Decentralized Robust Data-driven Predictive Control for Smoothing Mixed Traffic Flow

Xu Shang, Jiawei Wang, and Yang Zheng

***Abstract*—In a mixed traffic with connected automated vehicles (CAVs) and human-driven vehicles (HDVs) coexisting, data-driven predictive control of CAVs promises system-wide traffic performance improvements. Yet, most existing approaches focus on a centralized setup, which is not computationally scalable while failing to protect data privacy. The robustness against unknown disturbances has not been well addressed either, causing safety concerns. In this paper, we propose a decentralized robust `DeeP-LCC` (Data-EnablEd Predictive Leading Cruise Control) approach for CAVs to smooth mixed traffic flow. In particular, each CAV computes its control input based on locally available data from its involved subsystem. Meanwhile, the interaction between neighboring subsystems is modeled as a bounded disturbance, for which appropriate estimation methods are proposed. Then, we formulate a robust optimization problem and present its tractable computational solutions. Compared with the centralized formulation, our method greatly reduces computation burden with better safety performance, while naturally preserving data privacy. Extensive traffic simulations validate its wave-dampening ability, safety performance, and computational benefits.**

I. Introduction

UNDESIRABLE traffic waves, such as phantom traffic jams [1], significantly reduce travel efficiency, fuel economy and driving safety. The emergence of connected automated vehicles (CAVs) promises to greatly mitigate this issue [2]. One typical technology is Cooperative Adaptive Cruise Control (CACC), which groups a series of CAVs into a platoon and applies cooperative control strategies to mitigate undesired traffic oscillations [3], [4]. However, such technologies require a fully CAV environment while the near future will see a transition phase of mixed traffic, where human-driven vehicles (HDVs) coexist with CAVs [5]–[7]. To enhance their societal benefits, CAVs need to be cooperative with other HDVs that still take the majority in mixed traffic.

A. CAV Control in Mixed Traffic

Mixed traffic systems are complex human-in-the-loop cyber-physical systems. Existing research on CAV control in mixed traffic can be divided into model-based and model-free categories, depending on how to address human driving behaviors. Model-based methods typically rely on classical car-following models for HDVs, such as the optimal velocity model (OVM) [8] or the intelligent driver model (IDM) [9]. Lumping the dynamics of HDVs and CAVs, one can derive a parametric model of the entire mixed traffic system [6], allowing for common model-based synthesis approaches, such as optimal control [10], [11], $\mathcal{H}_\infty$ control [12], [13] and model predictive control (MPC) [14], [15]. However, it is non-trivial to accurately identify HDVs' car-following behavior, due to the nonlinearity and stochastic nature of human driving behaviors. On the other hand, model-free methods for CAV control, which bypass system identification and directly construct controllers from data, have received increasing attention. For example, reinforcement learning [16], [17] and adaptive dynamic programming [18], [19] have shown their potential in learning wave-dampening strategies for CAVs. Nonetheless, these model-free methods suffer from several drawbacks, *e.g.*, the lack of interpretability, heavy computation burden for offline training, and difficulties of providing safety guarantees.

Recently, a class of data-driven predictive control, which combines learning methods with the well-established MPC, has shown promising results for constrained control with safety performance [20]–[22]. Notably, a recent work in [22] extends the Data-EnableEd Predictive Control (`DeePC`) technique [23] into Leading Cruise Control [24] (LCC) for mixed traffic, leading to a new notion of `DeeP-LCC`. Compared with traditional frameworks such as CACC and Connected Cruise Control [25], LCC not only enables the CAVs to adaptively follow the HDVs ahead, but also explores CAVs' potential in actively leading the motion of the HDVs behind. Accordingly, LCC is applicable to general cases of mixed traffic with randomly distributed CAVs [7], and enables system-wide improvement for global traffic flow [6], [24]. Note that `DeeP-LCC` relies on Willems' Fundamental Lemma [26] for data-driven behavior representation of a mixed traffic system. This method requires no prior knowledge of the HDVs' car-following behaviors, and explicitly incorporates input/output constraints (*e.g.*, control saturation and driving safety) into receding horizon control. Both numerical simulations [22] and real-world experiments [27] have validated the capability of `DeeP-LCC` in achieving optimal and safe control for CAVs.

B. Decentralization and Robustness

Despite the advantages of data-driven methods in addressing unknown human driving behaviors, the aforementioned studies are still non-trivial in practical traffic systems due to their common centralized setup: a central unit is required to collect trajectory data from all vehicles and to calculate control inputs for all CAVs in a mixed traffic flow (see [27] for implementation details). This setup might be suitable for a small number

The work of X. Shang and Y. Zheng is supported by NSF ECCS-2154650 and NSF CMMI-2320697.

X. Shang and Y. Zheng are with the Department of Electrical and Computer Engineering, University of California San Diego, CA 92093, USA. (x3shang@ucsd.edu; zhengy@ucsd.edu),

J. Wang is with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109, USA. (jiawe@umich.edu).

of vehicles with one or two CAVs, but is not scalable for practical traffic considering real-time computation, data privacy, and communication delays [2], [28]. Moreover, the formation pattern, captured by the spatial distribution pattern of CAVs, is not fixed in real traffic due to the free joining and leaving maneuvers of surrounding vehicles [7]. When any formation changes, the centralized setup requires data recollection and controller redesign for the entire system. To address this problem, some very recent research has designed distributed data-driven predictive control strategies via distributed optimization techniques [20], [29]. Particularly, the authors in [29] solve the original centralized `DeeP-LCC` in a distributed manner via operator-splitting strategies. However, these distributed algorithms require multiple iterations to obtain the control inputs, requiring high-frequency data exchange between CAVs in a small time period which may be often beyond current vehicle-to-vehicle communication capabilities [30].

In addition to scalability, robustness against external disturbances also plays a critical role in enhancing the CAVs' potential in practical traffic. Existing research mostly focuses on maximizing the mitigation ratio with respect to external disturbances to improve CAVs' wave-dampening capability. Typical tools include string stability analysis [31], [32] or optimal and robust control [11], [33]. However, these studies have not explicitly addressed safety performance against unknown disturbances. Very recently, some safety-related techniques like control barrier function [34] or tube-based MPC [14] have been proposed, but they require prior knowledge of HDVs' car-following dynamics. In the centralized `DeeP-LCC` [22], a simple constant assumption is made for external disturbances, which is often inconsistent with human driving behaviors. Rear-end collisions could still occur due to the mismatch of future external disturbances for `DeeP-LCC` [22]. Robust data-driven control has been discussed in `DeePC` for applications in power grids [35], but it is not directly applicable to mixed traffic control because of different system dynamics.

C. Contributions

In this paper, we develop a decentralized robust data-driven predictive control approach to improve the scalability, robustness, and data privacy of CAV control in mixed traffic (particularly in the spirit of `DeeP-LCC` [22]). As shown in Fig. 1, we first decentralize the problem by decomposing the entire mixed traffic system into multiple CF-LCC (Car-Following Leading Cruise Control) subsystems. Each CAV utilizes the measurable data from its own subsystem to compute safe and optimal control inputs. We form a decentralized robust optimization problem for online predictive control, in which different disturbance estimation methods are incorporated. We finally provide an efficient computational method to solve the robust optimization problem. Some preliminary results are reported in [36]. Our contributions of this paper are as follows:

- We propose a decentralized `DeeP-LCC` formulation for CAVs in mixed traffic with locally measured trajectory data of each CF-LCC subsystem. Compared with centralized `DeeP-LCC` [22], it significantly reduces the computational time with a smaller amount of required data. Also, unlike existing distributed optimization approaches [20], [29], our formulation bypasses frequent inter-vehicle communications between neighboring subsystems and naturally contributes to better data privacy.
- We robustify the decentralized `DeeP-LCC` with appropriate disturbance estimation methods to handle the coupling constraints between CF-LCC subsystems. The disturbance bound is estimated based on the system dynamics of the mixed traffic system. Integrating a time-varying disturbance estimation with the aforementioned robustification, our method provides better safety guarantees for each CAV while maintaining comparable wave-damping performance with respect to centralized `DeeP-LCC` [22].
- Finally, we provide an efficient computational method to solve the decentralized robust `DeeP-LCC` problem. The robustification and safety requirements of the mixed traffic system significantly increase the computational complexity of online predictive control. For this aspect, we compare and analyze two computational methods, which are motivated and adapted from robust optimization literature [37], [38]. We have further developed an efficient automatic transformation that reformulates each decentralized robust `DeeP-LCC` problem into its standard conic form, reducing the modeling time compared to general tools, such as YALMIP [39].

Numerical experiments validate the superior performance of our decentralized robust `DeeP-LCC` in terms of both computational efficiency and safety performance. In particular, its computational time is $0.058\,\mathrm{s}$, which is **85.1%** less than the centralized `DeeP-LCC`, facilitating its real-time computation. Moreover, in our experiments, it achieves **0%** violation rate for safety constraints when using both small ($T=700$) and large ($T=1500$) data sets, while these numbers are **99%** and **88%** respectively, for the normal `DeeP-LCC` without robustification [22].

D. Paper Structure

The remainder of this paper is structured as follows. Section II reviews the basics of centralized `DeeP-LCC`. The formulation of decentralized robust `DeeP-LCC` is introduced in Section III, and the disturbance estimation methods are discussed in Section IV. Section V analyzes different computational approaches. Section VI illustrates our numerical traffic simulations. Finally, Section VII concludes this paper.

Notations: We denote $\mathbb{N}$ as the set of all natural numbers. We use $\mathbb{0}$ and I as a zero matrix and an identity matrix with a compatible size. For a series of vectors $a_1, a_2, \ldots, a_n$ and matrices $A_1, A_2, \ldots, A_n$ with the same column size, we denote $\mathrm{col}(a_1, a_2, \ldots, a_m) = [a_1^\mathsf{T}, a_2^\mathsf{T}, \ldots, a_n^\mathsf{T}]^\mathsf{T}$ and $\mathrm{col}(A_1, A_2, \ldots, A_m) = [A_1^\mathsf{T}, A_2^\mathsf{T}, \ldots, A_n^\mathsf{T}]^\mathsf{T}$. For a vector a and a square matrix X, We also denote $||a||_X^2 = a^\mathsf{T} X a$, and $\mathrm{diag}(D_1, D_2, \ldots, D_n)$ denotes a block-diagonal matrix with diagonal blocks $D_1, D_2, \ldots, D_n$. The floor function $\lfloor \alpha \rfloor$ gives the largest integer that is less than α. Finally, we denote the Kronecker product between matrices A and B as $A \otimes B$.

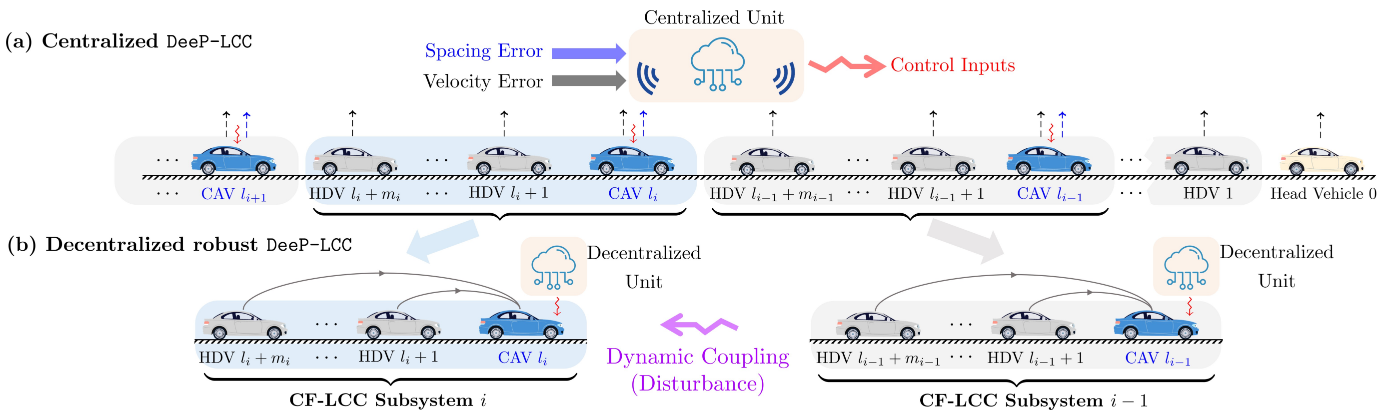


Fig. 1. Schematic of centralized and decentralized robsut DeeP-LCC for CAVs in mixed traffic. (a) Centralized DeeP-LCC. It collects the data of the whole mixed traffic system, including velocity errors of all vehicles (black dashed arrow) and spacing errors of CAVs collectively (blue dashed arrow), and utilizes these data to design control input of CAVs (red squiggle arrow). (b) Decentralized robust DeeP-LCC. It decomposes the mixed traffic system into multiple CF-LCC subsystems and only requires locally available data. The dynamic coupling between subsystems is modeled as the disturbance (purple squiggle arrow).

II. Review of Centralized DeeP-LCC and Problem Statement

In this section, we briefly review the modeling process for mixed traffic system consisting of multiple CAVs and HDVs and the centralized DeeP-LCC [22]. In the end, we give a precise problem statement for this work.

A. Input/Output of Mixed Traffic System

As shown in Fig. 1, we consider a general single-lane mixed traffic system with one head vehicle, indexed as 0, and n following vehicles, indexed as $1, \ldots, n$ from front to end. Among the following vehicles, there exist q CAVs, $n-q$ HDVs and the i-th CAV is indexed as l_i ($1 \leq l_1 < l_2 < \cdots < l_q \leq n$). The number of the HDVs between CAVs i and $i+1$ is represented as m_i, *i.e.*, $m_i = l_{i+1} - l_i$. We denote the index set Ω for all the following vehicles, $\mathcal{S}$ for all the CAVs, $\mathcal{F}_i$ for the HDVs between CAV i and CAV $i+1$, and $\mathcal{F}$ for all the HDVs:

$$\Omega = \{1, 2, \ldots, n\}, \quad \mathcal{S} = \{l_1, l_2, \ldots, l_q\},$$
$$\mathcal{F}_i = \{l_i, l_i + 1, \ldots, l_i + m_i\}, \quad \mathcal{F} = \mathcal{F}_1 \cup \ldots \mathcal{F}_n.$$

The position, velocity, and acceleration of vehicle i at time t are denoted as $p_i(t)$, $v_i(t)$ and $a_i(t)$, $i \in \Omega$, respectively. The spacing between vehicle i and its preceding vehicle $i-1$ is denoted as $s_i(t) = p_{i-1}(t) - p_i(t)$, and their relative velocity is $\dot{s}_i(t) = v_{i-1}(t) - v_i(t)$. At an equilibrium state, all vehicles move at an identical velocity v^* and maintain the corresponding equilibrium space s_i^*, which may vary from different vehicles. We consider the error state of each vehicle with respect to its equilibrium state to facilitate system analysis and controller design. In particular, the velocity error and spacing error for each vehicle are defined as:

$$\tilde{v}_i(t) = v_i(t) - v^*, \quad \tilde{s}_i(t) = s_i(t) - s_i^*, \quad i \in \Omega.$$

We then utilize the aggregation of the velocity and spacing errors of the following vehicles to represent the global state $x \in \mathbb{R}^{2n}$ of the mixed traffic system, given by

$$x(t) = \left[\tilde{s}_1(t), \tilde{v}_1(t), \tilde{s}_2(t), \tilde{v}_2(t), \ldots, \tilde{s}_n(t), \tilde{v}_n(t)\right]^{\mathsf{T}}. \tag{1}$$

The state (1) is not fully measurable. Indeed, due to the unknown car-following behaviors of HDVs, it is impractical to obtain the equilibrium spacing $s_i^*, i \in \mathcal{F}$ for HDVs. The measurable output of the traffic system consists of velocity errors for all vehicles and spacing errors for CAVs only, as we can estimate the equilibrium velocity v^* from the head vehicle and design equilibrium spacing s_i^* for CAVs ($i \in \mathcal{S}$). Accordingly, the output $y(t) \in \mathbb{R}^{n+q}$ is defined as

$$y(t) = \left[\tilde{v}_1(t), \tilde{v}_2(t), \ldots, \tilde{v}_n(t), \tilde{s}_{l_1}(t), \ldots, \tilde{s}_{l_q}(t)\right]^{\mathsf{T}}. \tag{2}$$

As widely used before [6], [11], [25], we assume the acceleration of each CAV can be directly controlled in this study. We denote $u_i(t) = a_i(t) \in \mathbb{R}$ as the control input of each CAV ($i \in \mathcal{S}$). Then, The control input of the entire mixed traffic system $u(t) \in \mathbb{R}^q$ is defined as

$$u(t) = \left[u_{l_1}(t), u_{l_2}(t), \ldots, u_{l_q}(t)\right]^{\mathsf{T}}. \tag{3}$$

Additionally, the head vehicle's velocity error is considered as an external disturbance signal since it cannot be controlled, which is denoted as

$$\epsilon(t) = \tilde{v}_0 = v_0(t) - v^*. \tag{4}$$

Upon defining the state (1), output (2), input (3) and disturbance (4), after linearization and discretization, the parametric model of the mixed traffic system can be derived as

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + H\epsilon(k), \\ \quad y(k) = Cx(k), \end{cases} \tag{5}$$

where k denotes the discrete time step and the specific form of the matrices $A \in \mathbb{R}^{2n \times 2n}$, $B \in \mathbb{R}^{2n \times q}, C \in \mathbb{R}^{(n+q) \times 2n}$, and $H \in \mathbb{R}^{2n}$ can be found in [22, Section II-C]; also see [24].

One goal of CAV control is to design suitable input $u(t)$ in (3) to smooth the mixed traffic flow, represented by $x(t)$ in (1), based on the observation $y(t)$ in (2), in the presence of the external disturbance $\epsilon(t)$. We remark that the parametric model (5), characterized by matrices A, B, C, H, is not available for controller design due to the unknown HDVs' behavior.

To address this, the recently emerging data-driven methods, particularly the centralized `DeeP-LCC` [22], bypass system identification and directly use the input/output trajectories for traffic behavior prediction and CAV control.

B. Basics of Centralized `DeeP-LCC`

The centralized `DeeP-LCC` [22] adapts and extends the standard `DeePC` [23] for CAV control in mixed traffic. It utilizes a pre-collected input/output trajectory to construct a data-driven behavior representation for the mixed traffic system (5). The pre-collected data needs to be rich enough, which is required by a persistent excitation condition.

Definition 1 (Persistently Exciting): The sequence of signal $\omega = \mathrm{col}(\omega(1), \omega(2), \ldots, \omega(T))$ with length T ($T \in \mathbb{N}$) is persistently exciting of order L ($L < T$) if its associated Hankel matrix with depth L has full row rank:

$$\mathcal{H}_L(\omega) = \begin{bmatrix} \omega(1) & \omega(2) & \cdots & \omega(T-L+1) \\ \omega(2) & \omega(3) & \cdots & \omega(T-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ \omega(L) & \omega(L+1) & \cdots & \omega(T) \end{bmatrix}.$$

We start by collecting a length-T input/output data sequence from the mixed traffic system:

$$\begin{aligned} u^{\mathrm{d}} &= \mathrm{col}(u^{\mathrm{d}}(1), u^{\mathrm{d}}(2), \ldots, u^{\mathrm{d}}(T)) \in \mathbb{R}^{qT}, \\ \epsilon^{\mathrm{d}} &= \mathrm{col}(\epsilon^{\mathrm{d}}(1), \epsilon^{\mathrm{d}}(2), \ldots, \epsilon^{\mathrm{d}}(T)) \in \mathbb{R}^{T}, \\ y^{\mathrm{d}} &= \mathrm{col}(y^{\mathrm{d}}(1), y^{\mathrm{d}}(2), \ldots, y^{\mathrm{d}}(T)) \in \mathbb{R}^{(n+q)T}, \end{aligned} \tag{6}$$

and then partition associated Hankel matrices into past data with length T_{ini} and future data with length N

$$\begin{bmatrix} U_{\mathrm{P}} \\ U_{\mathrm{F}} \end{bmatrix} := \mathcal{H}_L(u^{\mathrm{d}}), \ \begin{bmatrix} E_{\mathrm{P}} \\ E_{\mathrm{F}} \end{bmatrix} := \mathcal{H}_L(\epsilon^{\mathrm{d}}), \ \begin{bmatrix} Y_{\mathrm{P}} \\ Y_{\mathrm{F}} \end{bmatrix} := \mathcal{H}_L(y^{\mathrm{d}}), \tag{7}$$

where $L = T_{\mathrm{ini}} + N$, and U_{P} and U_{F} consist of the first T_{ini} rows and the last N rows of $\mathcal{H}_L(u^{\mathrm{d}})$, respectively (similarly for E_{P} and E_{F}, Y_{P} and Y_{F}). We note that the mixed traffic system (5) is controllable [24] and we have the following result.

Proposition 1 ([22, Proposition 2]): Let the most recent past input trajectory with length T_{ini} and the future input trajectory with length N as $u_{\mathrm{ini}} = \mathrm{col}(u(t - T_{\mathrm{ini}}), u(t - T_{\mathrm{ini}} + 1), \ldots, u(t - 1))$ and $u = \mathrm{col}(u(t), u(t+1), \ldots, u(t+N-1))$, respectively (similarly for $\epsilon_{\mathrm{ini}}, \epsilon, y_{\mathrm{ini}}, y$). If the offline data $\bar{u}^{\mathrm{d}} := \mathrm{col}(u^{\mathrm{d}}, \epsilon^{\mathrm{d}})$ is persistently exciting of order $L + 2n$, then the online data $\mathrm{col}(u_{\mathrm{ini}}, \epsilon_{\mathrm{ini}}, y_{\mathrm{ini}}, u, \epsilon, y)$ is a valid length-L trajectory of (5) if and only if there exists $g \in \mathbb{R}^{T - T_{\mathrm{ini}} - N + 1}$ such that

$$\begin{bmatrix} U_{\mathrm{P}} \\ E_{\mathrm{P}} \\ Y_{\mathrm{P}} \\ U_{\mathrm{F}} \\ E_{\mathrm{F}} \\ Y_{\mathrm{F}} \end{bmatrix} g = \begin{bmatrix} u_{\mathrm{ini}} \\ \epsilon_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u \\ \epsilon \\ y \end{bmatrix}. \tag{8}$$

If $T_{\mathrm{ini}} \geq l$, where l denotes the lag of system (5), y is uniquely determined from (8) for each pair of u_{ini}, ϵ_{ini}, y_{ini}, u and ϵ.

The data-driven behavior representation (8) holds for the linear time-invariant mixed traffic system (5). Given the nonlinear and nondeterministic nature of practical mixed traffic, the centralized `DeeP-LCC` [22] further adapts (8) and solves the following optimization problem at each time step t:

$$\min_{g, \sigma_y, u, y} \quad J(y, u, g, \sigma_y) \tag{9a}$$

$$\text{subject to} \quad \begin{bmatrix} U_{\mathrm{P}} \\ E_{\mathrm{P}} \\ Y_{\mathrm{P}} \\ U_{\mathrm{F}} \\ E_{\mathrm{F}} \\ Y_{\mathrm{F}} \end{bmatrix} g = \begin{bmatrix} u_{\mathrm{ini}} \\ \epsilon_{\mathrm{ini}} \\ y_{\mathrm{ini}} \\ u \\ \epsilon \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_y \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{9b}$$

$$\epsilon = \mathbb{0}_N, \quad y \in \mathcal{Y}, \quad u \in \mathcal{U}, \tag{9c}$$

where $(u_{\mathrm{ini}}, \epsilon_{\mathrm{ini}}, y_{\mathrm{ini}})$ is an initial trajectory that needs to be updated at each time step. We will detail the cost function $J(y, u, g, \sigma_y)$, slack variable σ_y, output constraint $\mathcal{Y}$ and input constraint $\mathcal{U}$ in Section III-B (see (14a), (14c), (14d)). One key aspect of `DeeP-LCC` (9) is the data-driven behavior predictor (9b) from Proposition 1, which requires no model information. For the future disturbance ϵ, its value is estimated as $\mathbb{0}_N$ in [22] (see (9c)) based on the assumption that the head vehicle always tries to maintain its equilibrium state.

C. Problem Statement

Despite its promising numerical [22] and experimental performance [27], the original `DeeP-LCC` (9) has two major limits: 1) its centralized control setup and 2) the simplistic zero assumption of the future external disturbance in (9c). In particular, as shown in Fig. 1(a), the centralized `DeeP-LCC` requires global information of the whole traffic system to solve (9). Given a practical mixed traffic system with multiple CAVs and HDVs, the length of pre-collected data (6) might be huge, affecting data privacy, communication burden, and numerical efficiency of solving (9) in real time. Moreover, the zero future velocity error estimation (*i.e.*, $\epsilon = \mathbb{0}_N$) is not consistent with the real-world driving behaviors, and strong oscillations may happen during the occurrence of traffic waves. An inaccurate estimation results in a mismatch between prediction and real traffic behavior, posing safety concerns (rear-end collisions).

In this work, our first objective is to address the centralized and zero disturbance settings, formally stated as follows:

Problem 1 (Decentralization and Robustification): Develop a decentralized and robust formulation of `DeeP-LCC`, which relies on locally available data and incorporates explicit considerations of potential future disturbances.

Our strategy is to decompose the entire mixed traffic system into a series of subsystems, and formulate a robust optimization problem against an uncertainty set of future disturbances, which will be detailed in Section III. Then, we need an appropriate uncertainty estimation for the future disturbance ϵ. This is our second objective, addressed in Section IV.

Problem 2 (Disturbance Estimation and Approximation): Estimate an accurate disturbance set from the historical data of the preceding vehicle's velocity error and develop a low-dimension approximation.

Finally, the third objective is to establish an efficient computation method for the robust data-driven predictive control.

Problem 3 (Efficient Computations): Develop a computationally efficient method to solve the decentralized robust DeeP-LCC online at each time step.

For this objective, we will exploit the problem structure of DeeP-LCC and adapt techniques from robust optimization [37] [38]. Details will be presented in Section V.

III. Decentralized Robust DeeP-LCC

In this section, we present a new decentralized robust DeeP-LCC formulation to control CAVs in mixed traffic. In particular, we first divide the whole traffic system in Fig. 1 into multiple subsystems, called Car-Following LCC (CF-LCC). We then derive a data-driven representation for each CF-LCC subsystem, allowing for its local DeeP-LCC formulation.

A. Input/Output of CF-LCC Subsystem

As shown in Fig. 1, by exploiting the cascading structure, the entire mixed traffic with q CAVs can be naturally divided into q subsystems, each of which represents a small mixed traffic system with one CAV l_i and its following m_i HDVs (denoted by $\mathcal{F}_i$). These subsystems are called CF-LCC in [24], as each CAV leads the motion of the HDVs behind it while following one single vehicle immediately ahead, which is typically HDV l_i-1 (but could also be CAV l_{i-1} if $\mathcal{F}_{i-1}=\varnothing$).

Here, we focus on the dynamics of each CF-LCC subsystem and *view the coupling dynamics between two neighboring CF-LCC subsystems as an external disturbance*. Similar to (1)-(4) for the global mixed traffic system, the state $x_i(t)\in\mathbb{R}^{2(m_i+1)}$ of the i-th CF-LCC is defined as

$$x_i(t)=[\tilde{s}_{l_i}(t),\tilde{v}_{l_i}(t),\tilde{s}_{l_i+1}(t),\tilde{v}_{l_i+1}(t),\cdots,\tilde{s}_{l_i+m_i}(t),\tilde{v}_{l_i+m_i}(t)]^{\mathsf{T}},$$

and the output $y_i(t)\in\mathbb{R}^{m_i+2}$ is defined as

$$y_i(t)=[\tilde{v}_{l_i}(t),\tilde{v}_{l_i+1}(t),\tilde{v}_{l_i+2}(t),\ldots,\tilde{v}_{l_i+m_i}(t),\tilde{s}_{l_i}(t)]^{\mathsf{T}},$$

and the control input $u_i(t)$ is the control signal of the CAV, which is $u_{l_i}(t)\in\mathbb{R}$. In addition, the velocity error of the preceding vehicle is considered as an external disturbance:

$$\epsilon_i(t)=\tilde{v}_{l_i-1}=v_{l_i-1}-v^*\in\mathbb{R}. \tag{10}$$

Similar to (5), the linearized discrete-time state-space model of the i-th CF-LCC subsystem can be written as

$$\begin{cases} x_i(k+1)=A_ix_i(k)+B_iu_i(k)+H_i\epsilon_i(k),\\ \quad y_i(k)=C_ix_i(k), \end{cases} \tag{11}$$

where A_i,B_i,C_i and H_i can be found in [24].

B. Data-driven Representation and Local DeeP-LCC

The CF-LCC model (11) (*i.e.*, the knowledge of A_i,B_i,C_i and H_i) is unknown in practice. We here derive a data-driven representation for the dynamics of each CF-LCC subsystem, allowing for the construction of the local DeeP-LCC.

Offline Data Collection: We collect an input/output trajectory of length T_i for the i-th subsystem:

$$\begin{aligned} u_i^{\mathrm{d}}&=\mathrm{col}(u_i^{\mathrm{d}}(1),u_i^{\mathrm{d}}(2),\ldots,u_i^{\mathrm{d}}(T_i))\in\mathbb{R}^{T_i},\\ \epsilon_i^{\mathrm{d}}&=\mathrm{col}(\epsilon_i^{\mathrm{d}}(1),\epsilon_i^{\mathrm{d}}(2),\ldots,\epsilon_i^{\mathrm{d}}(T_i))\in\mathbb{R}^{T_i},\\ y_i^{\mathrm{d}}&=\mathrm{col}(y_i^{\mathrm{d}}(1),y_i^{\mathrm{d}}(2),\ldots,y_i^{\mathrm{d}}(T_i))\in\mathbb{R}^{(m_i+2)T_i}. \end{aligned}$$

These collected data are then used to construct the Hankel matrix of order L, which is partitioned as follows:

$$\begin{bmatrix}U_{i,\mathrm{P}}\\U_{i,\mathrm{F}}\end{bmatrix}:=\mathcal{H}_L(u_i^{\mathrm{d}}),\ \begin{bmatrix}E_{i,\mathrm{P}}\\E_{i,\mathrm{F}}\end{bmatrix}:=\mathcal{H}_L(\epsilon_i^{\mathrm{d}}),\ \begin{bmatrix}Y_{i,\mathrm{P}}\\Y_{i,\mathrm{F}}\end{bmatrix}:=\mathcal{H}_L(y_i^{\mathrm{d}}), \tag{12}$$

where $U_{i,\mathrm{P}}$ and $U_{i,\mathrm{F}}$ consist of the first T_{ini} rows and the last N rows of $\mathcal{H}_L(u_i^{\mathrm{d}})$, respectively (similarly for $E_{i,\mathrm{P}}$ and $E_{i,\mathrm{F}}$, $Y_{i,\mathrm{P}}$ and $Y_{i,\mathrm{F}}$).

Online Behavior Representation: We will utilize the Hankel matrix (12) for online behavior predictions of CF-LCC. We have the following result.

Proposition 2 ([29, Lemma 2]): At time step k, denote the most recent past input sequence $u_{i,\mathrm{ini}}$ with length T_{ini} and the future input sequence u_i with length N as

$$\begin{aligned} u_{i,\mathrm{ini}}&=\mathrm{col}(u_i(k-T_{\mathrm{ini}}),u_i(k-T_{\mathrm{ini}}+1),\ldots,u_i(k-1)),\\ u_i&=\mathrm{col}(u_i(k),u_i(k+1),\ldots,u_i(k+N-1)), \end{aligned}$$

and $\epsilon_{i,\mathrm{ini}}$, ϵ, y_{ini} and y are denoted similarly. If the data sequence $\bar{u}_i^{\mathrm{d}}=\mathrm{col}(u_i^{\mathrm{d}},\epsilon_i^{\mathrm{d}})$ is persistently exciting of order $L+2(m_i+1)$ (where $L=T_{\mathrm{ini}}+N$), then the sequence $\mathrm{col}(u_{i,\mathrm{ini}},\epsilon_{i,\mathrm{ini}},y_{i,\mathrm{ini}},u_i,\epsilon_i,y_i)$ is a valid trajectory with length L of the CF-LCC subsystem (11), if and only if there exists a vector $g_i\in\mathbb{R}^{T_i-L+1}$ such that

$$\begin{bmatrix}U_{i,\mathrm{P}}\\E_{i,\mathrm{P}}\\Y_{i,\mathrm{P}}\\U_{i,\mathrm{F}}\\E_{i,\mathrm{F}}\\Y_{i,\mathrm{F}}\end{bmatrix}g_i=\begin{bmatrix}u_{i,\mathrm{ini}}\\\epsilon_{i,\mathrm{ini}}\\y_{i,\mathrm{ini}}\\u_i\\\epsilon_i\\y_i\end{bmatrix}. \tag{13}$$

If T_{ini} is no smaller than the lag of system (11), then the future output y_i is unique for fixed $(u_{i,\mathrm{ini}},\epsilon_{i,\mathrm{ini}},y_{i,\mathrm{ini}},u_i,\epsilon_i)$.

This proposition establishes a data-driven representation of the dynamics of the CF-LCC subsystem (11), *i.e.*, all of its valid trajectories can be expressed as a linear combination of the pre-collected trajectories, which are encoded in the Hankel matrices (12). Accordingly, one can directly utilize the local pre-collected trajectories $(u_i^{\mathrm{d}},\epsilon_i^{\mathrm{d}},y_i^{\mathrm{d}})$ to predict the future output y_i given the input u_i, the disturbance ϵ_i, and the initial condition $(u_{i,\mathrm{ini}},\epsilon_{i,\mathrm{ini}},y_{i,\mathrm{ini}})$.

Using local data representation (13), we formulate the local DeeP-LCC for the i-th CF-LCC at each time step k as

$$\min_{g_i,\sigma_{y_i},u_i,\epsilon_i,y_i}\quad V_i(u_i,y_i)+\lambda_{g_i}||g_i||_2^2+\lambda_{y_i}||\sigma_{y_i}||_2^2 \tag{14a}$$

$$\text{subject to}\quad \begin{bmatrix}U_{i,\mathrm{P}}\\E_{i,\mathrm{P}}\\Y_{i,\mathrm{P}}\\U_{i,\mathrm{F}}\\E_{i,\mathrm{F}}\\Y_{i,\mathrm{F}}\end{bmatrix}g_i=\begin{bmatrix}u_{i,\mathrm{ini}}\\\epsilon_{i,\mathrm{ini}}\\y_{i,\mathrm{ini}}\\u_i\\\epsilon_i\\y_i\end{bmatrix}+\begin{bmatrix}0\\0\\\sigma_{y_i}\\0\\0\\0\end{bmatrix}, \tag{14b}$$

$$\tilde{s}_{\min}\leq G_1y_i\leq\tilde{s}_{\max}, \tag{14c}$$

$$u_{\min}\leq u_i\leq u_{\max}, \tag{14d}$$

$$\epsilon_i=\epsilon_{i,\mathrm{est}}, \tag{14e}$$

Algorithm 1 Decentralized Robust DeeP-LCC

Input: Pre-collected local data $(u_i^{\mathrm{d}}, \epsilon_i^{\mathrm{d}}, y_i^{\mathrm{d}})$ for i-th subsystem, initial time step k_0, terminal time step k_f;

Construct data Hankel matrices for input, disturbance and output as $U_{i,\mathrm{P}}, U_{i,\mathrm{F}}, E_{i,\mathrm{P}}, E_{i,\mathrm{F}}, Y_{i,\mathrm{P}}, Y_{i,\mathrm{F}}$;
Initialize the most recent past traffic data $(u_{i,\mathrm{ini}}, \epsilon_{i,\mathrm{ini}}, y_{i,\mathrm{ini}})$ before the initial time k_0;
while $k_0 \leq k \leq k_f$ **do**
  Estimate $\mathcal{W}_i$ from $\epsilon_{i,\mathrm{ini}}$;
  Solve (16) and get optimal predicted input $u_i^* = \mathrm{col}(u_i^*(k), u_i^*(k+1), \ldots, u_i^*(k+N-1))$;
  Apply the input $u_i(k) \leftarrow u_i^*(k)$ to the i-th CAV;
  $k \leftarrow k+1$;
  Update past local data $(u_{i,\mathrm{ini}}, \epsilon_{i,\mathrm{ini}}, y_{i,\mathrm{ini}})$;
end while

where $V_i(u_i, y_i)$ penalizes the output deviation from equilibrium states and the energy of the input:

$$V_i(u_i, y_i) = ||u_i||_{R_i}^2 + ||y_i||_{Q_i}^2, \tag{15}$$

with $R \in \mathbb{S}_+^{N\times N}$ and $Q \in \mathbb{S}_+^{N(m_i+2)\times N(m_i+2)}$. The constraint (14b) is used for online behavior prediction (see Proposition 2), while the constraints (14c) and (14d) are used to ensure safety and driving comfort, respectively. Precisely, $G_1 = I_N \otimes \left[\mathbb{0}_{1\times(m_i+1)},\ 1\right]$ selects the spacing of the CAV from the output vector y_i, and $\tilde{s}_{\min}, \tilde{s}_{\max}$ and $u_{\max}, u_{\min}$ denote the upper and lower bounds for the spacing error and the control input, respectively. The constraint (14e) denotes the estimation of future velocity errors of the preceding vehicle.

Note that Proposition 2 only works for LTI mixed traffic systems with no measurement noises. Yet, any practical mixed traffic system is nonlinear with noises, and therefore, we introduce the regularization term $||g_i||_2^2$ to mitigate overfitting caused by noisy data, the extra decision variable σ_y to ensure the feasibility of the optimization problem, and the term $\|\sigma_y\|_2^2$ to penalize the deviation from the initial condition (which is consistent with the original DeePC [23]). Coefficients λ_g and λ_y are weighting parameters, and λ_y is usually large enough to satisfy the system's initial condition; see [40], [41] for details.

Remark 1 (Coupling dynamics and estimation of future velocity errors): The local DeeP-LCC in (14) designs the CAV's control input based on an estimated future velocity error trajectory ϵ_i of the preceding vehicle. A naive choice is the simplistic zero assumption in centralized DeeP-LCC (*i.e.*, $\epsilon_{i,\mathrm{est}} = \mathbb{0}_N$). This inaccurate estimation will cause a mismatch between the prediction and real behavior, which degrades control performance and may lead to a collision. In [29], the value of $\epsilon_{i,\mathrm{est}}$ is approximated through inter-vehicle data exchange via a distributed algorithm, where impractical high-frequency data exchange is required. In this paper, we estimate a set of potential future velocity error trajectories $\mathcal{W}_i$, which is more likely to contain the real trajectory. We then require the online optimization (14) to consider the worst performance against all trajectories in $\mathcal{W}_i$ and establish a robust formulation. □

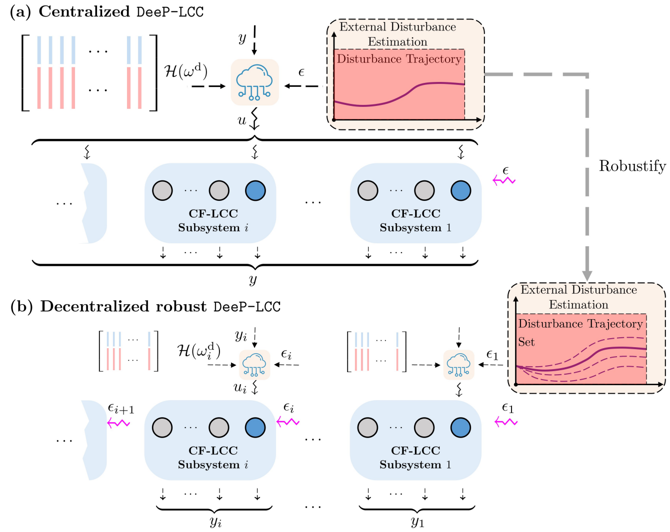


Fig. 2. Schematic of comparison between centralized and decentralized robust DeeP-LCC. The Hankel matrix is partitioned into the past trajectories (represented by blue columns) and future trajectories (represented by red columns). The future external disturbance is represented by purple squiggle arrows.

C. Decentralized Robust DeeP-LCC Formulation

As discussed in Remark 1, instead of considering a single velocity error trajectory, we introduce a velocity error set $\mathcal{W}_i$ as the estimation, *i.e.*, $\epsilon_i \in \mathcal{W}_i$ (the design of this set $\mathcal{W}_i$ will be detailed in Section IV). Our key idea here is to formulate an online robust optimization problem by considering the worst performance against $\mathcal{W}_i$. The decentralized robust DeeP-LCC formulation is formally presented as follows:

$$\begin{aligned} \min_{g_i,\sigma_{y_i},u_i,y_i} \quad \max_{\epsilon_i\in\mathcal{W}_i} \quad & V_i(u_i, y_i) + \lambda_{g_i}||g_i||_2^2 + \lambda_{y_i}||\sigma_{y_i}||_2^2 \\ \text{subject to} \quad & (14\text{b}),\ (14\text{c}),\ (14\text{d}). \end{aligned} \tag{16}$$

For notational simplicity, we will use cDeeP-LCC and dDeeP-LCC to represent centralized DeeP-LCC (9) and decentralized robust DeeP-LCC (16), respectively. Compared with the basic version (14), this robust formulation (16) improves the online behavioral prediction and thus provides a better safety performance (our numerical experiments also validate this). In the implementation of dDeeP-LCC, (16) is solved in a receding horizon manner, and we re-estimate $\mathcal{W}_i$ iteratively based on the updated velocity errors of the preceding vehicles at each time k (see Section IV for details). Overall, Algorithm 1 lists the procedure of dDeeP-LCC.

Remark 2 (Centralized vs decentralized formulations): In general, cDeeP-LCC considers the entire mixed traffic system with aggregated input $u(t)$ and output $y(t)$ (see Fig. 2(a)). In contrast, dDeeP-LCC focuses on each CF-LCC subsystem (see Fig. 2(b)) which is more scalable in terms of computation, communication and data privacy. Note that the external disturbance ϵ is simply estimated as zeros in cDeeP-LCC (which is unrealistic), while a set of trajectories ($\epsilon_i \in \mathcal{W}_i$) is under consideration in dDeeP-LCC. Additionally, the velocity error of the preceding vehicle, shown in (10), indeed comes from the dynamics of the $(i-1)$-th CF-LCC subsystem. Thus, CF-

LCC subsystems are coupled in a cascading structure, forming the mixed traffic. Our cDeeP-LCC decouples the coupling dynamics as a bounded disturbance using $\epsilon_i \in \mathcal{W}_i$. □

Remark 3 (Data requirement in dDeeP-LCC): The centralized cDeeP-LCC collects inputs and outputs of all vehicles to construct the associated Hankel matrix for the entire mixed traffic system, while dDeeP-LCC only uses the local input and output to form the Hankel matrix for each CF-LCC subsystem. Thus, our dDeeP-LCC requires less pre-collected data for online behavioral prediction because of the smaller system's dimension, leading to less complexity for online optimization. In particular, persistently exciting conditions in Propositions 1 and 2 require Hankel matrices $\mathcal{H}_{L+2n}(\bar{u}^{\mathrm{d}})$ and $\mathcal{H}_{L+2(m_i+1)}(\bar{u}_i^{\mathrm{d}})$ to have full row rank. To make them square matrices, the minimum data lengths L_{cen} and L_{decen} for cDeeP-LCC and dDeeP-LCC are, respectively, as

$$L_{\text{cen}} = (q+2)(L+2n) - 1, \tag{17a}$$

$$L_{\text{decen}} = 3(L + 2(m_i+1)) - 1. \tag{17b}$$

Recall that q is the number of CAVs. Since the size of each subsystem is much smaller than the entire system (*i.e.,* we have $q+2 \geq 3$, $n \gg m_i + 1$ generally), the dDeeP-LCC needs much less data for a smaller Hankel matrix; see Fig. 2. Thus, dDeeP-LCC has a smaller data representation in (13) which will improve the online computation efficiency. □

IV. Disturbance Estimation and Approximation

Estimating the set of velocity error trajectories ($\epsilon_i \in \mathcal{W}_i$) is a key step in dDeeP-LCC that decomposes the whole system since CF-LCC subsystems are dynamically coupled through velocity perturbation from their preceding vehicles. In this section, we first introduce three disturbance estimation approaches based on different assumptions of the preceding vehicle. We then provide a low-dimensional approximation method via a down-sampling strategy for more efficient implementation.

A. Uncertainty Quantification

Our general strategy is to bound the set of velocity errors using an N-dimensional polytope

$$\mathcal{W}_i = \{\epsilon_i \in \mathbb{R}^N \mid A_\epsilon \epsilon_i \leq b_\epsilon\}, \tag{18}$$

where $A_\epsilon = \mathrm{col}(I_N, -I_N)$, $b_\epsilon = \mathrm{col}(\epsilon_{i,\max}, \epsilon_{i,\min})$ and $\epsilon_{i,\min}, \epsilon_{i,\max} \in \mathbb{R}^N$ are the lower and upper bound of ϵ at each time step. We will estimate the bounds b_ϵ of $\mathcal{W}_i$ for uncertainty quantification. To avoid an overly conservative bound of future velocity errors, we estimate the bound b_ϵ based on the past velocity errors ϵ_{ini} and the dynamic of the mixed traffic system.

We consider three different disturbance estimation methods: 1) zero estimation, 2) constant velocity, and 3) constant acceleration (see Fig. 3 for illustration). We detail them below:

1) Zero estimation: A simplistic approach is to assume the future velocity error of the preceding vehicle is around zero, which is adopted in cDeeP-LCC [22]. This assumption makes $\epsilon_{i,\min} = \epsilon_{i,\max} = \mathbb{0}_N$ and (16) is reduced to (14) with $\epsilon_{i,\text{est}} = \mathbb{0}_N$. We consider this choice as a baseline where we only decentralize the mixed traffic system without robustification.

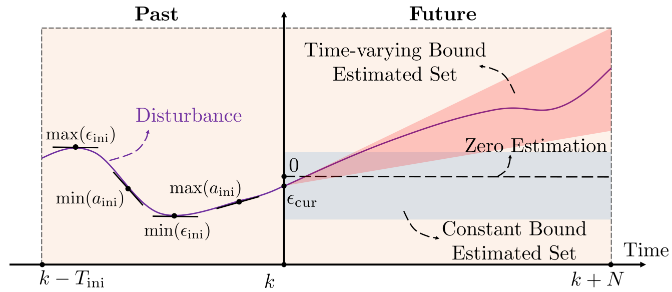


Fig. 3. Schematic of three disturbance estimation methods. The purple line denotes the actual disturbance trajectory, whose past is known while its future needs to be estimated. The zero estimation is denoted as the black dashed line, while the time-varying bound estimated set and the constant bound estimated set are represented as the red region and the blue region, respectively.

2) Constant bound estimation: We here assume a constant velocity model, *i.e.*, the velocity error of the preceding vehicle will not deviate from its current value in a short time, and the variation for the future velocity trajectory is close to its past trajectory. From the historical disturbance values $\epsilon_{i,\text{ini}}$, we can get the value of the current disturbance as $\epsilon_{i,\text{cur}} = \epsilon_{i,\text{ini}}(\text{end})$, and the disturbance variation can be estimated as $\Delta\epsilon_{i,\text{low}} = \min(\epsilon_{i,\text{ini}}) - \text{mean}(\epsilon_{i,\text{ini}})$ and $\Delta\epsilon_{i,\text{up}} = \max(\epsilon_{i,\text{ini}}) - \text{mean}(\epsilon_{i,\text{ini}})$. Then, the bound of the future disturbance is

$$\epsilon_{i,\min} = \epsilon_{i,\text{cur}} + \Delta\epsilon_{\text{low}},\ \ \epsilon_{i,\max} = \epsilon_{i,\text{cur}} + \Delta\epsilon_{\text{up}}.$$

3) Time-varying bound estimation: We then assume a constant acceleration model, *i.e.*, the acceleration of the preceding vehicle will not deviate significantly from its current value, and its variation in the future is close to the variation in the past. The past acceleration can be obtained from $\epsilon_{i,\text{ini}}$ as $a_{i,\text{ini}} = \frac{\epsilon_{i,\text{ini}}(k+1) - \epsilon_{i,\text{ini}}(k)}{\Delta t}$, where Δt is the sampling time. Utilizing a similar procedure as above, we can estimate the bound of acceleration variation as $\Delta a_{i,\text{low}} = \min(a_{i,\text{ini}}) - \text{mean}(a_{i,\text{ini}})$ and $\Delta a_{i,\text{up}} = \max(a_{i,\text{ini}}) - \text{mean}(a_{i,\text{ini}})$. Thus, we can derive the future velocity error in time step k using the bounds below:

$$\begin{aligned}\epsilon_{i,\text{ini}}(\text{end}) + (a_{i,\text{cur}} + \Delta a_{i,\text{low}}) \cdot k\Delta t &\leq \epsilon_i(k) \\ &\leq \epsilon_{i,\text{ini}}(\text{end}) + (a_{i,\text{cur}} + \Delta a_{i,\text{up}}) \cdot k\Delta t.\end{aligned}$$

Fig. 3 illustrates the three different disturbance estimation methods with a sampled disturbance trajectory. In Fig. 3, we observe that a large gap can exist between the actual disturbance trajectory and the zero assumption. For the constant bound, the actual disturbance trajectory stays in the estimated region (see blue region) in the short term, but it then deviates outside the set. By contrast, the time-varying bound contains the actual trajectory in the estimated set (see red region) in this case. In most of our numerical experiments (see Section VI), the time-varying bounds outperform the other two methods. This is mainly because the velocity oscillations are usually low frequency but their amplitude can be high in traffic waves.

Remark 4: There also exist other disturbance estimation methods in the literature [29], [35]. For distributed optimization in [29], high-frequency inter-vehicle communications are required between two neighboring CF-LCC subsystems to approximate future disturbances. This might provide more accurate estimation, but it is non-trivial to implement. For min-max `DeePC` in the smart grid in [35], the disturbance bound is assumed to be the largest set that contains the entire disturbance trajectory throughout the process. However, this setup is too conservative for traffic control, degrading the resulting closed-loop control performance. □

B. Down-sampling Strategy

Due to the exponential growth of the number of constraints in robust optimization [37], [38], it may not be computationally feasible to use the full estimated disturbance set (18) of dimension N in problem (16); see our derivation of the constraint number in Section V-C. We here use a down-sampling strategy to reduce the dimension of the disturbance set.

The basic idea is to choose one point for every T_s step along the N-dimensional disturbance trajectory and perform a linear interpolation. We denote the low-dimensional representation of the future disturbance trajectory ϵ_i as $\tilde{\epsilon}_i \in \mathbb{R}^{n_\epsilon}$ where $n_\epsilon = (\lfloor(\frac{N-2}{T_\mathrm{s}}\rfloor + 2)$. The approximated representation $\hat{\epsilon}_i$ becomes:

- if $1 \leq k \leq \tilde{k} \cdot T_\mathrm{s}$:

$$\hat{\epsilon}_i^{(k)} = \tilde{\epsilon}_i^{(\bar{k}+1)} + ((k-1) \bmod T_\mathrm{s}) \times \frac{\tilde{\epsilon}_i^{(\bar{k}+2)} - \tilde{\epsilon}_i^{(\bar{k}+1)}}{T_\mathrm{s}};$$

- if $\tilde{k} \cdot T_\mathrm{s} < k \leq N$:

$$\hat{\epsilon}_i^{(k)} = \tilde{\epsilon}_i^{(\tilde{k}+1)} + (k - \tilde{k} \cdot T_\mathrm{s} - 1) \times \frac{\tilde{\epsilon}_i^{(\tilde{k}+2)} - \tilde{\epsilon}_i^{(\tilde{k}+1)}}{N - \tilde{k} \cdot T_\mathrm{s} - 1},$$

where $\bar{k} = \lfloor \frac{k-1}{T_\mathrm{s}} \rfloor$ and $\tilde{k} = \lfloor \frac{N-2}{T_\mathrm{s}} \rfloor$. Then, $\epsilon_i \in \mathbb{R}^N$ can be represented by $\tilde{\epsilon}_i \in \mathbb{R}^{n_\epsilon}$ as

$$\epsilon_i \approx \hat{\epsilon}_i = E_\epsilon \tilde{\epsilon}_i, \tag{19}$$

where $\tilde{\epsilon}_i \in \tilde{\mathcal{W}}_i$ and $\tilde{\mathcal{W}}_i$ are estimated using the same approaches as those for $\mathcal{W}_i$, and E_ϵ denotes the linear interpolation process above.

As discussed in [35], using down-sampling by replacing ϵ_i with $\hat{\epsilon}_i$ may fail to cover the worst-case scenario, as the signal space of $\hat{\epsilon}_i$ is a subspace of ϵ_i. Nevertheless, our extensive traffic simulations in Section VI show that this approximation method could provide satisfactory traffic control performance.

V. Tractable Reformulations and Efficient Computations

Upon estimating $\tilde{W}_i$, the robust optimization problem (16) is well-defined. To run Algorithm 1, we need to solve (16) online efficiently. In this section, we first provide a sequence of reformulations (and relaxations) for (16) since the standard solvers are not suitable to the current form. We next adapt techniques from robust optimization [38] into our context, and develop an efficient automatic transformation that reformulates each decentralized robust `DeeP-LCC` problem (16) into a standard conic form. With slight abuse of notations, we will use x to denote the decision variable in this section, following the convention in optimization. For notational simplicity, we will also omit the subscript i for the i-th CF-LCC subsystem when it is clear from the context.

A. Reformulations via Constraint Elimination

The equality constraints (14b) in (16) make the robust optimization complicated. We first express the decision variables g_i and y_i in terms of u_i, σ_{y_i}, and $\tilde{\epsilon}_i$. Recall that ϵ_i is replaced by its low-dimensional approximation $\tilde{\epsilon}_i$ from (19). We have

$$g_i = H_i^\dagger b + H_i^\perp z, \tag{20a}$$

$$y_i = Y_{i,\mathrm{F}} g_i = Y_{i,\mathrm{F}} H_i^\dagger b + Y_{i,\mathrm{F}} H_i^\perp z, \tag{20b}$$

where $H_i = \mathrm{col}(U_{i,\mathrm{P}}, E_{i,\mathrm{P}}, Y_{i,\mathrm{P}}, U_{i,\mathrm{F}}, E_{i,\mathrm{F}})$ denotes the Hankel data matrix for each CF-LCC system, $H_i^\dagger$ is its pseudo-inverse matrix, $H^\perp = I - H^\dagger H$, z is an arbitrary vector in $\mathbb{R}^{T-L+1}$, and $b = \mathrm{col}(u_{i,\mathrm{ini}}, \epsilon_{i,\mathrm{ini}}, y_{i,\mathrm{ini}} + \sigma_{y_i}, u_i, E_\epsilon \tilde{\epsilon}_i)$ denotes the vector on the right-hand side of (14b).

In the following derivation, we set $z = 0$, which leads to the least-norm solution of g_i for the equality constraint (14b). After some algebra (see Appendix A), the min-max robust problem (16) can be rewritten into the form of

$$\min_{\sigma_{y_i}, u_i} \max_{\tilde{\epsilon}_i \in \tilde{\mathcal{W}}_i} \quad x^\mathsf{T} M x + d^\mathsf{T} x + c_0 \tag{21a}$$

$$\text{subject to} \quad \tilde{s}_{\min} \leq P_1 x + c_1 \leq \tilde{s}_{\max}, \tag{21b}$$

$$u_{\min} \leq P_2 x \leq u_{\max}, \tag{21c}$$

where x denotes the decision variables $\mathrm{col}(u_i, \sigma_{y_i}, \tilde{\epsilon}_i)$, the weight M is a constant positive semidefinite matrix, d is a constant vector, c_0 is a scalar, P_1, c_1 are constants that construct y_i in (20b) from x, and P_2 is an index matrix extracting u_i in (14d) from x. We note that M, P_1, P_2 can be pre-computed before running Algorithm 1. Precisely, M, P_1 only depend on the Hankel matrix and the weights λ_g, λ_y, and P_2 is a constant indexing matrix. On the other hand, d, c_0 and c_1 depend on the initial trajectory and need to be computed at each time step. More details can be found in Appendix A.

We then treat $\tilde{\epsilon}_i$ as uncertainty parameters, eliminate the constant c_0 and transform (21) into the epi-graph form below

$$\min_{x,t} \quad t$$

$$\text{subject to} \quad x^\mathsf{T} M x + d^\mathsf{T} x \leq t, \quad \forall \tilde{\epsilon}_i \in \tilde{\mathcal{W}}_i \tag{22a}$$

$$\tilde{s}_{\min} \leq P_1 x + c_1 \leq \tilde{s}_{\max}, \quad \forall \tilde{\epsilon}_i \in \tilde{\mathcal{W}}_i \tag{22b}$$

$$u_{\min} \leq P_2 x \leq u_{\max}, \tag{22c}$$

which ensures safety constraints for all $\tilde{\epsilon}_i$ in $\tilde{\mathcal{W}}_i$.

B. Vertex-based and Duality-based Strategies for Computation

The uncertainty set $\tilde{\mathcal{W}}_i$ is a compact polytope (see Section IV), and thus (22a) and (22b) have an infinite number of constraints, which is not directly solvable in the current form. We here adapt techniques from robust optimization [38] to solve (22) by exploiting the representations of $\tilde{\mathcal{W}}_i$.

In particular, it is known that the compact polytope $\tilde{\mathcal{W}}_i$ has two different representations: 1) internal representations as its convex hull of extreme points (*i.e.*, its vertices), and 2) external representations as an intersection of affine subspaces. We proceed to derive a computable form for each of them, leading to two different methods respectively: 1) the vertex-based strategy and 2) the duality-based strategy.

To simplify notations, we denote

$$\tilde{\mathcal{W}}_i = \mathrm{conv}(w_1, \dots, w_{n_\mathrm{v}}) \tag{23a}$$

$$\tilde{\mathcal{W}}_i = \{\tilde{\epsilon}_i \in \mathbb{R}^{n_\epsilon} \mid \tilde{A}_\epsilon \tilde{\epsilon}_i \le \tilde{b}_\epsilon\}, \tag{23b}$$

where n_v is the number of extreme points, $w_1, \dots, w_{n_\mathrm{v}} \in \mathbb{R}^{n_\epsilon}$ are the extreme points of $\tilde{\mathcal{W}}_i$, and $\tilde{A}_\epsilon = \mathrm{col}(I_{n_\epsilon}, -I_{n_\epsilon})$, $\tilde{b}_\epsilon = \mathrm{col}(\tilde{\epsilon}_{i,\max}, \tilde{\epsilon}_{i,\min})$ are the low-dimensional approximation for affine constraints in (18).

The first method is summarized in the following proposition.

Proposition 3 (Method I: Vertex-based Strategy): Using the extreme point representation (23a), the optimization (22) is equivalent to the following problem

$$\begin{aligned}
\min_{x,t} \quad & t \\
\text{subject to} \quad & x_j^\mathsf{T} M x_j + d^\mathsf{T} x_j \le t, j = 1, \dots, n_\mathrm{v}, && (24a)\\
& \tilde{s}_{\min} \le P_1 x_j + c_1 \le \tilde{s}_{\max}, j = 1, \dots, n_\mathrm{v}, && (24b)\\
& u_{\min} \le P_2 x \le u_{\max}, && (24c)
\end{aligned}$$

where x_j denotes the decision variables $\mathrm{col}(u_i, \sigma_{y_i}, w_j)$ by fixing $\tilde{\epsilon}_i$ to be one of the extreme points w_j.

Proof: We only need to prove that (22a) and (22b) are equivalent to (24a) and (24b), respectively.

The equivalence (22b) ⇔ (24b) is a direct consequence of the basic result in linear programs (LPs). In particular, the right-hand inequality of (22b) is the same as

$$\tilde{s}_{\max} \ge \max_{\tilde{\epsilon}_i \in \tilde{\mathcal{W}}_i} \; p_l^\mathsf{T} x + c_{1,l}, \qquad l = 1, \dots, N, \tag{25}$$

where p_l^T is the l-th row vector in P_1, and $c_{1,l}$ is l-th element in c_1. Recall that N is the predication horizon, and (25) just enforces the spacing error of the i-th CAV to be upper bounded by the maximum spacing error $\tilde{s}_{\max}$ for all time steps. Since (25) is an LP with a bounded feasible region, the maximum is always attained at one of its vertices of $\tilde{\mathcal{W}}_i$, represented in (23a). Therefore, (25) is equivalent to the right-hand inequality of (24b). Similarly, the argument for the left-hand inequality of (22b) can be obtained.

The equivalence (22a) ⇔ (24a) requires a slightly different argument from convex sets. The direction (22a) ⇒ (24a) is immediate since all vertices are contained in set $\tilde{\mathcal{W}}_i$. As for the converse direction (24a) ⇒ (22a), we consider any element $\tilde{\epsilon}_i \in \tilde{\mathcal{W}}_i$. A basic result in convex analysis guarantees the convex representation

$$\tilde{\epsilon}_i = \sum_{j=1}^{n_\mathrm{v}} \alpha_j w_j, \quad \sum_{j=1}^{n_\mathrm{v}} \alpha_j = 1, \quad \alpha_j \ge 0,$$

where $w_j, j = 1, \dots, n_\mathrm{v}$ are the extreme points (23a). Following the same notation in (24), we also have $x = \sum_{j=1}^{n_\mathrm{v}} \alpha_j x_j$. Then, we have

$$\begin{aligned}
x^\mathsf{T} M x + d^\mathsf{T} x &\le \sum_{j=1}^{n_\mathrm{v}} \alpha_j (x_j^\mathsf{T} M x + d^\mathsf{T} x_j) \\
&\le \sum_{j=1}^{n_\mathrm{v}} \alpha_j t = t, \quad \forall \tilde{\epsilon}_i \in \tilde{\mathcal{W}}_i,
\end{aligned}$$

where the first inequality comes from the fact that $x^\mathsf{T} M x + d^\mathsf{T} x$ is a convex function, and the second inequality is from (24a). This completes the proof. ■

Problem (24) is a standard convex problem and can be solved using existing solvers. However, the number of extreme points n_v can be large even for a simple polytope, which indicates the number of constraints in (24) can be large. We next introduce another equivalent formulation that utilizes duality analysis to reduce the number of constraints, which is summarized below.

Proposition 4 (Method II: Duality-based strategy): Using the representation as an intersection of affine subspaces (23b), the optimization (22) is equivalent to the following problem

$$\begin{aligned}
\min_{x_\mathrm{d}, t, \lambda_1, \lambda_2} \quad & t \\
\text{subject to} \quad & p_{l,\mathrm{d}}^\mathsf{T} x_\mathrm{d} + \tilde{b}_\epsilon^\mathsf{T} \lambda_{l,1} + c_{1,l} \le \tilde{s}_{\max}, && (26a)\\
& \tilde{A}_\epsilon^\mathsf{T} \lambda_{l,1} - p_{l,\epsilon} = 0, && (26b)\\
& - p_{l,\mathrm{d}}^\mathsf{T} x_\mathrm{d} + \tilde{b}_\epsilon^\mathsf{T} \lambda_{l,2} - c_{1,l} \le -\tilde{s}_{\min}, && (26c)\\
& \tilde{A}_\epsilon^\mathsf{T} \lambda_{l,2} + p_{l,\epsilon} = 0, && (26d)\\
& \lambda_{l,1} \ge 0, \lambda_{l,2} \ge 0, \; l = 1, 2, \dots, N, && (26e)\\
& \text{(24a)}, \text{(24c)},
\end{aligned}$$

where $x_\mathrm{d} = \mathrm{col}(u_i, \sigma_{y_i})$ denotes the decision variable, and $\lambda_{l,1} \in \mathbb{R}^{2n_\mathrm{v}}$ is the dual variable corresponding to the affine constraints in (25), with $\lambda_1 = \mathrm{col}(\lambda_{1,1}\lambda_{2,1}, \dots, \lambda_{N,1})$. Similar definitions hold for $\lambda_{l,2}$ and λ_2. The coefficients $\tilde{A}_\epsilon, \tilde{b}_\epsilon$ can be found in (23b), and $c_{1,l}, p_l$ are the same as (25). We use $p_{l,\mathrm{d}}$ to denote $\mathrm{col}(p_{l,u}, p_{l,\sigma_y})$ with p_l being partitioned as $\mathrm{col}(p_{l,u}, p_{l,\sigma_y}, p_{l,\epsilon})$, corresponding to the original variables u_i, σ_{y_i} and $\tilde{\epsilon}_i$, respectively.

Proof: We have shown that (22a) is equivalent to (24a). We only need to prove the equivalence between (22b) and (26a) - (26e). Recall that the right-hand inequality of (22b) is equivalent to the LP in (25). Via duality analysis, we shall prove that (25) will be the same as (26a), (26b) and (26e). The argument for the left-hand inequality of (22b) is identical.

To derive the dual of the LPs in (25), we first write its Lagrangian function for $l = 1, \dots, N$ as

$$\begin{aligned}
L(\tilde{\epsilon}_i, \lambda_{l,1}) &= p_l^\mathsf{T} x + c_{1,l} + \lambda_{l,1}^\mathsf{T} (\tilde{b}_\epsilon - \tilde{A}_\epsilon \tilde{\epsilon}_i) \\
&= (p_{l,\epsilon}^\mathsf{T} - \lambda_{l,1}^\mathsf{T} \tilde{A}_\epsilon) \tilde{\epsilon}_i + p_{l,\mathrm{d}}^\mathsf{T} x_\mathrm{d} + \lambda_{l,1}^\mathsf{T} \tilde{b}_\epsilon + c_{1,l}.
\end{aligned}$$

The dual function is thus given by

$$h(\lambda_{l,1}) = \begin{cases} p_{l,\mathrm{d}}^\mathsf{T} x_\mathrm{d} + \lambda_{l,1}^\mathsf{T} \tilde{b}_\epsilon + c_{1,l} & \text{if } \; p_{l,\epsilon}^\mathsf{T} - \lambda_{l,1}^\mathsf{T} \tilde{A}_\epsilon = 0 \\ -\infty & \text{otherwise.} \end{cases}$$

Then, the dual problem for the LPs in (25) becomes

$$\begin{aligned}
f_l^\star = \min_{\lambda_{l,1}} \quad & p_{l,\mathrm{d}}^\mathsf{T} x_\mathrm{d} + \tilde{b}_\epsilon^\mathsf{T} \lambda_{l,1} + c_{1,l} \\
\text{subject to} \quad & \tilde{A}_\epsilon^\mathsf{T} \lambda_{l,1} - p_{l,\epsilon} = 0, \lambda_{l,1} \succeq 0.
\end{aligned} \tag{27}$$

The strong duality of LPs ensures that (25) is the same as

$$\tilde{s}_{\max} \ge f_l^\star, \qquad l = 1, \dots, N. \tag{28}$$

TABLE I
COMPLEXITY COMPARISON BETWEEN (24) AND (26).

	Decision Variables Number	Constraints Number
Method I	$(m_i+2)T_{\text{ini}}+N+1$	$2^{n_\epsilon}+N\cdot 2^{n_\epsilon+1}+2N$
Method II	$(m_i+2)T_{\text{ini}}+N+1+4Nn_\epsilon$	$2^{n_\epsilon}+2N(3n_\epsilon+2)$

[1] We recall that m_i is the number of HDVs in the i-th subsystem, T_{ini} is the length of the initial trajectory, N is the length of prediction horizon and n_ϵ is the dimension of the approximated disturbance set.

We can thus replace the right-hand inequality of (22b) with (27) and (28), leading to a bi-level minimization problem. Simple algebra allows us to combine both levels[1], confirming the equivalence between (27) and (26a), (26b), (26e). This completes our proof. ∎

C. Complexity Comparison and Automatic Transformation

We here compare the complexity of Methods I and II. One main complexity difference is caused by using different representations of (25) to derive (22b), which corresponds to (24b) in Method I and (26a)-(26e) in Method II.

In our problem, the uncertainty set is a polytope with dimension n_ϵ and we have $n_{\text{v}} = 2^{n_\epsilon}$. For Method I, (24b) represents $2N\cdot 2^{n_\epsilon}$ inequality constraints and its complexity will increase exponentially as n_ϵ grows up. Moreover, $2N$ will be a relatively large coefficient since the prediction horizon in our problem is usually larger than 20 (corresponding to 1 s). For Method II, (26a) - (26e) together represent $2N\cdot(3n_\epsilon+1)$ inequality constraints, whose size is much smaller than the $2N\cdot 2^{n_\epsilon}$ constraints in (24b) given a large value of n_ϵ. Table I illustrates the complexity of (24) and (26). As we can observe, when the prediction horizon N is fixed, Method II has a smaller complexity than Method I given a large value of n_ϵ.

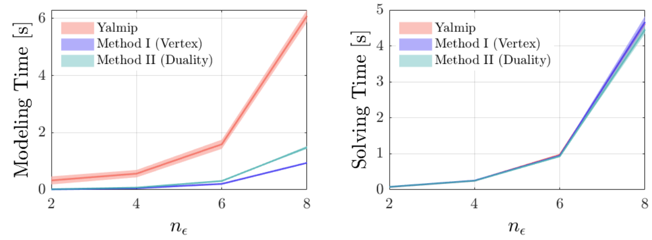
The theoretical complexity result is only one key factor that influences the time consumption. For numerical computation, we need to convert (24) or (26) into a standard conic optimization before passing them to a numerical solver. For this process, we find that modeling packages such as YALMIP [39] often introduce too much overhead time consumption. Also, similar instances of (24) or (26) need to be converted repeatedly for the receding horizon online control in Algorithm 1. Thus, we here implement an automatic transformation from (24) or (26) into standard conic optimization, so that they can be directly solved by solvers (*e.g.*, Sedumi [42], Mosek [43]).

Precisely, their standard conic dual form for (24) or (26) can be written as

$$\begin{aligned} \max_{y,s} \quad & b^{\mathsf{T}} y \\ \text{subject to} \quad & A^{\mathsf{T}} y + s = c \\ & s \in \mathcal{K}, \end{aligned} \tag{29}$$

where A is a matrix, y,b,c,s are vectors and $\mathcal{K}$ is a cone. We present a detailed transformation from (24) and (26) to (29) in

[1]This operation is standard. The interested reader may see, *e.g.*, Section 2.1 of this note: https://zhengy09.github.io/ECE285/lectures/L17.pdf.



(a) Modeling time (b) Solver time

Fig. 4. Comparison of computational time for different methods.

Appendix B. We note that our automatic transformation does not introduce extra decision variables or constraints.

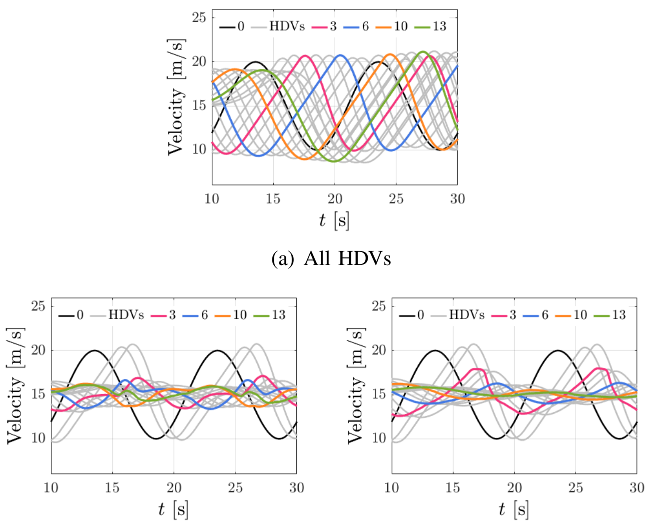
Here, we use a simple numerical experiment to demonstrate the advantage of our automatic transformation compared to modeling packages such as YALMIP [39] for solving (24) and (26). In particular, we consider two types of time costs: modeling time and solver time. Modeling time refers to the time needed to update the parameters of the conic form (29) in each iteration, while solver time is the time the solver takes to solve the resulting conic program (29). For the numerical experiment, we fix the parameters $m_i = 3, T_{\text{ini}} = 20, N = 50$ and vary n_ϵ. For each value of n_ϵ, we compute 20 instances using different methods. The results are shown in Fig. 4. As we can observe, both Vertex-based and Duality-based methods have comparable computational performance. With the increase of n_ϵ, Method I has faster modeling time while Method II performs slightly better in the solver time. Also, our automatic transformation provides much faster modeling time with similar solving time compared to YALMIP. In our following experiments (Section VI), we will thus use the automatic transformation for Method I to solve `dDeeP-LCC` (16).

VI. NUMERICAL EXPERIMENTS

In this section, we present three nonlinear and non-deterministic traffic simulations to validate the performance of `dDeeP-LCC`[2]. The driving behaviors of HDVs are modeled by a nonlinear OVM model, similar to [11]. A noise with a uniform distribution of $\mathbb{U}[-0.1, 0.1]\ \mathrm{m/s}^2$ is added to the acceleration signal of each HDV in the simulation. We utilize `dDeeP-LCC`(Zero), `dDeeP-LCC`(Constant), and `dDeeP-LCC`(Time-varying) to represent different disturbance estimation methods in Section IV.

For the mixed traffic system in Fig. 1, we consider a system with sixteen vehicles behind one head vehicle, among which there are four CAVs and twelve HDVs (*i.e.*, $n=16$, $q=4$). The CAVs are located in the third, sixth, tenth, and thirteenth vehicles, respectively (*i.e.*, $\mathcal{S} = \{3,6,10,13\}$). Accordingly, there are four CF-LCC subsystems, and the corresponding index sets for HDVs are $F_1 = \{4,5\}$, $F_2 = \{7,8,9\}$, $F_3 = \{11,12\}$ and $F_4 = \{14,15,16\}$. Other parameters are set as follows:

[2]All our implementation and experimental scripts can be found at https://github.com/soc-ucsd/Decentralized-DeeP-LCC.



(a) All HDVs

(b) `cDeeP-LCC`

(c) `dDeeP-LCC`

Fig. 5. Velocity profiles in Experiment A where the head vehicle is under sinusoidal perturbation. The black profile represents the head vehicle and the gray profile represents the HDVs. The red profile (vehicle 3), blue profile (vehicle 6), orange profile (vehicle 10) and green profile (vehicle 13) represent the first to the fourth CAV, respectively. (a) All vehicles are HDVs. (b) CAVs utilize `cDeeP-LCC`. (c) CAVs utilize `dDeeP-LCC`.

- *Offline data collection:* lengths of pre-collected trajectories are chosen as $T = 700$ for a small data set and $T = 1500$ for a large data set with $\Delta t = 0.05$ s. They are collected around the equilibrium state of the system with velocity $15\ \mathrm{m/s}$. We use a uniform distributed signal of $\mathbb{U}[-1, 1]$ to generate both u^{d} and ϵ^{d} which satisfies persistent excitation requirement in Propositions 1 and 2.
- *Online predictive control:* The prediction horizon and the initial signal sequence are set to $N = 50$ and $T_{\mathrm{ini}} = 20$, respectively. In the cost function (14a), we have $\lambda_{g_i} = 10$, $\lambda_{y_i} = 10000$, $R_i = 0.1 I_N$, $Q_i = I_N \otimes \mathrm{diag}(Q_{v_i}, w_{s_i})$ where $Q_{v_i} = \mathrm{diag}(1, \ldots, 1) \in \mathbb{R}^{m_i+1}$ and $w_{s_i} = 0.5$. The spacing constraints for the CAVs are set as $s_{\max} = 40$ m, $s_{\min} = 5$ m and the bound of the spacing error is updated at each time step as $\tilde{s}_{\max} = s_{\max} - s^*$ and $\tilde{s}_{\min} = s_{\min} - s^*$. The acceleration bound of the CAVs is set to $a_{\max} = 2\ \mathrm{m/s^2}$ and $a_{\min} = -5\ \mathrm{m/s^2}$. Note that s^* is updated in each iteration of Algorithm 1 based on the current equilibrium state, estimated by the past trajectory of the head vehicle (as suggested in [22]).

A. Performance Validation around an Equilibrium State

Our first experiment (Experiment A) simulates a traffic wave scenario by imposing a sinusoidal perturbation on the head vehicle. This is to validate the performance of `dDeeP-LCC` in smoothing traffic waves around a fixed equilibrium point. The head vehicle accelerates and decelerates periodically around the equilibrium velocity $15\,\mathrm{m/s}$ and its velocity profile is $v_{\text{head}}(t) = (15 + 5 \cdot \sin(0.2\pi t))\ \mathrm{m/s}$ (see black profile in Fig. 5). To quantify the closed-loop performance, we consider

TABLE II
MSVE STATISTICS IN EXPERIMENT A

	HDVs	`cDeeP-LCC`	`dDeeP-LCC` (Zero)	`dDeeP-LCC` (Constant)	`dDeeP-LCC` (Time-varying)
MSVE	9.25	0.57	1.28	1.33	0.76
Reduction	N/A	**93.8%**	86.2%	85.6%	**91.8%**

[1] Units for MSVE is $\mathrm{m^2/s^2}$ in this table.

the mean squared velocity error (MSVE), defined as

$$\mathrm{MSVE} = \frac{\Delta t}{n(t_f - t_0)} \sum_{t=t_0}^{t_f} \sum_{i=1}^{n} (v_i(t) - v_0(t))^2,$$

where t_0 and t_f denote the start and end time of the simulation.

As shown in Fig. 5(a), when all vehicles are HDVs, the amplitude of velocity perturbation from the head vehicle is amplified along the vehicle string. In contrast, when the CAVs are equipped with either `cDeeP-LCC` or `dDeeP-LCC`(Time-varying), the amplitude of these perturbations is greatly reduced, as shown in Fig. 5(b) and Fig. 5(c), respectively. Also, `dDeeP-LCC`(Time-varying) reduces 91.8% of the MSVE, which is comparable to 93.8% for `cDeeP-LCC`; see Table II. This indicates that `dDeeP-LCC` with an appropriate disturbance estimation method can effectively dissipate traffic waves with negligible performance degradation compared with `cDeeP-LCC`. One major benefit of `dDeeP-LCC` is its computational scalability, thanks to the decentralization. We discuss this in Section VI-D after introducing other experiments.

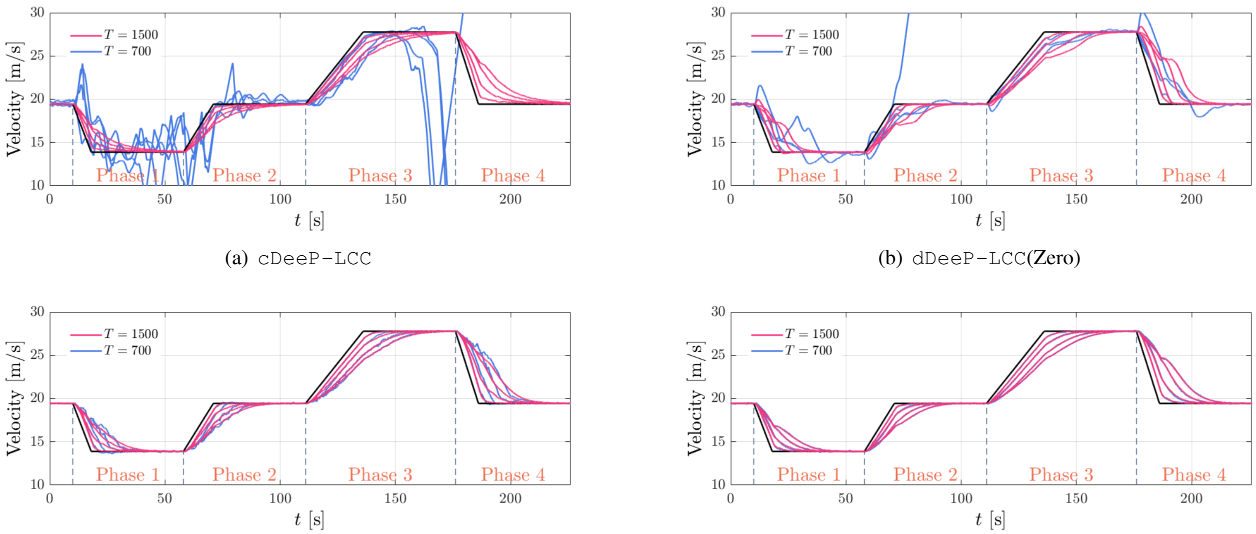
B. Traffic Improvement in Comprehensive Simulations

Our next Experiment B validates the performance of `dDeeP-LCC` in a comprehensive simulation scenario with time-varying equilibrium states. In particular, we consider the New European Driving Cycle (NEDC) [44], where the velocity profile for the head vehicle is shown in the black profile in Fig. 6. We also compute the fuel consumption of 4 CF-LCC subsystems for comparison. The fuel consumption rate f_i (mL/s) for the i-th vehicle is calculated by [45]

$$f_i = \begin{cases} 0.444 + 0.090 R_i v_i + [0.054 a_i^2 v_i]_{a_i > 0}, & \text{if } R_i > 0, \\ 0.444, & \text{if } R_i \leq 0, \end{cases}$$

where a_i represents the acceleration of vehicle i and $R_i = 0.333 + 0.00108 v_i^2 + 1.200 a_i$; see [45] for details.

Fig. 6 shows the velocity profiles for `cDeeP-LCC` and `dDeeP-LCC` utilizing different estimation methods with two data sets ($T = 700, 1500$). When using the large data set ($T = 1500$), all the methods enable the CAVs to track the desired velocity and smooth traffic flow (red curves in Fig. 6). By contrast, when using the small data set ($T = 700$), both `cDeeP-LCC` and `dDeeP-LCC`(Zero) fail to stabilize the closed-loop traffic (blue curves in Fig. 6(a) 6(b)). Still, `dDeeP-LCC`(Constant) and `dDeeP-LCC`(Time-varying) can stabilize the mixed traffic system. Moreover, `dDeeP-LCC`(Time-varying) achieves the best car-following



(a) cDeeP-LCC
(b) dDeeP-LCC(Zero)
(c) dDeeP-LCC(Bound)
(d) dDeeP-LCC(Time-varying)

Fig. 6. Velocity profiles in Experiment B. The black profile denotes the head vehicle. The red profiles and the blue profiles represent the data-driven control for CAVs with data sets of size $T = 1500$ and $T = 700$, respectively. (a) The CAV utilizes cDeeP-LCC. (b)(c)(d) The CAV utilizes dDeeP-LCC with zero, constant bound and time-varying bound estimation approaches.

TABLE III
FUEL CONSUMPTION COMPARISON (UNIT: mL)

	HDVs	cDeeP-LCC	dDeeP-LCC		
			Zero	Constant	Time-varying
NEDC	6548.36	6346.16	6388.47	6360.69	6356.14
Braking	1296.15	894.68	900.36	880.35	874.35

TABLE IV
FUEL CONSUMPTION COMPARISON IN EXPERIMENT B (UNIT: mL)

	All HDVs	cDeeP-LCC	dDeeP-LCC
Phase 1	749.23	700.99 (↓ 6.44%)	699.89 (↓ 6.58%)
Phase 2	1351.18	1331.75 (↓ 1.44%)	1333.29 (↓ 1.32%)
Phase 3	2904.48	2882.21 (↓ 0.77%)	2886.86 (↓ 0.61%)
Phase 4	1304.59	1191.91 (↓ 8.64%)	1196.34 (↓ 8.30%)
Total Process	6548.36	6346.16 (↓ **3.09%**)	6356.14 (↓ **2.94%**)

[1] We use time-varying bound estimation for dDeeP-LCC in this table.

behaviors for the CAVs with the smallest velocity oscillations (see the blue curves in Fig. 6(d)).

Compared with dDeeP-LCC(Time-varying), the performance degradation of the other methods is caused by the behavior representation error due to a smaller data set and the disturbance estimation error. Both types of errors can lead to a mismatch between the online prediction and real system behavior. We note that, via (17), the theoretical minimum data length is 611 for a centralized setting and 233 for a decentralized setting (computed using the largest subsystem). Since we use a linearized model to approximate the nonlinear system, the data length needs to be much larger than the theoretical value. Thus, cDeeP-LCC relies on a relatively large data set ($T = 1500$) for a valid representation of the entire system behavior, while the small data set ($T = 700$) is not large enough which fails to stabilize the closed-loop performance. In dDeeP-LCC, from the zero bound, the constant bound to the time-varying bound, the estimation error decreases and provides more margin for potential representation errors, leading to improved control performance. Thus, dDeeP-LCC(Time-varying) outperforms other methods for a small data set.

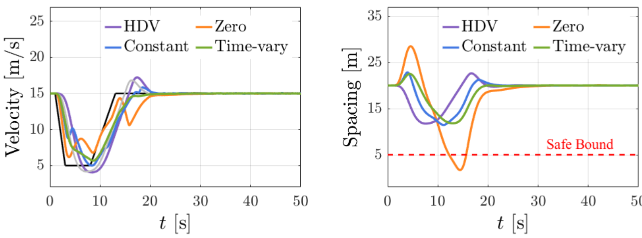
The fuel consumption results with the large data set ($T = 1500$) are listed in the first row of Table III, and a detailed comparison between cDeeP-LCC and dDeeP-LCC(Time-varying) is shown in Table IV. Compared with the case with all HDVs, all control methods can reduce fuel consumption, and the dDeeP-LCC(Time-varying) performs the best among different disturbance estimation methods. Also, we observe from Table IV that the improvement of implementing DeeP-LCC in the braking phase (Phases 1 and 4) is higher than the accelerating phases (Phases 2 and 3). Moreover, the dDeeP-LCC(Time-varying) achieves comparable fuel economy as the cDeeP-LCC, 2.94% and 3.09% for overall reduction of fuel consumption.

C. Safety Performance in Emergence Braking

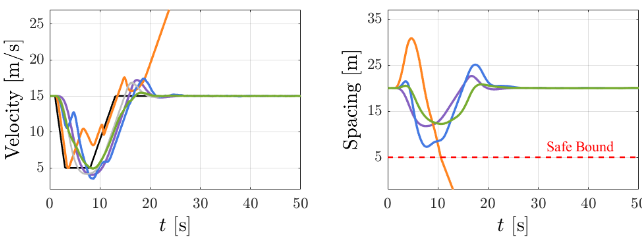
Our last Experiment C uses an emergency braking scenario to validate the safety performance of dDeeP-LCC with different disturbance estimation methods. The head vehicle that moves at $15\,\mathrm{m/s}$ will brake with maximum deceleration $-5\,\mathrm{m/s^2}$, remain at $5\,\mathrm{m/s}$ for a while and accelerate back to $15\,\mathrm{m/s}$. We carry out the same experiment for all the dDeeP-LCC controllers with 100 small data sets ($T = 700$) and 100 large data sets ($T = 1500$). As discussed in Section IV-A, we take the zero estimation as a baseline since it can be considered as applying cDeeP-LCC for each subsystem without robustification. We recall that the safety spacing constraint for the CAV is set from $5\,\mathrm{m}$ to $40\,\mathrm{m}$. The "violation"

TABLE V
COLLISION AND SAFETY CONSTRAINT VIOLATION RATE

	Zero		Constant		Time-varying	
	$T=700$	$T=1500$	$T=700$	$T=1500$	$T=700$	$T=1500$
Violation	99%	88%	**0%**	0%	**0%**	0%
Emergency	97%	68%	**0%**	0%	**0%**	0%



(a) Large offline data set with $T = 1500$



(b) Small offline data set with $T = 700$

Fig. 7. Simulation results in Experiment C. The black profile and the gray profile denote the head vehicle and the preceding vehicle, respectively. The orange profile, blue profile and the green profile represent `dDeeP-LCC`(Zero), `dDeeP-LCC`(Constant) and `dDeeP-LCC`(Time-varying) respectively, while the purple profile corresponds to the all HDV case. (a) and (b) show the velocity and spacing profiles at different sizes of data sets.

and "emergency" are defined as cases where at least one CAV's spacing deviates more than $1\,\mathrm{m}$ and $5\,\mathrm{m}$ from the safe range, respectively. We note that one of the following three undesired situations will happen when an emergency occurs: 1) A rear-end collision happens; 2) The CAV's spacing is too large which decreases the traffic capacity; 3) The controller can not stabilize the mixed traffic system.

The safety performance results are listed in Table V. With large data sets, `dDeeP-LCC`(Zero) has a high violation rate and emergency rate, which are 88% and 68% respectively, while the other two methods can provide much better safety guarantee (0% violation rate). The zero estimation method shows the worst robustness performance against large disturbances, which leads to a high risk of rear-end collisions. Given the small data sets, the violation rate and emergency rate for both `dDeeP-LCC`(Constant) and `dDeeP-LCC`(Time-varying) remain 0%, but they are close to 100% for `dDeeP-LCC`(Zero). Moreover, `dDeeP-LCC`(Time-varying) performs the best in reducing the fuel consumption among all methods with the large data set ($T = 1500$) as shown in the second row of Table III which decreases 32.5% fuel cost compared with the case of all HDVs. These results validate the superior performance of `dDeeP-LCC`(Time-varying), which works for small data sets and can also provide the best safety performance.

TABLE VI
COMPUTATION TIME (SECONDS)

	`cDeeP-LCC`	`dDeeP-LCC`		
		Zero	Constant	Time-varying
Traffic Wave	0.41	0.030	0.059	0.059
NEDC	0.37	0.032	0.059	0.057
Braking	0.39	0.027	0.057	0.059

In Fig. 7, we further illustrate the trajectories of the first CAV to compare the performance of utilizing different disturbance estimation methods. Compared with the case of all human drivers, we observe that all methods using a large data set can smooth the traffic flow. However, using a small data set, `dDeeP-LCC` with both zero and constant bound estimation approaches leads to undesired velocity oscillations. For the spacing profiles, `dDeeP-LCC`(Time-varying) and `dDeeP-LCC`(Constant) satisfy the safety constraint for both large and small data sets, but `dDeeP-LCC`(Zero) is likely to cause a rear-end collision at a small data set. Although the safety constraint is imposed in `dDeeP-LCC`(Zero), the mismatch between the prediction and real system behavior leads to its failure in the implementation. This is due to the oversimplified assumption in `dDeeP-LCC`(Zero) that the preceding vehicle tracks the equilibrium velocity accurately. On the other hand, the `dDeeP-LCC`(Time-varying) relies on a more accurate estimation of the future disturbances, and introduces a robust design that requires safety satisfaction for all the possible future disturbances. Therefore, `dDeeP-LCC`(Time-varying) provides a much better safety guarantee with a smoother velocity profile.

D. *Computational Efficiency of `dDeeP-LCC`*

Finally, we compare computational time for each method. In our experiments, we solve (9) and (24) using Mosek [43] on an Intel Core i7-9750H CPU with 16GB RAM. We consider (9) as a specific form of (24) by setting $n_{\mathrm{v}} = 1$ and $w_1 = \mathbb{0}_{n_\epsilon}$, and utilize the same method to solve them for comparison.

As shown in Table VI, the three methods of `dDeeP-LCC` achieve much faster computation time than `cDeeP-LCC`. The `dDeeP-LCC`(Time-varying) method not only demonstrates superior control and safety performance, as previously discussed but is also 85.1% faster than the `cDeeP-LCC` in terms of average computational time. Note that all the CAVs can solve their control inputs individually in the decentralized setting, and thus the computational cost is related to the size of the subsystem and will not increase as the size of the entire system grows. As the number of subsystems increases, the computational cost for `cDeeP-LCC` will keep increasing, while it will remain nearly unchanged for `dDeeP-LCC`. This reveals the scalability of the decentralized approach in terms of both computation and data privacy, which supports real-time implementation for large-scale mixed traffic systems.

VII. CONCLUSIONS

In this paper, we have proposed a decentralized robust data-driven predictive control for CAVs, called `dDeeP-LCC`, to smooth mixed traffic flow. We first decouple a mixed traffic system into several cascading CF-LCC subsystems, where the dynamical coupling is treated as a bounded disturbance. We formulate `dDeeP-LCC` as a robust min-max optimization problem against the worst disturbance, where three estimation methods are discussed for the disturbance bound. We have also developed an efficient computational method to solve the `dDeeP-LCC` for online predictive control. Traffic simulations confirm that `dDeeP-LCC` achieves comparable control performance with centralized `DeeP-LCC` in [22], while providing better safety guarantees, achieving shorter computational time, and naturally preserving data privacy.

Interesting future directions include 1) adapting different regularization strategies and data-preprocessing methods from [40], [41], [46] for further improved performance, and 2) developing direct data-driven methods for nonlinear traffic control (such as [47]). Practical experimental validations similar to [27] and more large-scale numerical experiments are also interesting future topics.

References

[1] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, no. 3, p. 033001, 2008.

[2] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, and H. Zhang, "Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, 2017.

[3] Y. Zheng, S. E. Li, J. Wang, D. Cao, and K. Li, "Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 14–26, 2015.

[4] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296–305, 2013.

[5] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, H. Pohlmann, F. Wu, B. Piccoli *et al.*, "Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205–221, 2018.

[6] Y. Zheng, J. Wang, and K. Li, "Smoothing traffic flow via control of autonomous vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3882–3896, 2020.

[7] K. Li, J. Wang, and Y. Zheng, "Cooperative formation of autonomous vehicles in mixed traffic flow: Beyond platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 951–15 966, 2022.

[8] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Physical Review E*, vol. 51, no. 2, p. 1035, 1995.

[9] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.

[10] I. G. Jin and G. Orosz, "Optimal control of connected vehicle systems with communication delay and driver reaction time," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2056–2070, 2016.

[11] J. Wang, Y. Zheng, Q. Xu, J. Wang, and K. Li, "Controllability analysis and optimal control of mixed traffic flow with human-driven and autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7445–7459, 2021.

[12] M. Di Vaio, G. Fiengo, A. Petrillo, A. Salvi, S. Santini, and M. Tufo, "Cooperative shock waves mitigation in mixed traffic flow environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4339–4353, 2019.

[13] J. Li, J. Wang, S. E. Li, and K. Li, "Learning optimal robust control of connected vehicles in mixed traffic flow," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 1112–1117.

[14] S. Feng, Z. Song, Z. Li, Y. Zhang, and L. Li, "Robust platoon control in mixed traffic flow based on tube model predictive control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 711–722, 2021.

[15] L. Guo and Y. Jia, "Anticipative and predictive control of automated vehicles in communication-constrained connected mixed traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7206–7219, 2021.

[16] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A modular learning framework for mixed autonomy traffic," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 1270–1286, 2021.

[17] E. Vinitsky, K. Parvate, A. Kreidieh, C. Wu, and A. Bayen, "Lagrangian control through deep-RL: Applications to bottleneck decongestion," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 759–765.

[18] W. Gao, Z.-P. Jiang, and K. Ozbay, "Data-driven adaptive optimal control of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1122–1133, 2016.

[19] M. Huang, Z.-P. Jiang, and K. Ozbay, "Learning-based adaptive optimal control for connected vehicles in mixed traffic: robustness to driver reaction time," *IEEE Transactions on Cybernetics*, vol. 52, no. 6, pp. 5267–5277, 2020.

[20] J. Zhan, Z. Ma, and L. Zhang, "Data-driven modeling and distributed predictive control of mixed vehicle platoons," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 572–582, 2022.

[21] J. Lan, D. Zhao, and D. Tian, "Data-driven robust predictive control for mixed vehicle platoons using noisy measurement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6586–6596, 2021.

[22] J. Wang, Y. Zheng, K. Li, and Q. Xu, "DeeP-LCC: Data-enabled predictive leading cruise control in mixed traffic flow," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2760–2776, 2023.

[23] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *2019 18th European Control Conference (ECC)*. IEEE, pp. 307–312.

[24] J. Wang, Y. Zheng, C. Chen, Q. Xu, and K. Li, "Leading cruise control in mixed traffic flow: System modeling, controllability, and string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 861–12 876, 2021.

[25] G. Orosz, "Connected cruise control: modelling, delay effects, and nonlinear behaviour," *Vehicle System Dynamics*, vol. 54, no. 8, pp. 1147–1176, 2016.

[26] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[27] J. Wang, Y. Zheng, J. Dong, C. Chen, M. Cai, K. Li, and Q. Xu, "Implementation and experimental validation of data-driven predictive control for dissipating stop-and-go waves in mixed traffic," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4570–4585, 2023.

[28] F. Gao, S. E. Li, Y. Zheng, and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay," *IET Intelligent Transport Systems*, vol. 10, no. 7, pp. 503–513, 2016.

[29] J. Wang, Y. Lian, Y. Jiang, Q. Xu, K. Li, and C. N. Jones, "Distributed data-driven predictive control for cooperatively smoothing mixed traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 155, p. 104274, 2023.

[30] F. Lyu, H. Zhu, N. Cheng, H. Zhou, W. Xu, M. Li, and X. Shen, "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2586–2602, 2019.

[31] Y. Zhou, S. Ahn, M. Wang, and S. Hoogendoorn, "Stabilizing mixed vehicular platoons with connected automated vehicles: An h-infinity approach," *Transportation Research Part B: Methodological*, vol. 132, pp. 152–170, 2020.

[32] J. Monteil, M. Bouroche, and D. J. Leith, "$\mathcal{L}_2$ and $\mathcal{L}_\infty$ stability analysis of heterogeneous traffic with application to parameter optimization for the control of automated vehicles," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 934–949, 2018.

[33] S. S. Mousavi, S. Bahrami, and A. Kouvelas, "Synthesis of output-feedback controllers for mixed traffic systems in presence of disturbances and uncertainties," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 6, pp. 6450–6462, 2023.

[34] C. Zhao, H. Yu, and T. G. Molnar, "Safety-critical traffic control by connected automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 154, p. 104230, 2023.
[35] L. Huang, J. Coulson, J. Lygeros, and F. Dörfler, "Decentralized data-enabled predictive control for power system oscillation damping," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1065–1077, 2021.
[36] X. Shang, J. Wang, and Y. Zheng, "Smoothing mixed traffic with robust data-driven predictive control for connected and autonomous vehicles," *American Control Conference (ACC) 2024, to appear. Available at arXiv preprint arXiv:2310.00509*.
[37] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011.
[38] J. Löfberg, "Automatic robust convex programming," *Optimization Methods and Software*, vol. 27, no. 1, pp. 115–129, 2012.
[39] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation*. IEEE, pp. 284–289.
[40] F. Dörfler, J. Coulson, and I. Markovsky, "Bridging direct and indirect data-driven control formulations via regularizations and relaxations," *IEEE Transactions on Automatic Control*, vol. 68, no. 2, pp. 883–897, 2022.
[41] X. Shang and Y. Zheng, "Convex approximations for a bi-level formulation of data-enabled predictive control," *arXiv preprint arXiv:2312.15431*, 2023.
[42] J. F. Sturm, "Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999. [Online]. Available: https://doi.org/10.1080/10556789908805766
[43] E. D. Andersen and K. D. Andersen, "The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm," in *High Performance Optimization*. Springer, 2000, pp. 197–232.
[44] DieselNet, "Emission test cycles ece 15 + eudc/nedc," 2013. [Online]. Available: https://dieselnet.com/standards/cycles/ece_eudc.php
[45] D. P. Bowyer, R. Akçelik, and D. Biggs, *Guide to fuel consumption analyses for urban traffic management*, 1985, no. 32.
[46] K. Zhang, Y. Zheng, C. Shang, and Z. Li, "Dimension reduction for efficient data-enabled predictive control," *IEEE Control Systems Letters*, vol. 7, pp. 3277–3282, 2023.
[47] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Linear tracking mpc for nonlinear systems—part ii: The data-driven case," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, 2022.

APPENDIX

In this appendix, we first present the transformation of our decentralized robust `DeeP-LCC` (16) into the robust quadratic optimization problem (21). We then provide standard conic forms of the vertex-based strategy (Method I) (24) and duality-based strategy (Method II) (26) for directly solving the decentralized robust `DeeP-LCC` using solvers, without utilizing time-consuming modeling packages.

A. Quadratic reformulation

We show the process of getting Q, P_1, P_2, d, c_0 and c_1 in (21). The main idea is substituting (20) into (16) and using x to replace variables u_i, σ_{y_i} and $\tilde{\epsilon}_i$.

We first derive the objective function of (21a). After substituting, the objective function of (16) is

$$\begin{aligned}&||u_i||_{R_i}^2 + ||y_i||_{Q_i}^2 + \lambda_{g_i}||g_i||_2^2 + \lambda_{y_i}||\sigma_{y_i}||_2^2 \\ =\ & u_i^\top R_i u_i + b^\top S_1 b + \lambda_{g_i} b^\top S_2 b + \lambda_{y_i}\sigma_{y_i}^\top \sigma_{y_i}.\end{aligned} \tag{30}$$

where $S_1 = H_i^{\dagger\top} Y_{i,\mathrm{F}}^\top Q_i Y_{i,\mathrm{F}} H_i^\dagger$ and $S_2 = H_i^{\dagger\top} H_i^\dagger$ are symmetric positive semidefinite matrix. We denote u_i, σ_{y_i} and b as

$$u_i = F_1 x, \quad \sigma_{y_i} = F_2 x, \quad b = b_{\mathrm{ini}} + F_3 x,$$

where $b_{\mathrm{ini}} = \mathrm{col}(u_{i,\mathrm{ini}}, \epsilon_{i,\mathrm{ini}}, y_{i,\mathrm{ini}}, \mathbb{0}, \mathbb{0})$ and

$$F_1 = \begin{bmatrix} I_N & \mathbb{0} & \mathbb{0}\end{bmatrix}, \qquad F_2 = \begin{bmatrix}\mathbb{0} & I_{(m_i+2)T_{\mathrm{ini}}} & \mathbb{0}\end{bmatrix},$$
$$F_3 = \begin{bmatrix} \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & I_{(m_i+2)T_{\mathrm{ini}}} & \mathbb{0} \\ I_N & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & E_\epsilon \end{bmatrix}.$$

Then (30) could be derived as

$$\begin{aligned}& u_i^\top R_i u_i + b^\top S_1 b + \lambda_{g_i} b^\top S_2 b + \lambda_{y_i}\sigma_{y_i}^\top\sigma_{y_i} \\ =\ & x^\top F_1^\top R_i F_1 x + (b_{\mathrm{ini}} + F_3 x)^\top S_1 (b_{\mathrm{ini}} + F_3 x) \\ & + \lambda_{g_i}(b_{\mathrm{ini}} + F_3 x)^\top S_2 (b_{\mathrm{ini}} + F_3 x) + \lambda_{y_i} x^\top F_2^\top F_2^\top x \\ =\ & x^T (F_1^\top R_i F_1 + F_3^\top S_1 F_3 + \lambda_{g_i} F_3^\top S_2 F_3 + \lambda_{y_i} F_2^\top F_2^\top) x \\ & + 2(b_{\mathrm{ini}}^\top S_1 F_3 + \lambda_{g_i} b_{\mathrm{ini}}^\top S_2 F_3) x + b_{\mathrm{ini}}^\top (S_1 + \lambda_{g_i} S_2) b_{\mathrm{ini}},\end{aligned}$$

so that we can get

$$\begin{aligned} M &= F_1^\top R_i F_1 + F_3^\top S_1 F_3 + \lambda_{g_i} F_3^\top S_2 F_3 + \lambda_{y_i} F_2^\top F_2^\top, \\ d &= 2(F_3^\top S_1 b_{\mathrm{ini}} + \lambda_{g_i} F_3^\top S_2 b_{\mathrm{ini}}), \\ c_0 &= b_{\mathrm{ini}}^\top (S_1 + \lambda_{g_i} S_2) b_{\mathrm{ini}}.\end{aligned}$$

We then show the derivation for constraints. For the safety constraint (21b), after substituting (20b) into (14c) and using the same decomposition of b in the previous part, we have

$$\begin{aligned} & \tilde{s}_{\min} \le G_1 Y_{i,\mathrm{F}} H_i^\dagger (b_{\mathrm{ini}} + F_3 x) \le \tilde{s}_{\max} \\ \Leftrightarrow\ & \tilde{s}_{\min} \le G_1 Y_{i,\mathrm{F}} H_i^\dagger F_3 x + G_1 Y_{i,\mathrm{F}} H_i^\dagger b_{\mathrm{ini}} \le \tilde{s}_{\max}.\end{aligned}$$

Thus, we have

$$P_1 = G_1 Y_{i,\mathrm{F}} H_i^\dagger F_3, \quad c_1 = G_1 Y_{i,\mathrm{F}} H_i^\dagger b_{\mathrm{ini}}.$$

For the input limitation (21c), we need to extract u_i from x and we have $P_2 = F_1$.

B. Conic form transformation

We show how to change (24) and (26) to their conic form and get corresponding $A, b, c, \mathcal{K}$. The key point is to derive each constraint to its corresponding conic inequality. We use subscript "s", "o", "z" to represent parameters corresponding to the second-order cone, non-negative orthant and zero cone. Recall that x_j denotes fixing $\tilde{\epsilon}_i$ in x with one of extreme points w_j and x_{d} is defined as $\mathrm{col}(u_i, \sigma_{y_i})$.

1) Method I: For Method I, the decision variable y is $\mathrm{col}(t, u_i, \sigma_{y_i})$ and it is obvious that $b = \mathrm{col}(1, \mathbb{0}, \mathbb{0})$ for the objective function. There are two types of conic inequalities at the constraints which are second-order cone and non-negative orthant.

We first derive constraint (24a) to its second order cone form. It is normal to transform the convex quadratic constraints to the following form

$$\left\| \begin{bmatrix} 2\Gamma x_j \\ t - d^T x_j - 1 \end{bmatrix} \right\|_2 \le t - d^T x_j + 1, \; j = 1, 2, \ldots n_{\mathrm{v}}, \tag{31}$$

where Γ is gotten from decomposing M as $\Gamma^\top\Gamma$ using the fact that M is a positive semidefinite matrix. We further divide Γ and d as $\Gamma = \begin{bmatrix}\Gamma_{\mathrm{d}}, \Gamma_w\end{bmatrix}$ and $d = \mathrm{col}(d_{\mathrm{d}}, d_w)$ which corresponds to the multiplication with x_d and w_j. Then (31) can be derived as the second order cone form

$$\begin{bmatrix} t - d_{\mathrm{d}}^{\mathsf{T}} x_d - d_w^{\mathsf{T}} w_j + 1 \\ 2(\Gamma_d x_d + \Gamma_w w_j) \\ t - d_{\mathrm{d}}^{\mathsf{T}} x_d - d_w^{\mathsf{T}} w_j - 1 \end{bmatrix} \in \mathcal{K}_{\mathrm{s}j}, \quad j = 1, \dots, n_{\mathrm{v}},$$

and the corresponding $A_{\mathrm{s}j}$ and $c_{\mathrm{s}j}$ are

$$A_{\mathrm{s}j}^{\mathsf{T}} = \begin{bmatrix} -1 & d_{\mathrm{d}}^{\mathsf{T}} \\ \mathbb{0} & -2\Gamma_{\mathrm{d}} \\ -1 & d_{\mathrm{d}}^{\mathsf{T}} \end{bmatrix}, \; c_{\mathrm{s}j} = \begin{bmatrix} -d_w^{\mathsf{T}} w_j + 1 \\ 2\Gamma_w w_j \\ -d_w^{\mathsf{T}} w_j - 1 \end{bmatrix}, \; j = 1, \dots, n_{\mathrm{v}}.$$

We then derive constraints (24b) and (24c) to their non-negative orthant form. For constraint (24b), we could represent P_1 as $[P_{1\mathrm{d}}, P_{1w}]$ and each evaluation of (24b) in w_j can be derived as

$$\begin{bmatrix} s_{\max} - P_{1\mathrm{d}} x_{\mathrm{d}} - P_{1w} w_j - c_1 \\ -s_{\min} + P_{1\mathrm{d}} x_{\mathrm{d}} + P_{1w} w_j + c_1 \end{bmatrix} \in \mathcal{K}_{\mathrm{o}j},$$

and the corresponding $A_{\mathrm{o}j}$ and $c_{\mathrm{o}j}$ for $j = 1, \dots, n_{\mathrm{v}}$ are

$$A_{\mathrm{o}j}^{\mathsf{T}} = \begin{bmatrix} \mathbb{0} & P_{1\mathrm{d}} \\ \mathbb{0} & -P_{1\mathrm{d}} \end{bmatrix}, \quad c_{\mathrm{o}j} = \begin{bmatrix} s_{\max} - P_{1w} w_j - c_1 \\ -s_{\min} + P_{1w} w_j + c_1 \end{bmatrix}.$$

For the constraint (24c), similar to derivation of (24b), we have $\begin{bmatrix} u_{\max} - u_i \\ u_i - u_{\min} \end{bmatrix} \in \mathcal{K}_{\mathrm{o}u}$ and the corresponding $A_{\mathrm{o}u}$ and $c_{\mathrm{o}u}$ are

$$A_{\mathrm{o}u}^{\mathsf{T}} = \begin{bmatrix} \mathbb{0} & I & \mathbb{0} \\ \mathbb{0} & -I & \mathbb{0} \end{bmatrix}, \quad c_{\mathrm{o}u} = \begin{bmatrix} u_{\max} \\ -u_{\min} \end{bmatrix}.$$

The final form of A and c for (24) are gotten from stacking matrices derived above and we have

$$A = \mathrm{col}(A_{\mathrm{o}u}, A_{\mathrm{o}1}, \dots, A_{\mathrm{o}n_{\mathrm{v}}}, A_{\mathrm{s}1}, \dots, A_{\mathrm{s}n_{\mathrm{v}}}),$$
$$c = \mathrm{col}(c_{\mathrm{o}u}, c_{\mathrm{o}1}, \dots, c_{\mathrm{o}n_{\mathrm{v}}}, c_{\mathrm{s}1}, \dots, c_{\mathrm{s}n_{\mathrm{v}}}).$$

Cone $\mathcal{K}$ is the product of $n_{\mathrm{v}} + 1$ non-negative orthants and n_{v} second-order cone which is

$$\mathcal{K} = \mathcal{K}_{\mathrm{o}u} \times \mathcal{K}_{\mathrm{o}1} \times \cdots \mathcal{K}_{\mathrm{o}n_{\mathrm{v}}} \times \mathcal{K}_{\mathrm{s}1} \times \cdots \times \mathcal{K}_{\mathrm{s}n_{\mathrm{v}}}.$$

2) Method II: For Method II, the decision variable is $y = \mathrm{col}(t_i, u_i, \sigma_{y_i}, \lambda_1, \lambda_2)$ and b in the objective function is $\mathrm{col}(1, \mathbb{0}, \mathbb{0}, \mathbb{0}, \mathbb{0})$. There are three types of conic inequalities at the constraints which are second-order cone, non-negative orthant and zero cone.

We omit the detailed derivation for the same constraints (24a) and (24b) in (26). We only need to add an extra zero block column at the right side of each $A_{\mathrm{s}j}$ and $A_{\mathrm{o}u}$ because of additional decision variables λ_1 and λ_2. Forms of $A_{\mathrm{s}j}$ for $j = 1, \dots, n_{\mathrm{v}}$ and $A_{\mathrm{o}u}$ can be represented as

$$A_{\mathrm{s}j}^{\mathsf{T}} = \begin{bmatrix} -1 & d_{\mathrm{d}}^{\mathsf{T}} & \mathbb{0} \\ \mathbb{0} & -2\Gamma_{\mathrm{d}} & \mathbb{0} \\ -1 & d_{\mathrm{d}}^{\mathsf{T}} & \mathbb{0} \end{bmatrix}, \quad A_{\mathrm{o}u}^{\mathsf{T}} = \begin{bmatrix} \mathbb{0} & I & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & -I & \mathbb{0} & \mathbb{0} \end{bmatrix}.$$

We will have the same $\mathcal{K}_{\mathrm{s}j}, c_{\mathrm{s}j}, j = 1, \dots, n_{\mathrm{v}}$ and $\mathcal{K}_{\mathrm{s}u}, c_{\mathrm{o}u}$ as in Method I.

We then present the conic derivation for (26a)-(26e). For (26a), (26c), (26e), they could be represented by non-negative orthants which are

$$\begin{bmatrix} \tilde{s}_{\max} - p_{l,\mathrm{d}}^{\mathsf{T}} x_{\mathrm{d}} - \tilde{b}_\epsilon^{\mathsf{T}} \lambda_{l,1} - c_{1,l} \\ -\tilde{s}_{\min} + p_{l,\mathrm{d}}^{\mathsf{T}} x_{\mathrm{d}} - \tilde{b}_\epsilon^{\mathsf{T}} \lambda_{l,2} + c_{1,l} \end{bmatrix} \in \mathcal{K}_{\mathrm{o}l}, \quad l = 1, \dots, N,$$

and $\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \in \mathcal{K}_{\mathrm{o}\lambda}$. The corresponding $A_{\mathrm{o}l}$ and $c_{\mathrm{o}l}$ for $\mathcal{K}_{\mathrm{o}l}$ are

$$A_{\mathrm{o}l}^{\mathsf{T}} = \begin{bmatrix} 0 & p_{l,d}^{\mathsf{T}} & \mathbb{0}_{1 \times 2n_\epsilon(l-1)} & \tilde{b}_\epsilon^{\mathsf{T}} & \mathbb{0} \\ 0 & -p_{l,d}^{\mathsf{T}} & \mathbb{0}_{1 \times 2n_\epsilon(N+l-1)} & \tilde{b}_\epsilon^{\mathsf{T}} & \mathbb{0} \end{bmatrix},$$
$$c_{\mathrm{o}l}^{\mathsf{T}} = \begin{bmatrix} \tilde{s}_{\max} - c_{1,l} \\ -\tilde{s}_{\min} - c_{1,l} \end{bmatrix}, \quad l = 1, \dots, N,$$

and the corresponding $A_{\mathrm{o}\lambda}$ and $c_{\mathrm{o}\lambda}$ for $\mathcal{K}_{\mathrm{o}\lambda}$ are

$$A_{\mathrm{o}\lambda}^{\mathsf{T}} = \begin{bmatrix} \mathbb{0} & -I & \mathbb{0} \end{bmatrix}, \quad c_{\mathrm{o}\lambda} = \mathbb{0}.$$

For (26b) and (26d), they can be considered as zero cones which are

$$\begin{bmatrix} \tilde{A}_\epsilon^{\mathsf{T}} \lambda_{l,1} - p_{l,\epsilon} \\ \tilde{A}_\epsilon^{\mathsf{T}} \lambda_{l,2} + p_{l,\epsilon} \end{bmatrix} \in \mathcal{K}_{zl}, \quad l = 1, \dots, N,$$

and $A_{\mathrm{z}l}$ and $c_{\mathrm{z}l}$ correspond to them are

$$A_{\mathrm{z}l}^{\mathsf{T}} = \begin{bmatrix} \mathbb{0} & \mathbb{0}_{n_\epsilon \times 2n_\epsilon(l-1)} & \tilde{A}_\epsilon^{\mathsf{T}} & \mathbb{0} \\ \mathbb{0} & \mathbb{0}_{n_\epsilon \times 2n_\epsilon(N+l-1)} & \tilde{A}_\epsilon^{\mathsf{T}} & \mathbb{0} \end{bmatrix}, \quad c_{\mathrm{z}l} = \begin{bmatrix} p_{l,\epsilon} \\ -p_{l,\epsilon} \end{bmatrix}.$$

The final form of A and c for (26) could be represented as

$$A = \mathrm{col}(A_{\mathrm{z}1}, \dots, A_{\mathrm{z}N}, A_{\mathrm{o}\lambda}, A_{\mathrm{o}u}, A_{\mathrm{o}1}, \dots, A_{\mathrm{o}N}, A_{\mathrm{s}1}, \dots, A_{\mathrm{s}n_{\mathrm{v}}}),$$
$$c = \mathrm{col}(c_{\mathrm{z}1}, \dots, c_{\mathrm{z}N}, c_{\mathrm{o}\lambda}, c_{\mathrm{o}u}, c_{\mathrm{o}1}, \dots, c_{\mathrm{o}N}, c_{\mathrm{s}1}, \dots, c_{\mathrm{s}n_{\mathrm{v}}}).$$

Cone $\mathcal{K}$ is the product of N zero cones, $N + 2$ non-negative orthants and n_v second-order cones, which is

$$\mathcal{K} = \mathcal{K}_{\mathrm{z}1} \times \cdots \times \mathcal{K}_{\mathrm{z}N} \times \mathcal{K}_{\mathrm{o}\lambda} \times \mathcal{K}_{\mathrm{o}u} \times \mathcal{K}_{\mathrm{o}1} \times \cdots \times \mathcal{K}_{\mathrm{o}N} \times \mathcal{K}_{\mathrm{s}1} \times \cdots \times \mathcal{K}_{\mathrm{s}n_{\mathrm{v}}}.$$