \delta_t v_{x,t} d_t}{I_z v_{x,t} - (l_f^2 k_f + l_r^2 k_r) d_t} \end{bmatrix},\end{aligned} \tag{1}$$

where t represents the current time, d_t denotes the step time, m is the mass of the vehicle, l_f and l_r are the distance from the center of mass to the front and rear axle, k_f and k_r are the cornering stiffness of the front and rear wheels, $L_k = l_f k_f - l_r k_r$, and I_z is the polar moment of inertia.

B. Chance-Aware Lane-Change Task

In this work, the focus lies on the spatial-temporal characteristics of the entirety of traffic flow, as opposed to individual surrounding vehicles. Consequently, the driving behaviors of each vehicle within the traffic flow are unrestricted, encompassing motions such as accelerating, braking, and steering. Notably, to preserve the underlying topology of the modeled traffic flow, lane-change behaviors of vehicles within the traffic flow are prohibited. To facilitate the modeling of traffic flow, we assume that the traffic flow maintains consistent speed with inherent uncertainties. Additionally, the ego vehicle is obstructed by a dead-end, where the front vehicles drive slowly. The dynamic chance for the ego vehicle to perform lane change is recognized as the gap of the traffic flow on the middle lane. The illustration of the traffic is depicted in Fig. 1.

In this context, the objective of the chance-aware lane-change mission is to plan the optimal state trajectory $\mathbf{x}_k^*, \forall k \in [0, 1, \cdots, N]$ towards the goal states $\mathbf{x}_g$ and obtain a sequence of optimal control commands $\mathbf{u}_k^*, \forall k \in [0, 1, \cdots, N-1]$ over a receding horizon N, such that the ego vehicle merges into the traffic flow successfully on the target lane. Concurrently, the lane change necessitates the adoption of appropriate maneuvers by the ego vehicle within a specified temporal window, which occurs precisely before any potential collision with the front vehicles.

III. High-Level Model Predictive Control with Augmented Decision Variables

A. MPC Formulation

To solve the chance-aware lane-change problem, we formulate a nonlinear MPC with augmented decision variables over the receding horizon N at time step t:

$$\begin{aligned}\min_{\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}} \; & J_{x_N} + \sum_{k=0}^{N-1} \left(J_{x_k} + J_{u_k} + J_{\Delta u_k} + J_{\text{tra},k}\right) \\ & = \boldsymbol{\delta}_{x,N}^T \mathbf{Q}_x \boldsymbol{\delta}_{x,N} + \sum_{k=0}^{N-1} \left(\boldsymbol{\delta}_{x,k}^T \mathbf{Q}_x \boldsymbol{\delta}_{x,k} + \boldsymbol{\delta}_{u,k}^T \mathbf{Q}_u \boldsymbol{\delta}_{u,k}\right. \\ & \left. + \Delta_{u,k}^T \mathbf{Q}_{\Delta_u} \Delta_{u,k} + \boldsymbol{\delta}_{\text{tra},k}^T \mathbf{Q}_{\text{tra}}\left(t_{\text{tra}}, k\right) \boldsymbol{\delta}_{\text{tra},k}\right) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{x}_k + f_{\text{dyn}}\left(\mathbf{x}_k, \mathbf{u}_k\right) d_t, \\ & \mathbf{x}_0 = \mathbf{x}_{\text{init}}\,, \mathbf{u}_{-1} = \mathbf{u}_{\text{init}}\,, \\ & p_{y,\min} \le p_y \le p_{y,\max}, \\ & a_{\min} \le a \le a_{\max}, \\ & -\delta_{\max} \le \delta \le \delta_{\max},\end{aligned} \tag{2}$$

where $\boldsymbol{\delta}_{x,k} = (\mathbf{x}_k - \mathbf{x}_g)$ denotes the difference between the current states $\mathbf{x}_k$ and the goal states $\mathbf{x}_g$, $\boldsymbol{\delta}_{\text{tra},k} = (\mathbf{x}_k - \mathbf{x}_{\text{tra}})$ represents the difference between the current states $\mathbf{x}_k$ and the

learnable full-state references $\mathbf{x}_{\text{tra}}$, $\boldsymbol{\delta}_{u,k} = \mathbf{u}_k$ is a regularization term for control commands $\mathbf{u}_k$, $\boldsymbol{\Delta}_{u,k} = (\mathbf{u}_k - \mathbf{u}_{k-1})$ is another regularization term considering the variation of control commands for driving comfort. The quadratic cost terms in (2) are weighted by diagonal matrices $\mathbf{Q}_x$, $\mathbf{Q}_{\text{tra}}(t_{\text{tra}}, k)$, $\mathbf{Q}_u$, and $\mathbf{Q}_{\Delta_u}$. The initial states and initial control commands are represented by $\mathbf{x}_{\text{init}}$ and $\mathbf{u}_{\text{init}}$, respectively. The Y-coordinate position is bounded within $p_{y,\min}$ and $p_{y,\max}$ introduced by the road width. Additionally, control commands are constrained by $a_{\min}$, $a_{\max}$, $-\delta_{\max}$, and $\delta_{\max}$ considering the physical limits of vehicle dynamics.

B. MPC Parameterization with Augmented Decision Variables

We introduce the learnable full-state references $\mathbf{x}_{\text{tra}}$ as the desired pose and speed of a lane change manuver:

$$\mathbf{x}_{\text{tra}} = [p_{x,\text{tra}}, p_{y,\text{tra}}, \varphi_{\text{tra}}, v_{x,\text{tra}}, v_{y,\text{tra}}, \omega_{\text{tra}}]^\top, \tag{3}$$

where $\mathbf{x}_{\text{tra}}$ are intermediate states for temporary tracking of MPC. To endow the vehicle with the ability to automatically balance the importance between tracking $\mathbf{x}_{\text{tra}}$ and tracking $\mathbf{x}_g$, the weighting matrix $\mathbf{Q}_{\text{tra}}(t_{\text{tra}}, k)$ with adaptive adjustment is defined as:

$$\mathbf{Q}_{\text{tra}}(t_{\text{tra}}, k) = \mathbf{Q}_{\max} \exp\left(-\gamma \left(k d_t - t_{\text{tra}}\right)^2\right), \tag{4}$$

where $\mathbf{Q}_{\max}$ is a learnable maximum of $\mathbf{Q}_{\text{tra}}$, $\gamma \in \mathbb{R}_+$ is the exponential decay rate for costs in terms of tracking $\mathbf{x}_{\text{tra}}$, and t_{tra} is a learnable tracking time reference, which determines the opportune timing for a lane change. In order to further modulate the relative importance of each state reference respectively, $\mathbf{Q}_{\max}$ is defined as:

$$\begin{aligned}\mathbf{Q}_{\max} = \operatorname{diag}(&Q_{p_{x,\text{tra}}}, Q_{p_{y,\text{tra}}}, Q_{\varphi_{\text{tra}}}, Q_{v_{x,\text{tra}}}, \\ &Q_{v_{y,\text{tra}}}, Q_{\omega_{\text{tra}}}).\end{aligned} \tag{5}$$

Specifically, $Q_{p_{x,\text{tra}}}$ and $Q_{p_{y,\text{tra}}}$ in $\mathbf{Q}_{\max}$ are assigned relatively large values whereas $\mathbf{x}_{\text{tra}}$ and t_{tra} tend to confine themselves to small values. The considerable discrepancy in magnitude could potentially hinder the training process. Therefore, $\mathbf{Q}_{\max}$ adopts the proportion form of $\mathbf{Q}_x$, i.e., $\tilde{\mathbf{Q}}_{\max} \leftarrow \mathbf{Q}_{\max} \odot \mathbf{Q}_x$, where $\odot$ is the Hadamard product for element-wise multiplication.

Therefore, the regulatory factors $\mathbf{Q}_{\max}$ and t_{tra} modulate the cost functions of MPC collaboratively, which balance the importance between tracking $\mathbf{x}_{\text{tra}}$ and tracking $\mathbf{x}_g$. Therefore, the lane change behavior is divided into two distinct phases. During the initial phase, the ego vehicle prepares for the maneuvers of the lane-change behavior by tracking $\mathbf{x}_{\text{tra}}$; subsequently, in the latter phase, the ego vehicle concentrates on tracking $\mathbf{x}_g$ to effectively merge into the traffic flow.

We integrate all decision variables to an augmented decision vector $\mathbf{z}$ as:

$$\begin{aligned}\mathbf{z} = [&p_{x,\text{tra}}, p_{y,\text{tra}}, \varphi_{\text{tra}}, v_{x,\text{tra}}, v_{y,\text{tra}}, \omega_{\text{tra}}, Q_{p_{x,\text{tra}}}, \\ &Q_{p_{y,\text{tra}}}, Q_{\varphi_{\text{tra}}}, Q_{v_{x,\text{tra}}}, Q_{v_{y,\text{tra}}}, Q_{\omega_{\text{tra}}}, t_{\text{tra}}]^\top \in \mathbb{R}^{13}.\end{aligned} \tag{6}$$

In this sense, the MPC is parameterized by decision vector $\mathbf{z}$. By feeding MPC with different $\mathbf{z}$, different corresponding optimal state trajectories are generated, denoted as $\boldsymbol{\xi}^*(\mathbf{z}) = f_{\text{MPC}}(\mathbf{z})$ (i.e., $\boldsymbol{\xi}^*(\mathbf{z}) = \{\mathbf{x}_k^*(\mathbf{z})\}_{k=0}^N$). f_{MPC} is defined as the mapping function of MPC. Hence, we can incorporate the episodic RL technique for policy search [12]–[14] to determine the optimal policy π^* that automatically tune the augmented decision variables.

IV. Curriculum Reinforcement Learning with Policy Transfer and Enhancement

A. Observation and Policy Representation

The observation vector of the ego vehicle is defined as follows:

$$\begin{aligned}\mathbf{o} = [&p_{x,\text{init}}, p_{y,\text{init}}, \varphi_{\text{init}}, v_{x,\text{init}}, p^{\text{c}}_{x,\text{init}}, p^{\text{c}}_{y,\text{init}}, v^{\text{c}}_{x,\text{init}}, \\ &p^{\text{f}}_{x,\text{init}}, p^{\text{f}}_{y,\text{init}}, v^{\text{f}}_{x,\text{init}}]^\top \in \mathbb{R}^{10},\end{aligned} \tag{7}$$

where $p_{x,\text{init}}$, $p_{y,\text{init}}$, $v_{x,\text{init}}$, $p^{\text{c}}_{x,\text{init}}$, $p^{\text{c}}_{y,\text{init}}$, $v^{\text{c}}_{x,\text{init}}$, $p^{\text{f}}_{x,\text{init}}$, $p^{\text{f}}_{y,\text{init}}$, and $v^{\text{f}}_{x,\text{init}}$ represent the initial X-coordinate and Y-coordinate position as well as the longitudinal speed of the ego vehicle, dynamic chance and the nearest front vehicle, respectively. φ_{init} is the initial heading angle of the ego vehicle.

We further exploit a deep neural network to parameterize the policy π, with which the augmented decision vector $\mathbf{z}$ is modeled as:

$$\mathbf{z} = \pi(\mathbf{o}) = f_{\boldsymbol{\theta}}(\mathbf{o}), \tag{8}$$

where $\boldsymbol{\theta}$ are the parameters of the deep neural network, $\mathbf{o}$ is the observation vector of the vehicle. Moreover, we apply z-score normalization to input features in $\mathbf{o}$. In this work, we present a novel CRL framework to determine the optimal policy π^*.

B. Multi-Task Reward Formulation

Sparse Lane-Change Reward: The sparse reward function, which evaluates the quality of the optimal state trajectory $\boldsymbol{\xi}^*(\mathbf{z})$ for the lane-change task, is designed as:

$$R_{\text{LC}}(\boldsymbol{\xi}^*(\mathbf{z})) = R_{\max} - c_c \sum_{k=0}^{N} \rho_c \left|\boldsymbol{v}_k\right|^2, \tag{9}$$

where ρ_c is a binary flag indicating whether a collision with surrounding vehicles occurs, $c_c \in \mathbb{R}_+$ is a hyperparameter for weighting the collision penalty, and $R_{\max} \in \mathbb{R}_+$ is the goal reward, which is gained when the ego vehicle merges into the traffic flow successfully.

Reward Shaping: The sparsity of the lane-change reward poses challenges to learning a viable policy within a reasonable time frame. Therefore, we introduce a novel reward term to directly evaluate the decision variables, which guides the RL agent to explore the policy space in directions of larger policy gradients:

$$\begin{aligned}&R_{\text{DV}}(\mathbf{z}) = \\ &- c_x \left|\Delta p_x\right| - c_y \rho_y \Delta p_y - c_t \left|t\right| - c_{\Delta t} \rho_{\Delta t} \Delta t - c_{\Delta\varphi} \rho_{\Delta\varphi} \Delta\varphi \\ &- c_{p_x} \rho_{p_x} \left|Q_{p_{x,\text{tra}}}\right| - c_{p_y} \rho_{p_y} \left|Q_{p_{y,\text{tra}}}\right| - c_\varphi \rho_\varphi \left|Q_{\varphi_{\text{tra}}}\right| \\ &- c_{v_x} \rho_{v_x} \left|Q_{v_{x,\text{tra}}}\right| - c_{v_y} \rho_{v_y} \left|Q_{v_{y,\text{tra}}}\right| - c_\omega \rho_\omega \left|Q_{\omega_{\text{tra}}}\right|,\end{aligned} \tag{10}$$

Algorithm 1: Curriculum Reinforcement Learning with Policy Transfer and Enhancement

Input: f_{MPC}, $\mathscr{C}$
Output: $\pi^* = f_{\boldsymbol{\theta}^*}$

Initialize $\boldsymbol{\theta}$;
while *not terminated* **do**
  Select Curriculum C_i from Curricula $\mathscr{C}$;
  Reset the environment to get $\mathbf{o}$ and $\mathbf{x}_g$ according to C_i;
  if *curriculum switched* **then**
    Load policy π^* trained with C_{i-1};
  end
  Compute $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{o})$;
  Solve MPC($\mathbf{z}$) as (2) online to obtain $\boldsymbol{\xi}^*(\mathbf{z})$;
  for $j \leftarrow 1$ **to** $dim(\mathbf{z})$ **do**
    Perturb j-th row in $\mathbf{z}$;
    Estimate g_j using (13);
  end
  Update $\boldsymbol{\theta}$ using gradient ascent with $\boldsymbol{g_{s1}}$ and $\boldsymbol{g_{s2}}$;
end

where

$$\begin{aligned}
\Delta p_x &= p_{x,\mathrm{tra}} - p^{\mathrm{c}}_{x,\mathrm{init}},\\
\Delta p_y &= \min(|p_{y,\mathrm{tra}} - p_{y,\max}|, |p_{y,\mathrm{tra}} - p_{y,\min}|),\\
\Delta t &= \min(|t_{\mathrm{tra}} - t_{\max}|, |t_{\mathrm{tra}} - t_{\min}|),\\
\Delta \varphi &= \min(|\varphi_{\mathrm{tra}} - \varphi_{\max}|, |\varphi_{\mathrm{tra}} - \varphi_{\min}|).
\end{aligned}$$

In Eq. (10), t denotes the current time within the simulation episode, c_x, c_y, c_t, $c_{\Delta t}$, $c_{\Delta\varphi}$, c_{p_x}, c_{p_y}, c_{φ}, c_{v_x}, c_{v_y}, and c_{ω} are weighting coefficients, ρ_y, $\rho_{\Delta t}$, and $\rho_{\Delta\varphi}$ are binary flags indicating whether the learnable position reference of Y-coordinate is within the range of values $[p_{y,\min}, p_{y,\max}]$, whether the reference of the learnable time exceeds the simulation duration or falls below zero, and whether the learnable heading angle exceeds the range of the feasible heading angle, respectively. ρ_{p_x}, ρ_{p_y}, ρ_{φ}, ρ_{v_x}, ρ_{v_y}, ρ_{ω} are also binary flags implying whether the learnable maximum of weighting is smaller than zero. $t_{\min}$ and $t_{\max}$ represent the lower bound and upper bound of the duration of simulation. $\varphi_{\min}$ and $\varphi_{\max}$ are the lower and upper bound of the predefined feasible heading angle range.

C. Curriculum Reinforcement Learning with Policy Transfer and Enhancement

We operate the neural policy at the beginning of each episode to model $\mathbf{z}$, with which the MPC can generate a sequence of optimal trajectories $\boldsymbol{\xi}^*(\mathbf{z})$. Therefore, the corresponding reward signal $R(\boldsymbol{\xi}^*(\mathbf{z}))$ is received to evaluate the quality of generated trajectories. Hence, inspired by [14], the problem of finding the optimal neural policy with RL is reformulated to the following reward maximization problem:

$$\begin{aligned}
\max_{\boldsymbol{\theta}} \quad & R\left(\boldsymbol{\xi}^*\left(\mathbf{z}\left(\boldsymbol{\theta}\right)\right)\right)\\
\text{s.t.} \quad & \boldsymbol{\xi}^*\left(\mathbf{z}\left(\boldsymbol{\theta}\right)\right) = f_{\mathrm{MPC}}\left(\mathbf{z}\left(\boldsymbol{\theta}\right)\right).
\end{aligned} \tag{11}$$

The gradient of the episode reward R with respect to neural network parameters $\boldsymbol{\theta}$ is decomposed with chain rule, where

$$\frac{dR}{d\boldsymbol{\theta}} = \frac{\partial R}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \boldsymbol{\theta}}. \tag{12}$$

Specifically, the sub-gradient $\boldsymbol{g_{s2}} = \partial \mathbf{z}/\partial \boldsymbol{\theta}$ is calculated automatically by loss backward when updating the neural network. However, obtaining another sub-gradient $\boldsymbol{g_{s1}} = \partial R/\partial \mathbf{z}$ is computationally expensive, as it requires differentiation through the nonlinear MPC optimization problem over the whole receding horizon.

We denote the value in the j-th row in $\boldsymbol{g_{s1}}$ as g_j. With the finite difference policy gradient method, we can estimate g_j as:

$$g_j = \frac{R\left(\boldsymbol{\xi}^*\left(\mathbf{z} + \epsilon \boldsymbol{e_j}\right)\right) - R\left(\boldsymbol{\xi}^*\left(\mathbf{z}\right)\right)}{\epsilon}, \tag{13}$$

where $\boldsymbol{e_j} = [0, \cdots, 0, 1, 0, \cdots, 0]^\top$ is a unit vector with only 1 in j-th row, ϵ is a small random step in the direction $\boldsymbol{e_j}$ to perturb the augmented decision vector $\mathbf{z}$.

We exploit on-policy RL to train the neural network policy through the maximization of the designed reward. Nevertheless, the reward in such hierarchical framework does not exhibit a direct correspondence with the neural network output. That is to say, unfavorable decision variables can still yield an acceptable reward due to the inherent robustness of MPC. Furthermore, due to the sparsity of lane-change reward, the sufficiency of policy exploration is not guaranteed, so it is challenging to yield an acceptable policy in finite time. Consequently, the training of neural network suffers from inefficient and insufficient exploration in the policy space and even unstable learning.

Curriculum learning (CL) is a well-established routine to accelerate the exploration when RL is handling rather complex missions, especially addressing extrapolation error and premature policy convergence [7]. Therefore, we present a CRL framework to train our neural policy with policy transfer and policy enhancement. We generate three modes of curricula, which are represented by $\mathscr{C} = \{C_i\}, i \in \{1, 2, 3\}$. The curricula are designed with different tasks in different domains.

Curriculum 1: Transferable Policy Learning with Reward Shaping in Static Environment. In source domain 1 which is denoted as $\mathcal{M}_{s,1}$, the traffic flow on the fast lane keeps static. The objective of Curriculum 1 is to learn a transferable neural policy. We train our randomly initialized neural network through the maximization of the reward term directly evaluating the decision variables as (10). In $\mathcal{M}_{s,1}$, the task $\mathcal{T}_{s,1}$ is to train an acceptable neural policy $\pi_{s,1}$ in finite time, with which the learnable state references $\mathbf{x}_{\mathrm{tra}}$, maximum of weighting matrix $\mathbf{Q}_{\max}$, and tracking time reference t_{tra} are expected to converge to a feasible range roughly. Therefore, the RL agent is guided by the empirically-designed reward, increasing the exploration efficiency in the policy space.

Curriculum 2: Lane-Change Policy Learning Under Low-Speed Setting. In Curriculum 2, we load the transferable policy $\pi_{s,1}$ and train it in source domain $\mathcal{M}_{s,2}$, where the traffic flow on the fast lane move at a speed lower than the normal setting. The objective of Curriculum 2 is to generalize the

transferable $\pi_{s,1}$ to a lane-change policy $\pi_{s,2}$ by maximizing the lane-change reward as (9).

Curriculum 3: Lane-Change Policy Enhancement Under Normal-Speed Setting. In Curriculum 3, we aim to obtain an optimal neural policy π^* for the lane-change task under a normal speed setting. In the target domain $\mathcal{M}_t$, the traffic flow on the fast lane moves at a normal speed and the weight of the penalty term in terms of collisions in lane-change reward is increased. We load and enhance the lane-change policy $\pi_{s,2}$, and obtain the optimal policy π^* eventually by maximizing the revised lane-change reward.

To this end, the proposed CRL framework with policy transfer and enhancement is summarized in Algorithm 1.

V. Experiments

A. Numerical Setup

The MPC problem is solved using CasADi [16] with IPOPT. In MPC, we set the receding horizon and the step time to $T = 5.0\,\mathrm{s}$ and $d_t = 0.1\,\mathrm{s}$. Furthermore, we take the weighting matrices $\mathbf{Q}_x$, $\mathbf{Q}_u$, and $\mathbf{Q}_{\Delta_u}$ to $\mathrm{diag}\left([100, 100, 100, 10]\right)$, $\mathrm{diag}\left([1, 1]\right)$, and $\mathrm{diag}\left([0.1, 0.1]\right)$, respectively. Also, we set the lower and upper bounds of acceleration and steering angle to $a_{\min} = -6.0\,\mathrm{m/s^2}$, $a_{\max} = 3\,\mathrm{m/s^2}$, $\delta_{\min} = -0.6\,\mathrm{rad}$, and $\delta_{\max} = 0.6\,\mathrm{rad}$, respectively.

The deep neural network is constructed in PyTorch [17], with the structure of 4 hidden layers with 128 LeakyReLU nodes. The neural network is trained by Adam optimizer [18] with an initial learning rate 3×10^{-4}, where the learning rate decays in 0.96 every 32 steps. Weights and Biases [19] is utilized to monitor the training process.

We train the policy network and evaluate the driving performance in numerical simulations, where the ego vehicle is placed at $[p_{x,\mathrm{init}} \sim \mathcal{N}(30, 2.5), -2.5]$ and the dynamic chance moves from $\left[p^{\mathrm{c}}_{x,\mathrm{init}} \sim \mathcal{N}(50, 10), 2.5\right]$ at a time-variant speed of $v^{\mathrm{c}}_x \sim \mathcal{N}(\mu_i, \sigma_i)$ m/s. Here, μ_i and σ_i are the mean speed and speed standard deviation of each curriculum from $i \in \{1, 2, 3\}$. In our settings, $\mu_1, \sigma_1, \mu_2, \sigma_2, \mu_3, \sigma_3$ are $0, 0, 2, 0.5, 4, 1$, respectively. The goal states $\mathbf{x}_g$ are set to $[p^{\mathrm{c}}_x, 2.5, 0, v^{\mathrm{c}}_x, 0, 0]$, where p^{c}_x is the X-coordinate position of the dynamic chance obtained from traffic in real time.

B. Training Result

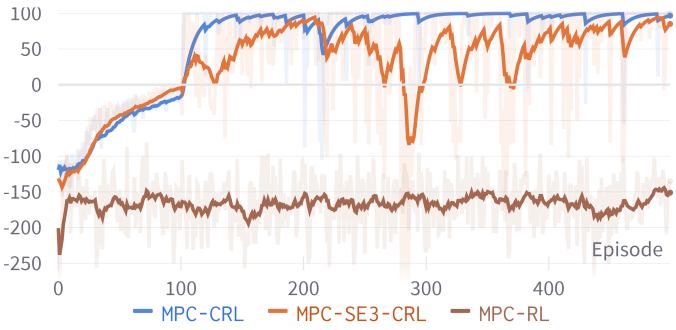

Figure 2. Reward curves of different methods. The training curves are smoothed by exponential moving average with a degree of 0.8, and the curriculum is switched at episodes 100 and 200.

In order to illustrate the effectiveness of the proposed method (denoted as MPC-CRL) in policy learning, we employ the subsequent learning-based baselines for comparison:

- High-level MPC with augmented decision variables and vanilla RL (denoted as MPC-RL).
- High-level MPC with SE(3) decision variables [14] and CRL (denoted as MPC-SE3-CRL).

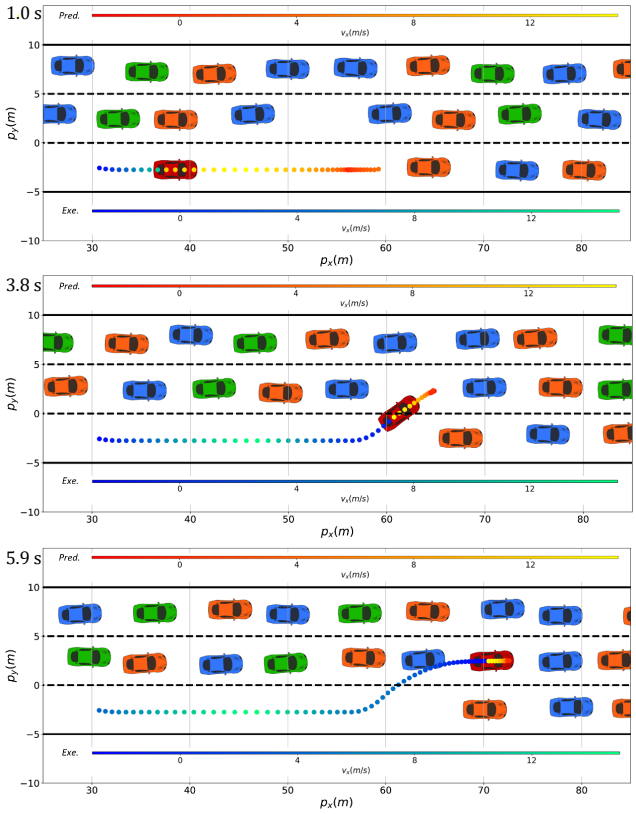

Figure 3. Key frames of a trail with our proposed approach for chance-aware lane change in numerical simulations. The vehicles on the middle and upper lane represent the traffic flow, the vehicle in red is the ego vehicle, the vehicles ahead of the ego vehicle on the lower lane are front vehicles, the dotted line in red and blue represent the future trajectory and the executed trajectory of the ego vehicle. The colorbars refer to the different longitudinal speed of predicted and executed trajectories of the ego vehicle.

The reward curves are presented in Fig. 2. The results exhibit that our proposed MPC-CRL surpasses MPC-RL both in terms of convergence speed and reward performance. Additionally, the incorporation of augmented decision variables, which adaptively modulate the costs of MPC, endows MPC-CRL with superior reward performance when compared to MPC-SE3-CRL. Therefore, the training results clearly indicate that the presented CRL framework effectively encourages the RL agent to efficiently and sufficiently explore the policy space, ultimately leading to the attainment of the satisfactory optimum. Additionally, the introduction of augmented decision variables notably improves the lane-change performance.

C. Performance Evaluation

To provide further insight into the driving performance attained by our framework, a set of trails with various settings are conducted for driving performance evaluation, where a

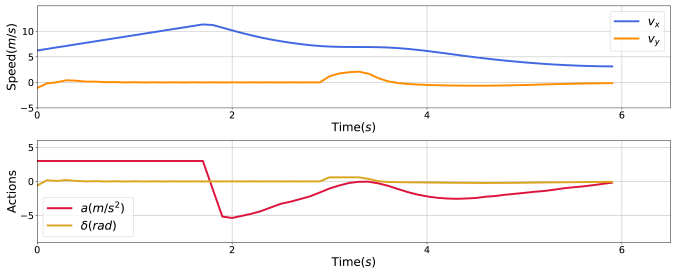

Figure 4. The speed and action profiles of a trail of chance-aware lane change.

series of snapshots from a trail are presented in Fig. 3, and the corresponding speed and action profiles are illustrated in Fig. 4. As shown in Fig. 3(a), when $t = 1.0\,\mathrm{s}$, the ego vehicle plans a long trajectory along the lower lane, intending to track the learnable state references $\mathbf{x}_{\mathrm{tra}}$. Simultaneously, the ego vehicle accelerates and exploits the maximum bounded acceleration, as indicated in Fig. 4. Then, Fig. 3(b) visualizes the ego vehicle's subsequent actions, wherein it decelerates and executes a left turn, preparing of a transition to the middle lane with enough safe margin at $t = 3.8\,\mathrm{s}$. The recorded longitudinal speed at this moment is approximately $6.1\,\mathrm{m/s}$. Ultimately, as depicted in Fig. 3(c), the ego vehicle successfully occupies the recognized dynamic chance and safely merges into the traffic flow on the middle lane at $t = 5.9\,\mathrm{s}$.

D. Comparison Analysis

Due to the failure of acquiring essential domain knowledge during training, MPC-RL does not qualify as a suitable baseline for the comparison of driving performance. We further exploit a new baseline, which adopts the paradigm of high-level MPC with augmented decision variables curated through human-expert experience (denoted as MPC-HE). Moreover, to quantitively analyze the results attained by different methods, we define a collision-free episode finished within finite time as a successful case. Otherwise, we record an episode involving any collision as a collided case and an episode terminated by the episode duration of simulation as a time-out case. Afterwards, we run the trails repeatedly 100 times and record the corresponding success, collision and time-out rate of various methods, as documented in Table I. The results indicate that our proposed approach outperforms all baselines in terms of both task accomplishment and safety assurance due to the highest success rate of 96% and the lowest collision rate of 4%. By leveraging augmented decision variables to automatically modulate the costs of MPC, the adaptiveness of our approach to dense and dynamic traffic is improved significantly. Hence, we conclude that our proposed framework manifests a stronger guarantee of collision avoidance and task success.

E. Validation in High-Fidelity Simulator

We further validate the effectiveness of our proposed framework in the high-fidelity simulator CARLA [20]. The outer ring road with three lanes in Town05 is the testbed for performance validation, where all traffic participants are set

Table I
SUCCESS, COLLISION, AND TIME-OUT RATE OF DIFFERENT METHODS FOR CHANCE-AWARE LANE-CHANGE TASKS.

Approaches	Succ. (%)	Coll. (%)	Time-out (%)
MPC-CRL	**96**	**4**	**0**
MPC-SE3-CRL	77	23	0
MPC-HE	69	28	3

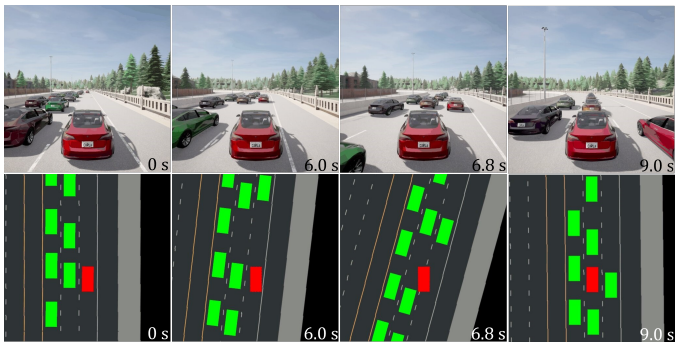

Figure 5. Key frames of the experimental validation of our method in high-fidelity simulator. The top shows the third-person view attached to the ego vehicle. The bottom shows the bird-eye view, where the red rectangle is the ego vehicle while the green rectangles denote the surrounding vehicles.

to Tesla Model 3. In order to reduce the gap between the environments in numerical simulations and high-fidelity validations, the global Cartesian coordinate frame is transformed to the coordinate frame of centerline reference [3]. We load the policy model trained in numerical simulations and fine-tune it. Key frames of a representative example are presented in Fig. 5. The results demonstrate clearly that the RL agent trained by our approach retains the capability to effectively determine suitable maneuvers and appropriate timing for a lane change in CARLA. The validation results highlight the efficacy and generalizability of our method when deployed to the high-fidelity simulator.

VI. CONCLUSIONS

In this paper, we proposed a novel learning-based MPC framework with appropriate parameterization using augmented decision variables. Instead of choosing partial variables (such as only the positions) as references, we utilized a neural policy to learn full-state references and regulatory factors corresponding to their relative importance. Hence, the cost terms were automatically modulated with our specifically-designed augmented decision variables. Furthermore, ordered multi-phase curricula were generated for learning a neural policy using RL, which leads to faster convergence speed and better policy quality. Furthermore, through a series of comparative experiments, our approach demonstrated superiority in terms of success rate in chance-aware lane-change tasks under dense and dynamic traffic settings. Moreover, the practicality and generalizability of our method are further illustrated through the experimental validations in the high-fidelity simulator. Our future work is to reformulate the RL to a step-based paradigm and develop a more efficient training pipeline with analytical gradients. Hardware validation is also part of our future interests.

Manuscript 3125 submitted to 2024 IEEE International Conference on Robotics and Automation (ICRA). Received September 15, 2023.

REFERENCES

[1] M. Mukai, H. Natori, and M. Fujita, "Model predictive control with a mixed integer programming for merging path generation on motor way," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2017, pp. 2214–2219.

[2] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis, and S. Fallah, "Trajectory planning for autonomous high-speed overtaking in structured environments using robust MPC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2310–2323, 2019.

[3] 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.

[4] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[5] I. Nishitani, H. Yang, R. Guo, S. Keshavamurthy, and K. Oguchi, "Deep merging: Vehicle merging controller based on deep reinforcement learning with embedding network," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 216–221.

[6] F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Dürr, "Super-human performance in gran turismo sport using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4257–4264, 2021.

[7] Y. Song, H. Lin, E. Kaufmann, P. Dürr, 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.

[8] 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.

[9] 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.

[10] S. Bae, D. Saxena, A. Nakhaei, C. Choi, K. Fujimura, and S. Moura, "Cooperation-aware lane change maneuver in dense traffic based on model predictive control with recurrent neural network," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1209–1216.

[11] S. Bae, D. Isele, A. Nakhaei, P. Xu, A. M. Añon, C. Choi, K. Fujimura, and S. Moura, "Lane-change in dense traffic with model predictive control and neural networks," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 2, pp. 646–659, 2022.

[12] 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.

[13] ——, "Policy search for model predictive control with application to agile drone flight," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2114–2130, 2022.

[14] 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.

[15] Q. Ge, Q. Sun, S. E. Li, S. Zheng, W. Wu, and X. Chen, "Numerically stable dynamic bicycle model for discrete-time control," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 128–134.

[16] 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.

[17] 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.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[20] 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.