Multi-Scale Signal-Vehicle Coupled Control with Connected and Autonomous Vehicles

Qiangqiang Guo

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Xuegang Ban, Chair

Yinhai Wang

Husain Aziz

Program Authorized to Offer Degree:

Civil and Environmental Engineering

University of Washington

Abstract

Multi-Scale Signal-Vehicle Coupled Control with Connected and Autonomous Vehicles

Qiangqiang Guo

Chair of the Supervisory Committee:
Xuegang Ban
Department of Civil and Environmental Engineering

With the rise of connected and autonomous vehicles (CAV) and vehicle-to-everything communication technologies, advanced urban traffic control (UTC) has been receiving increasing attention due to the abundant data and control freedom brought by the new techniques. The UTC is a multi-scale problem in both spatial and temporal domains. The goal of this study is to investigate the multi-scale UTC problems in the environment of CAVs. Firstly, we propose a general modeling and control framework for the multi-scale UTC problem and identify the key research needs. We then use this framework to explicitly model, analyze, and solve the two-scale signal-vehicle coupled control problem in the environment of single intersection with 100% CAV penetration, and demonstrate the performance of the proposed method using numerical experiments. Then, we extend the signal-vehicle coupled control algorithm to the environment of multiple intersections with mixed traffic flows of human driven vehicles and CAVs. Considering that for explicit model based methods, the computation burden increases with the size of the traffic network, we develop an imitation learning enhanced method to improve the algorithm efficiency. We conduct various numerical experiments to validate the performance of the proposed model-based and learning-enhanced algorithms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

AI: Artificial Intelligence

AV: Autonomous Vehicle

CACC: Cooperative Adaptive Cruise Control

CAV: Connected and Autonomous Vehicle

CNN: Convolutional Neural Network

CTM: Cell Transmission Model

CV: Connected Vehicle

DMD: Dynamic Mode Decomposition

DP: Dynamic Programming

DTA: Dynamic Traffic Assignment

EV: Electrical Vehicle

HDV: Human Driven Vehicle

HEV: Hybrid Electrical Vehicle

ICE: Internal Combustion Engine

IDM: Intelligent Driver Model

IL: Imitation Learning

ILP: Integer Linear Programming

INLP: Integer Non-linear Programming

ITS: Intelligent Transportation System

MFD: Macroscopic Fundamental Diagram

MILP: Mixed Integer Liner Programming

MINLP: Mixed Integer Non-liner Programming

ML: Machine Learning

MPC: Model Predictive Control

NLP: Non-linear Programming

RL: Reinforcement Learning

SVCC: Signal-Vehicle Coupled Control

UTC: Urban Traffic Control

V2I: Vehicle-to-Infrastructure

V2V: Vehicle-to-Vehicle

V2X: Vehicle-to-everything

ACKNOWLEDGMENTS

First, I would like to express my gratitude to my advisor and chair of committee Dr. Xuegang (Jeff) Ban for his invaluable guidance and consistent support during my PhD journey. I would also like to thank all my committee members, who generously provided knowledge and expertise to help improve this study.

I am also grateful to my lab mates in the intelligent urban transportation laboratory, for their constructive research feedback and moral support. Thanks should also go to the Department of Civil and Environmental Engineering and all its staff for all the considerate help and guidance.

Last but not least, many thanks and deep appreciation to my family, especially my wife Xinping Yuan, who believed in me and brought me the pleasure of life and hope for the future. I would also like to thank our cat Rocky and dog Coco for all the entertainment and emotional support.

Chapter 1

INTRODUCTION

1.1 Research motivation

Urbanization has incented dramatic growth of car usage in more and more cities globally. Motor vehicle miles have increased by 106% in the United States from 1990 to 2020 (United States Department of Transportation, USDOT). As a result, growing traffic congestion, accidents, and pollution are threatening sustainable mobility, especially in urban area. Various traffic management methods have been proposed to handle the fast-growing travel demands. For example, many cities are building better public transportation systems and launching low-price tickets to reduce private car usage (Nuzzolo and Comi [2016]). Ride-sharing is also advocated by many local governments to reduce car ownership (Dong et al. [2018], Nie [2017]). Among those traffic management methods, designing better urban traffic control (UTC) systems has always been receiving great attention due to the strong ability to improve urban traffic performance without building new infrastructures and disrupting people's travel habits (Hamilton et al. [2013], Li and Wang [2018]).

Urban transportation system is multi-modal, multi-scale in both spatial and temporal domains. Even when we limit ourselves to the control of vehicular traffic, as shown in Figure 1.1, the multi-scale nature of UTC is evident as it contains traffic control at multiple scales: from vehicle scale control to intersection scale control, corridor scale and sub-network (regional) scale control, and to network (global) scale control. Traffic at each scale has its own behavior (i.e., "dynamics"), while interacting with traffic at other scales, especially the "neighboring" scales (e.g., intersection control interacts with the dynamics at the vehicle scale and corridor scale more than those at other scales). Furthermore, each scale has its specific temporal and spatial scales when control is concerned. Using the temporal scale as an

example: vehicle control often occurs in sub-second or even milliseconds when safety-critical applications are considered (e.g., collision avoidance), intersection and corridor control run in seconds, while sub-network and regional control can happen in minutes.

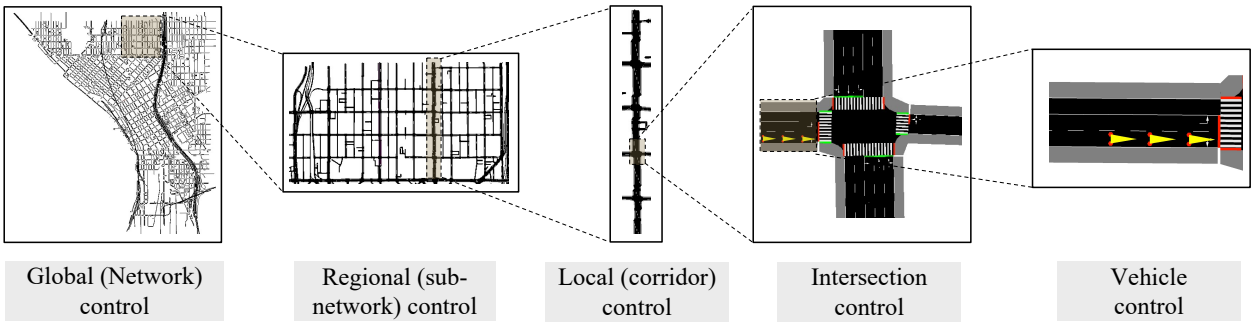

Figure 1.1: The multi-scale UTC system

Such a multi-scale system is extremely complex to model, analyze, and solve. While the multi-scale nature (or the so-called "hierarchical" or "multiresolution" feature) of UTC has long been recognized (Mirchandani et al. [2018]), UTC has been mainly studied separately for each individual scale, with sparse and limited integrative investigations on multi-scale UTC (e.g., signal timing optimization of both corridors and intersections, signal-vehicle co-optimization, etc.; Xu et al. [2018], Li and Ban [2017], Ge et al. [2019]). There are two main reasons for this. First, multi-scale control is a mathematically challenging problem, and to this day, there is no systematic theory, model, or algorithm that can properly analyze and efficiently solve a general multi-scale control system, despite some recent advances (e.g., Chapman and Mesbahi [2016]) and increasing applications in various domains (e.g., biology De et al. [2015]). This prevents one from adopting the multi-scale control framework to model and study the multi-scale UTC. Second, for long, the data and information transmission between different scales are quite limited such that the interactions with neighboring scales are often ignored or simplified. For example, when optimizing the signal timing plan at an intersection, only vehicle arrivals are detected and all vehicles can then be simplified as

homogeneous vehicles (or different groups of homogeneous vehicles). When effective methods to collect individual characteristics of different vehicles (large v.s. small, gasoline v.s. hybrid or electric) are lacking, albeit important for traffic signal optimization, such detailed characteristics and the more dynamic interactions between vehicles and the traffic signal have to be substantially simplified or even ignored in traditional modeling methods.

The emergence of new technologies in transportation, especially the connected and autonomous vehicles (CAVs) and "vehicle to everything" (V2X) communications, have changed and will continue to change profoundly the dynamics and interactions among different scales in urban traffic, which warrants a revisit to this multi-scale UTC framework. Usually, connected vehicles (CVs) refer to vehicles that can communicate with other vehicles (vehicle-to-vehicle, V2V), infrastructure (vehicle-to-infrastructure, V2I), and other traffic participants such as pedestrians and bicyclists. For autonomous vehicles (AVs), the U.S. Department of Transportation's National Highway Traffic Safety Administration (NHTSA) defined five levels of automation, from driving assistance (Level 1) to fully autonomous (Level 5). Fully autonomous vehicles refer to "the vehicle can do all the driving in all circumstances. The human occupants are just passengers and need never be involved in driving", which is also the definition of AVs discussed in this study. As shown in later sections, we aim to improve the vehicle performance by controlling the longitudinal accelerations. To this end, Level 2 automation is enough. Higher automation level could provide us with more reliable control performance.

With the help of CAVs and V2X technologies, the data and information collected by vehicles and the infrastructure, as well as those transmitted among vehicles, infrastructures and other transportation users (bicyclists, pedestrians, etc.), are now massive and unprecedented. This makes it possible to reveal not only the detailed dynamics of agents within a scale (e.g., vehicle scale), but also the dynamics and interactions across different scales (e.g., vehicles and the signals). That is, with the new technologies, agents at different scales can now "see" (perceive) not only other agents in the same scale, but also agents at other scales, collectively forming a true integrative multi-scale system. In addition, the new technologies

also bring more control freedoms to the urban transportation system. For example, AVs can be controlled more intelligently to improve mobility and save energy.

As will be shown in the literature review (i.e., Section 2), current studies on UTC usually focus on single scale, e.g., traffic signal control (Li and Ban [2018], Chen and Sun [2016], Feng et al. [2015]), vehicle control (Li et al. [2012], Rosolia et al. [2016], Kritayakirana and Gerdes [2012]), perimeter regional control (Guo and Ban [2020], Haddad and Mirkin [2016], Aalipour et al. [2018]). There are recent research efforts to apply the concept of multi-scale modeling framework for UTC. For examples, Xu et al. [2018] proposed a cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections. They calculated the optimal signal phases and the arrival times of AVs at a higher level to minimize the travel time of all vehicles, and optimized the engine power and brake force to minimize the fuel consumption at the lower level. Yang et al. [2021] extended the above idea to the corridor scale with more practical considerations such as the platooning behavior and the mixed traffic flow with both AVs and human driven vehicles (HDVs). However, there are several limitations of current studies. First, while focusing on practical multi-scale control frameworks that may be deployable in the near future, most current research works do not formally recognize the multi-scale feature, not to mention providing formal multi-scale frameworks for UTC. Thus, current studies lack methods or tools to analyze the proposed model and address critical issues such as the control stability. Second, current studies usually assume full CAV penetration. And the studies that do consider mixed traffic flow are usually tailored for single intersection or corridors. Network-wide multi-scale UTC under the mixed traffic flow condition is rare. Third, current studies usually adopt explicit models to formulate and solve the multi-scale UTC problem. The models could be very complex and suffer from low computation efficiency.

1.2 Research objectives and contributions

The goal of this study is to investigate the multi-scale UTC problem under the environment of CAVs. We firstly propose a general modeling and control framework for the multi-scale UTC

problem and identify the key research needs and potential research directions. Then, we show how to adopt such a framework to explicitly model, analyze, and solve the two-scale signal-vehicle coupled control (SVCC) problem under the environment of single intersection with 100% CAV penetration, and demonstrate the performance of the proposed framework using numerical experiments. We then extend the two-scale SVCC algorithm to the environment of multiple intersections and mixed traffic flows of HDVs and CAVs. Finally, we propose an imitation learning (IL) enhanced method to improve the computation efficiency of the explicit model based methods.

Major contributions of this study include:

1. Development of a general modeling and control framework for the multi-scale UTC problem, which can provide methods and tools to analyze the multi-scale model and address critical issues such as the stability of the control model and computational methods.

2. Development of a two-scale SVCC algorithm for single intersection under 100% CAV penetration. Development of an integrated model, an MPC-based decomposition and control method, and a stability analysis scheme for the SVCC problem.

3. Development of a two-scale SVCC framework under the environment of multiple intersections and mixed traffic flows of HDVs and CAVs. Development of an information-sharing strategy for multiple intersections and a "estimation-and-safety-check" method for the mixed traffic flows.

4. Development of an IL-enhanced algorithm to mimic the control strategy generated by the explicit model based methods, which can improve the computation efficiency and meanwhile maintain similar control performance.

1.3 Dissertation outline

In the following of this dissertation, we firstly give a thorough literature review of current progresses on the advanced UTC methods under CAVs (Section 2.1), and in particular, the multi-scale UTC (Section 2.2). Then, in Section 3, we present an optimal control based general framework for the multi-scale UTC, and identify the key research needs. Based on the general framework, we show how to model and solve the two-scale SVCC problem for single intersection with 100% CAV penetration in Section 4. We start from building an integrated SVCC problem (Section 4.2), then decompose it into two sub-problems based on an MPC scheme (Section 4.3), where we also analyze the stability of the proposed MPC scheme. We validate the effectiveness of the proposed algorithm in Section 4.4 using traffic simulation. In Section 5, we show how to extend the algorithm of Section 4 to the environment of multiple intersections (Section 5.2.1) and mixed traffic flow (Section 5.2.2). We then combine the two extensions as an integrated algorithm (Section 5.2.3) and demonstrate its performance by numerical experiments (Section 5.3). After that, we show how to design and implement a IL-enhanced algorithm to reduce the computation burden and improve the algorithm efficiency (Section 6.1). We also demonstrate the performance of the IL-enhanced algorithm using numerical experiments (Section 6.2). In addition, we discuss some practical considerations for real-world implementation and future research directions in Section 7. Finally, we conclude this dissertation in Section 8.

Chapter 2
LITERATURE REVIEW

UTC has been studied extensively for a long time. Previous studies usually focus on traffic signal control, from traditional methods such as fixed-time signal control to more advanced methods such as actuated and adaptive signal control (Roess et al. [2004]). The rise of CAVs provides more observability and controllability to the UTC system. In the first part of this section, i.e., Section 2.1, we give a thorough review about those advanced UTC techniques with CAVs.

We will see that most current advanced UTC techniques reviewed in Section 2.1 study the UTC problem from single scale (either in time or spatial domain). However, the UTC is naturally a multi-scale problem in both temporal and spatial domains. This study focuses on multi-scale UTC under CAVs. Thus, in the second part of this section, i.e., Section 2.2, we review current methodologies and research progresses regarding the multi-scale systems and in particular, the multi-scale UTC problem.

2.1 Advanced UTC with CAVs

In this section, we review advanced UTC technologies under the environment of CAVs. CAVs bring three advantages to UTC: First, the information of CAVs can help the traffic signal control system better understand the current traffic condition as well as predict future traffic conditions, thus design better signal plans to improve the traffic performance. Second, enabled by the V2I techniques, drivers could improve their operations (e.g., by following the suggestions provided by on-board computers or road-side units). And AVs could operate more efficiently because they can be fully controlled. Third, traffic signals and vehicles can cooperate with each other to improve the performance of both the overall traffic and

individual vehicles. Therefore, we categorize the advanced UTC methods with CAVs to three types: advanced traffic signal control, advanced traffic-oriented vehicle control, and SVCC. Considering that SVCC is usually modeled under a multi-scale scheme, we put the review of this topic to the second section (i.e., Section 2.2). Note that the review of this section is largely from Guo et al. [2019a].

2.1.1 Advanced traffic signal control with CAVs

Based on the control schemes, we can further divide advanced CAV-based traffic signal control into three types: actuated traffic signal enhanced by CAVs data, platoon-based traffic signal control based on CAVs coordination, and planning-based traffic signal control based on CAVs coordination. The key difference among these three control methods is in what detail they predict the future traffic states. Enhanced by CAVs data, actuated (adaptive) traffic signal control would estimate the current traffic states (e.g., queue length) and predict some relatively rough and aggregated traffic measures (e.g., average volume) in the future, based on which the control decisions (e.g., extend or terminate certain phases) could be generated. The platoon-based signal control would simplify the problem by categorizing individual vehicles to platoons and predict their arrivals/trajectories, which could make it easier to adjust the timing plan. The planning-based signal control would take the detailed trajectories of individual vehicles into account and optimize the signal timing/phases in a forward time horizon by adopting more accurate and complex models. The traffic flow dynamic models and control strategies usually become more exquisite when the control methods come from actuated to planning-based, while on the other hand, the robustness and flexibility would also decrease.

The CAV-based traffic signal control methods are usually designed first for isolated intersections, and then extended to corridors, and even networks. Compared with the network-wide traffic signal control, there are more studies on single intersection and corridor signal control. In this section, we firstly review studies on the three types of CAV-based traffic signal control methods, mainly focusing on the single intersection and corridor scenarios.

Then, we use a separate section to review the studies on network-wide traffic signal control with CAVs.

Actuated traffic signal control enhanced by CAVs data

Actuated signal control can dynamically adjust the timing parameters to respond to real-time traffic arrival changes. Since many existing studies consider prediction of traffic flow based on CAVs data, this kind of control is also called adaptive traffic signal control in the literature. This generally results in more efficient utilization of intersection capacity than fixed-time signal control in which signal phases and cycle lengths are pre-selected based on historical traffic patterns (Roess et al. [2004], Zhang and Wang [2010]).

Conventional actuated traffic signal control systems collect traffic information via inductive loop detectors that are usually installed tens of meters upstream to the stop lines. The obtained information is inaccurate and limited spatially. As a result, certain relatively rough models have been developed to describe traffic flow states which often fail to well present the variability in traffic demand and vehicular inter-arrival times (Yin et al. [2007], Yun and Park [2012], Zheng et al. [2010]). CAVs provides a remedy for such problems (Day and Bullock [2016], Goodall et al. [2013], Gradinescu et al. [2007], Kari et al. [2014], Li et al. [2014], Younes and Boukerche [2015]). Based on the accurate position information of the arriving vehicles, we can either extend/shorten the current phase or add an extra phase to make on-time changes.

For example, Gradinescu et al. [2007] proposed an actuated traffic signal control system based on CVs. The information of vehicles within a few miles range around an intersection were collected by the V2I and infrastructure to infrastructure (I2I) communication techniques. They firstly used these data to estimate the demand volume of each approach per cycle, and then calculated the optimum cycle length using the Webster's formula (Chaudhary et al. [2002]). The green time was allocated to produce equal degrees of saturation on each link. The preliminary signal plan for the next cycle was generated during the current cycle and adjusted to meet practical limitations like the minimum and maximum cycle lengths

and pedestrian minimum green times. They tested the system in simulation based on two real-world major intersections. The experiment results showed that the system could reduce traffic delay and fuel consumption compared with traditional pre-timed traffic signals. In this work, the estimated demand volume is the state variable and the information of vehicles around the intersection serve as the environment inputs. The objective is the equality of the degrees of saturation on each link, which is maximized by generating optimum cycle length and allocating green times proportionally.

The CAV-based actuated traffic signal control is essentially a passive method since it adjusts the signal plan according to the estimated traffic states without detailed predictions of future traffic conditions. From the optimal control point of view, the strategies generated by the actuated control may not be optimal in long-term since future traffic conditions are not considered. Compared with other methods, the main advantage of actuated control is that its computation burden is relatively light due to the smaller number of control variables as well as the simpler traffic models. The fixed sensor based actuated signal control has been widely used in real-world traffic management. This makes it more practical to implement the CAV-based actuated signal control in practice compared to other advanced control methods.

Platoon-based traffic signal control based on CAVs coordination

Actuated traffic control relies more on prevailing real-time traffic information and does not require too much future traffic conditions. In contrast, traffic prediction is essential to platoon-based traffic signal control (and also planning-based control). By identifying the platoons (or categorizing individual vehicles into pseudo platoons) and predicting their arrival time in advance, the platoon-based signal control aims to schedule the signal timing plans to allow the platoons to pass the intersections without severe interruptions, which can increase the overall traffic efficiency. Although the idea of platoon-based traffic signal control has been proposed for several decades (Mirchandani and Head [2001]), it became realistic only after V2X technique was introduced (He et al. [2012], Liang et al. [2018], Lioris et al. [2017], Xie et al. [2012]). This is because V2X makes it possible to properly identify platoons so

that platoon-based optimal signal timing plans can be generated accordingly.

Pandit et al. [2013] proposed a platoon-based "oldest arrival first" traffic signal control method to reduce delays at a single intersection. They reduced the traffic signal control problem to a job scheduling problem by enforcing that all jobs require equal processing time. The conflicts between jobs and the objective of minimizing job latency values were modeled as a two-competitive algorithm. After collecting real-time speed and position information of sample vehicles through vehicular ad-hoc networks, they grouped the vehicular traffic into approximately equal-sized platoons by searching all possible platoon configurations to minimize the difference between the maximum and minimum required processing time. The grouped platoons could be scheduled by solving the reduced job scheduling problem. The algorithm was tested under different approach arrival rates and penetration rates. Compared with traditional Webster's method and vehicle-actuated control method, the proposed method could significantly reduce delays when the traffic inflow rates are not large. The experimental results also showed that the proposed method did not perform well under low penetration rates, since the arrival rate cannot be accurately estimated under low penetration conditions. In this work, the speed and position information of sample vehicles serve as the environment input, which are further processed to be the arriving platoons. The state variable is the travel time of the vehicle platoons and the control variable is the signal timing and phases. The objective is the total delay at the intersection experienced by all vehicles, which is minimized by adopting the optimal control variables generated by the "oldest arrival first" method.

While Pandit et al. [2013] focused on a single intersection, He et al. [2012] presented a platoon-based multi-modal dynamical progression model to control arterial traffic signals. Sample vehicle data were firstly used to identify existing queues and platoons approaching each intersection by a headway-based recognition algorithm. Then, they formulated the traffic signal control problem into a mixed-integer linear program (MILP) based on the calculated platoon information, current signal status, and priority requests of special vehicles. By using platoon data instead of individual vehicle data, they reduced the number of integer

variables of the MILP, making it relatively easier to solve. Their model can deal with both under-saturated and over-saturated traffic conditions. The proposed method was tested in VISSIM simulations. Results showed that, under a 40% penetration rate, the method could reduce the overall average delay of two traffic modes (i.e., automobiles and transit buses) by 8% compared to the coordinated actuated signal control method optimized by SYNCHRO.

Compared with actuated signal control, platoon-based signal control using CAV data could achieve better performance since it could forecast some mid-level traffic flow states (i.e., the volume and arrival time of platoons) and make the best control decisions accordingly. Meanwhile, aggregating vehicles into platoons could reduce the computation burden, making it more practical to be implemented in the real world. However, the method may only generate sub-optimal strategies due to this simplification. Besides, the performance of platoon identification algorithms may significantly affect the performance of the method.

Planning-based traffic signal control based on CAVs coordination

Platoon-based methods categorize the incoming vehicles as platoons and ignore the inner dynamics and disturbances among vehicles in the same platoon. On the contrary, planning-based methods treat all vehicles at the same level, which can better describe the real traffic condition. Besides, platoon-based methods usually directly assume known arrival distributions (e.g., Poisson or uniform arrivals), or estimate the arrival time of the platoons and assume uniform arrivals within each platoon. Planning-based methods often estimate the actual arrival time of every vehicle and predict traffic conditions in a forward time horizon.

Planning-based control has been widely studied by many researchers (Goodall et al. [2013], Lee et al. [2013]). The optimization model of planning-based control is usually an integer non-linear programming (INLP) problem and hard to solve especially when individual vehicles' trajectories are considered (Li and Ban [2018]). Certain approximation and reformulation methods are usually applied. Dynamic programming (DP) is one of the most commonly used techniques to reformulate and solve such control problems (Chen and Sun [2016], Feng et al. [2015], Sen and Head [1997]). For example, Feng et al. [2015] proposed an

optimization-based real-time traffic signal control method in a CV environment. Assuming known vehicles' speeds and positions, they firstly separated the upstream roads into three regions (i.e., free-flow, slow-down, and queuing) and estimated the status of unequipped vehicles. Based on such information, they constructed a complete predicted arrival table for each phase for a certain (future) time horizon. Then they built a two-level optimization model. The upper level generates the minimum and maximum allowable barrier group lengths by DP, and the lower level is formulated as a utility minimization problem with two alternative objectives (i.e., minimizing total vehicle delay and queue length respectively). The outputs of the model are the optimal signal timing and phases. They tested the method by modeling a real-world intersection in VISSIM. The results showed that the method can reduce the total delay by 16.33% under high penetration rate compared to the fully actuated control method, which however generated the same delay under low penetration rates. In order to make the DP method more practical and efficient, Li and Ban [2017] proposed a DP based method to minimize both fuel consumption and travel time considering a fixed cycle length. They firstly formulated the signal control problem as a mixed integer non-linear programming problem, and then reformulated the problem as a DP model by dividing the timing decisions into stages (one stage for a phase) and approximating the fuel consumption and travel time of a stage as functions of the state and decision variables of that stage. By adding the end-stage cost and a branch and bound regulator to the DP formulation, the resulting optimal solution can be guaranteed to lead to the fixed cycle length. Simulation results showed that the proposed method could generate optimal solutions that lead to the fixed cycle length. The control performance was improved compared to the actuated control methods and was similar to the results by a global mixed integer non-linear programming (MINLP) solver in MATLAB. In this work, the objective is the weighted summation of total system travel times and fuel consumption. The state variables are the vehicle trajectories which are modeled by the intelligent driver model (IDM, Treiber et al. [2000]). The signal timing and phases serve as the control variables. The constraints include man/min green time, fixed cycle length, and the physical limitations of vehicles. The objective function is

minimized by DP, which also generates the optimal control variable in real-time.

Compared to other control methods, planning-based methods are harder to be implemented for real-time arterial or network control due to the high complexity of the optimization models. Another reason is that even with CVs, we still need to predict vehicles volumes, delays, speeds, or queue lengths in order to optimize signal timing. Such predictions are mutually dependent on signal timing, making the problem a complex, integrated optimization problem. This is particularly true for corridor level or network level control. Beak et al. [2017] proposed a two-level optimization method for corridor level signal control. At the intersection level, they used DP to allocate the optimal green time to each signal phase by considering the coordination constraints. At the corridor level, they formulated a mixed integer linear program based on the information of individual intersections to generate the optimal offsets, which were then sent to the intersection level as the coordination constraints. Simulation results showed that the proposed algorithm can reduce the average delay as well as the number of stops for a corridor compared to conventional actuated-coordinated signal control methods. Similarly, Li and Ban (2019) Li and Ban [2020]formulated the corridor signal control problem as a centralized MINLP considering the fixed cycle length constraint to reduce both fuel consumption and travel times. The MINLP was decentralized to a two-level model: At the intersection level, the phase duration were optimized by a DP algorithm initially proposed by Li and Ban [2018]. At the corridor level, the optimal offsets were updated by the MINLP using the optimal phases generated at the first level. They tested six cases considering different demands, and the simulation results showed that the performance of both major and minor streets improved under high traffic volumes.

Planning-based signal control could predict future traffic states and find the optimal solution within certain forward time horizon, which makes it more desirable. However, the computation cost might be high due to the resulting complex optimization problem, especially when dealing with large-scale networks or considering different movements of vehicles. In addition, the prediction horizon needs to be carefully selected. The longer the prediction horizon is, the more future information it could utilize, while the computation burden and

the prediction errors may also increase. On the contrast, shorter prediction horizon could make the computation faster but may decrease the control performance due to the lack of future traffic information.

Network-wide traffic signal control with CAVs

An urban traffic system usually consists of multiple intersections, making UTC a network-wide control problem. There are essentially two types of methods in the literature for the network-wide traffic signal control problem: centralized and decentralized (i.e., distributed) control. Centralized methods formulate the network-wide UTC as an integrated problem, which have the potential to obtain the global optimal solution. However, the model complexity increases as the size of the traffic network increases, making it harder and even impossible to solve the centralized control problem. Decentralized methods can avoid the increasing complexity problem by decomposing the network into small sub-networks or even single intersections, while the obtained solution may be sub-optimal.

Centralized methods are usually adopted to solve small network UTC, especially corridor level UTC problems. Apart from the corridor-level traffic signal control methods introduced in above sections, there are still some studies that use centralized methods to solve network-wide traffic signal control problem. For example, Li et al. [2016b] proposed a bi-level traffic signal plan optimization method by considering the network equilibrium condition. They optimized traffic signal settings at the intersection level (i.e., upper level) aiming to minimize average travel time, found the corresponding network equilibrium at lower level, and re-optimized the traffic signal settings until they converge. Case studies showed that the proposed method could reduce the average travel time of all travelers.

Decentralized methods have been receiving increasing attention due to the high computation efficiency and scalability, especially when it comes to network-wide UTC problems. There are basically two approaches to achieve decentralized network-wide traffic signal control: hierarchical control and distributed control. Hierarchical control methods usually decompose the network traffic control problem into a multi-level optimization problem such

that solving it is more manageable. Most hierarchical methods make macroscopic regional-level decisions (such as perimeter control) at the upper level and microscopic intersection-level decisions (such as signal phases and timings) at the lower level (Zhou et al. [2016b,a], Keyvan-Ekbatani et al. [2019], Yildirimoglu et al. [2018]). For example, Zhou et al. [2016a] proposed a hierarchical predictive UTC methods. They decomposed a heterogeneous large-scale urban traffic network into multiple homogeneous sub-networks which are modeled by the macroscopic fundamental diagram (MFD, Daganzo and Geroliminis [2008]). They formulated a higher-level problem to balance the travel demand and a lower-level problem to determine the optimal signal timings within each sub-network. Yildirimoglu et al. [2018] developed a hierarchical traffic management system to improve the network-wide traffic performance. At the upper-level, they used MPC scheme to optimize network performance by determining regional split ratios. At the lower-level, they modeled the path assignment problem as an integer linear programming (ILP) to generate the recommended routes. However, hierarchical control methods usually require extensive information from the network and the decomposition of different levels can seriously affect algorithm performance. In addition, it should be noted that, although the hierarchical UTC is naturally a spatial multi-scale problem, existing hierarchical control methods usually do not recognize different time scales.

Distributed signal control is a form of multi-agent systems in which a collection of intelligent agents (i.e., traffic signals) employ some form of cooperation to realize the shared goals of the organization (De Oliveira and Camponogara [2010]). In traffic signal control, the information of queue lengths and vehicle throughput of individual intersections were usually shared to provide this cooperation. Though distributed approaches are incapable of finding global signal timing optima, they can often find solutions that move traffic through the network more efficiently (Negenborn and Maestre [2014]) to achieve near-optimal results. Distributed signal control has been widely studied in the past several decades. For instance, Xie et al. [2011] proposed a platoon-based self-scheduling algorithm for real-time traffic network signal control. In this work, each intersection was controlled by a self-interested agent with the knowledge of platoon information of neighboring intersections. Based on the in-

formation of currently anticipated queues and incoming platoons from other intersections, the self-scheduling algorithm generated two possible actions (i.e., to extend or terminate the current phase) within each decision rolling horizon, aiming to keep vehicles moving rather than simply clear the queues. Timotheou et al. [2014] modeled the traffic signal network control problem as an MILP and described vehicle dynamics using the cell transmission model (CTM) in which road segments are discretized into units of space and time for easier analysis. Their simulations demonstrated close-to-optimal results that are scalable to larger networks. Mehrabipour and Hajbabaie [2017] also employed CTM in their study. They modeled the traffic control problem as an MILP at the intersection level. Their model maximized intersection vehicle throughput while promoting coordination among intersections through information sharing. A similar study by Al Islam and Hajbabaie [2017] attempted to maximize vehicle throughput through each intersection in a network while penalizing vehicle delays. They further extended the CTM-based traffic signal optimization method to the scenario of partially observable connected vehicle information Al Islam et al. [2020]. Other distributed control methods such as predictive-basedDe Oliveira and Camponogara [2010], backpressure-based Wongpiromsarn et al. [2012], and rule-based McKenney and White [2013], have also been proposed in the literature and tested/validated mainly in simulations.

The above-mentioned methods are all model-based, which usually suffer from low computation efficiency. For model-based methods, there is always a trade-off between model complexity (which is often positively related to the model's ability to capture real-world scenarios) and computation efficiency. Even for the distributed control, the computation burden increases as the model complexity and the size of network increase. Recently, as the rise of artificial intelligence (AI), data-driven methods for solving network-wide urban traffic signal control problems have been emerging, of which reinforcement learning (RL Sutton and Barto [2018]) and imitation learning (IL Pomerleau [1988]) are the two main techniques. For example, Zheng et al. [2021] proposed a delayed propagation transformer for the control problem in cyber-physical systems (of which the UTC problem is a specific example). They designed a cone-shaped spatial-temporal attention prior to capture the information propagation so

that the global view could be obtained. Integrating the attention prior with a centralized deep Q-network, the proposed method outperformed the state-of-the-art expert methods on synthetic and real-world urban traffic signal control scenarios. Wei et al. [2019b] developed a distributed RL method (which is called CoLight) for network-wide traffic signal control. They used a graph attentional network to capture the information communication between one specific intersection and the neighboring intersections, and the deep Q-network to learn the optimal traffic signal phases. Other RL based methods, such as Wei et al. [2019a], Chen et al. [2020], Xu et al. [2020], Chu et al. [2019], have also been proposed in literature. More methods and discussions can be found in the review Wei et al. [2019c]. Among these AI-based methods, RL-based methods have shown great potential in network-wide urban traffic signal control. However, they usually suffer from the exploration problem and even fail to converge. While studies of RL-based network-wide traffic signal control is numerous, IL-based methods are relatively sparse (Li et al. [2020], Xiong et al. [2019]). Li et al. [2020] proposed a deep IL method for traffic signal control and based on the graph convolutional neural networks. They firstly collected the traffic state data and the corresponding signal settings designed by experts, then developed a centralized graph CNN to map the network-wide traffic state to the signal plans of each intersection. Simulation results showed that the proposed method could reduce the average waiting time by 6.6%.

In summary, existing studies of network-wide UTC mainly focus on traffic signal control (rather than co-optimizing signal timings and vehicle trajectories simultaneously). Distributed control is the leading technique in model-based network-wide UTC problem due to the relatively low model complexity and high computation efficiency (compared with centralized, model-based methods). However, the global properties such as optimality and stability are hard to be analyzed and guaranteed. Learning-based network-wide urban traffic signal control can further improve the computation efficiency, but the mathematical properties can not be analyzed. RL-based methods are the dominant learning-based methods, but usually suffer from exploration and convergence problems, for which IL-based techniques may help.

2.1.2 Advanced traffic-oriented vehicle control with CAVs

Considering that this study focuses on traffic control, we do not include reviews on purely CAV control, which is a very broad topic in literature. Instead, we focus on the traffic-oriented driver assistance and vehicle control. We categorize the studies on this topic into two types: advanced driver guidance and CAV enhanced traffic control.

Advanced driver guidance based on CAVs

The information provided by CAVs could be used to better infer the traffic states. And the signal information can be shared through V2X technologies. In this way, we can develop advanced driver guidance system to provide optimal driving strategies based on those information to help drivers better operate the vehicles. The objectives of the guidance could be different, e.g., to avoid being caught by red lights (Li et al. [2011]) or save fuel by driving in economic modes (Katsaros et al. [2011], Schuricht et al. [2011], Tang et al. [2018], Ubiergo and Jin [2016], Wu et al. [2010]). These driving strategies may also be executed by automated vehicles, which can reduce the uncertainty of compliance of human drivers and help improve the control performance.

Reducing fuel consumption is one of the most important objectives of the driver guidance techniques, which is also called eco-driving guidance and is believed to have the potential of saving fuels from 5% to 15% (Van Mierlo et al. [2004]). Different eco-driving guidance algorithms may concentrate on different considerations: for examples, driver behaviors like lane changing and vehicle platooning, road structures like single/multiple intersections, traffic states like surrounding traffic and mixed traffic. Rakha and Kamalanathsharma [2011] proposed a rule-based eco-driving strategy at signalized intersections based on the V2I communication. They used a real-world data based statistical emission model that consists of linear, quadratic and cubic combinations of speed and acceleration levels to describe fuel consumption. Integrated with the vehicle dynamics model and the information of signal phases and timings, the rule-based eco-driving model was then designed to optimize fuel

consumption by providing speed profiles. The objective J is the total fuel consumption, which is calculated by the vehicle speed (state variable x) and acceleration (control variable u). The information of signal phases and timings serve as the environment inputs d. The vehicle speed follows the vehicle dynamics and also the constraints of the vehicle's physical limitations. Together, an approximate minimization of the objective function was achieved by the rule-based controller. While the work of Rakha and Kamalanathsharma [2011] focused on a single intersection, Boriboonsomsin et al. [2012] extended the eco-routing guidance to the entire trips of a vehicle, which is also known as eco-routing. They used a dynamic roadway network database to integrate and store historical and real-time traffic information. In addition, they used a hybrid method that combines the microscopic energy model with a large vehicle activity database to create the relationships between link-based energy factors and the link-based explanatory variables. Based on such relationship, they estimated the energy/emission operational parameter set by a multivariate regression method. The Dijkstra algorithm was then used to build the routing engine to search the shortest paths based on different objectives. The evaluation results showed that the eco-routes could generate about 13% fuel savings while the travel time might slightly increase.

Apart from fuel consumption, other objectives like those related to mobility may also be considered when designing driver guidance algorithms. For this, integrating multiple objectives in one algorithm is a promising direction. Katsaros et al. [2011] proposed a rule-based green light optimized speed advisory (GLOSA) algorithm to reduce fuel consumption and meanwhile improve traffic efficiency. The key idea of the method is to reduce the stop time at intersections. Their algorithm firstly calculated the distance and travel time to the front traffic signal, then calculated the target speed based on the rules that were predefined considering different signal phases at the estimated arrival time. The advisory speeds were then presented to drivers via V2X. Assuming that drivers would follow the advisory speeds, they built an integrated cooperative ITS simulation platform and tested their algorithm under different CAVs penetration rates. The results showed that the proposed method could improve fuel consumption by up to 7% and reduce stop time up to 80% at intersections.

Since drivers usually cannot perfectly follow the advisory speeds in reality, the effectiveness of the proposed methods needs to be proven in real traffic scenarios. To the best of our knowledge, some research groups and government agencies have conducted field experiments to evaluate the driver guidance systems. For example, the GLOSA system has been tested by field experiments in the city of Ingolstadt, Germany (Bodenheimer et al. [2014]). The fuel-saving oriented driver guidance platform "Glidepath Prototype system" has also been developed and tested through extensive field experiments, of which the results showed a 17% fuel improvement on average (Altan et al. [2017]). However, it should be pointed out that many existing studies about driver guidance systems were only tested in simulations. More efforts on real-world applications and testing of such systems need to be done in the future. In addition, the vehicle dynamics models and traffic models used in those methods were usually oversimplified, making the results less convincing. We expect more research studies can focus on this direction in the future to provide more empirical testing/validation results.

CAV enhanced traffic control

Studies on CAV enhanced traffic control can be categorized into two types: flow-based traffic assignment and vehicle-based traffic control. The flow-based traffic assignment usually treats the traffic network as a graph and study how to assign the mixed traffic flow from at least the link level. It usually aims to improve CAVs' or the whole system's performances by assigning routes for CAVs. While the flow-based traffic assignment is more macroscopic, the vehicle-based control, on the contrast, focuses on microscopic vehicle control. It usually tries to design CAV control algorithms considering the interactions with surrounding non-CAVs to achieve certain goals such as guaranteeing safety, reducing energy consumption, etc.

Wang et al. [2019] proposed a multi-class traffic assignment model for mixed traffic flow of HDVs and CAVs. They adopted the cross-nested logit model to formulate HDVs' route choice behaviors which could capture HDV user's uncertainty due to the limited knowledge of traffic conditions. They modeled CAVs' route choice principle by user equilibrium considering their capability for collecting accurate traffic conditions. They also developed

a new solution algorithm based on route-swapping and self-regulated step size choice to solve the problem. Sensitivity analysis and case studies showed that the proposed model could well capture the real mixed traffic flow traffic dynamics. Hu et al. [2018] developed a simulation-based dynamic traffic assignment method for different physical vehicle types (i.e., car, bus, motorcycle, and truck) considering four different behavior rules (i.e., pre-specified-path, user-equilibrium, system-optimal, and real-time information). Numerical experiments showed that the proposed simulation-based dynamic traffic assignment method was capable of capturing different travelers and route choice principles. Apart from above studies that focus more on "modeling", there are also studies aiming to "control" the mixed traffic flow in order to improve system's performance. For example, Guo et al. [2021] developed a dynamic bi-level framework to capture the behavior and interaction of HDVs and CAVs flows and formulated the problem of improving the system performance by dynamically controlling the routes of CAVs as an optimal control problem with equilibrium constraints (OCPEC). They also developed a decomposition based heuristic model predictive control (HMPC) method to efficiently solve the OCPEC. Experiment results showed that the proposed methods could significantly improve the network performance of both CAVs and HDVs.

For the vehicle-based traffic control, Wang et al. [2020b] analyzed the controllability and designed a system-level optimal control strategy using the Popov-Belevitch-Hautus criterion for a single-lane ring road. They proved that a ring-road mixed traffic flow system was not completely controllable, but was stabilizable under certain conditions. Numerical experiments showed that the proposed control method could use CAVs to smooth the mixed traffic flow. Gong and Du [2018] developed a cooperative platoon control method for the mixed flow to ensure the flow smoothness and stability as well as individual vehicles' mobility and safety. They designed a constrained One- or P-step MPC methods to control the CAV platoons, and a distributed algorithm to solve the MPC in order to be adaptive to the changes of the platoon topology. Numerical experiments indicated that the proposed algorithm could solve the MPC problem efficiently and the cooperative MPC could dampen traffic oscillation propagation and stabilize the traffic flow. There are also other CAVs control methods under

mixed traffic flow that focus on different scenarios and control techniques, e.g., lane-changing control (Du et al. [2020]), Monte Carlo tree search based control (Cheng et al. [2020]), leading cruise control (Wang et al. [2020a]), ecological adaptive cruise control (Lu et al. [2019]), speed harmonization (Ghiasi et al. [2019]), decentralized control (Yao and Li [2020]), etc.

2.2 Multi-scale UTC with CAVs

2.2.1 Multi-scale modeling, control theories and methods

A system can be considered as a multi-scale system if it is architecturally multi-layered due to modularity and hierarchy, i.e., spatially multi-scale, or has multiple time resolutions due to the state dependency, i.e., temporal multi-scale, or contains both spatial and temporal multi-scale features (Leskovec et al. [2010], Chapman and Mesbahi [2016]). It is a common case that the spatial and temporal multi-scale features are mixed for a multi-scale system. There are increasing applications of the multi-scale modeling/control in various domains such as the power systems (Ouyang et al. [2019], Zafar et al. [2019], Qiao et al. [2019], Xu et al. [2017b]), biology systems (Dada and Mendes [2011]), chemical systems(Lee and Othmer [2010]), and meteorology (Grabowski and Smolarkiewicz [2002], Castorina et al. [2018]). For example, Xu et al. [2017b] proposed a multi-scale coordinated voltage/var control method for high renewable-penetrated distribution systems. They modeled the voltage fluctuation counteraction problem as a two-scale (an hourly timescale and a 15-min timescale) stochastic programming problem, and converted it to a deterministic mixed-integer quadratic programming equivalence model. This study focuses on the multi-scale modeling and control of urban traffic, and we present current efforts and achievements towards this direction in Section 2.2.2.

To the best of our knowledge, there is no systematic theory that can properly model, analyze, and efficiently solve the control problem of a general multi-scale system, although some efforts and advances have been made. There are two theories behind current theoretical studies of the multi-scale system: the model-based theories such as singular perturbation

theory (Fenichel [1979], Smith [1985]) and data-driven theories such as dynamic mode decomposition (DMD, Schmid [2010], Schmid et al. [2011], Tu [2013]). For example, Chapman and Mesbahi [2016] proposed a singular perturbation theory based formulation to model the large-scale multi-time-scale system, and decomposed the system to multiple reduced systems based on a separation principle. They also derived the bounds on the time scale parameters in order to guarantee the asymptotic stability of the dynamics. There are extensive literature on how to design the control scheme given a well established singular perturbation system (Phillips and Kokotovic [1981], Kokotović et al. [1999], Ellis et al. [2013], Chen et al. [2011]). The key assumption of the singular perturbation theory is that the groups of strongly related dynamics are known and the weak interactions between different scales are considered as perturbations. In other words, the singular perturbation method assumes that the system contains clearly different dynamics, i.e., the system should be separable. As a result, the control performance can be hampered if the dynamics of the system is non-separable. For the multi-scale UTC problem discussed in this study, the dynamics of different scales are highly integrated (i.e., non-separable) in either temporal or spatial domains. For example, in the two-scale SVCC problem studied later in this study, vehicles should strictly follow the guides of traffic signals, while the signal's states and controls are the same (i.e., the signal phase is both the state and control variable for the signal level problem). There are even no explicit differential equation based dynamics for the signal level problem. This makes it very hard to separate the dynamics between the vehicle and signal scales. Therefore, the singular perturbation method cannot be directly applied for multi-scale UTC problems. Nevertheless, we adopt some insights of the singular perturbation method regarding the separation of different scales, i.e., under proper assumptions and conditions (see Chapman and Mesbahi [2016]), the multi-scale problem can be reduced to sub-problems where the dynamics of a specific scale are only relevant to the current and neighboring slower scales. Specifically, for the SVCC problem that will be discussed in this study, the slower-scale (i.e., signal-scale) dynamics are not related to the faster-scale (i.e., vehicle-scale) states, while the faster-scale dynamics should be constrained by the faster-scale states.

The singular perturbation theory is a model-based method. In recent years, with the rise of AI, data-driven methods have been developed to model multi-scale systems. Kevrekidis et al. [2003] proposed an equation-free multi-scale computation method based on the coarse time-stepper tool (Theodoropoulos et al. [2000]) when the dynamic equations conceptually exist but were not available in closed form. As an alternative, Kutz et al. [2016] extended the DMD methods (Schmid [2010]) to the multi-scale system. The key idea of DMD is to aggregate a range of continuous snapshots of a non-linear dynamic system and produces a least-square regression to approximate the system dynamics (Tu [2013]). The DMD modes with temporal frequencies near the origin can be used as the background (i.e., slower-scale) portions and the DMD modes with temporal frequencies bounded away from the origin are the faster-scale portions. Numerical results showed that the multi-scale DMD could naturally decompose the non-linear system into slower and faster sub-systems.

2.2.2 Multi-scale modeling of UTC and SVCC

In the field of UTC, the multi-scale modeling/control technique has long been recognized as a viable modeling framework. For example, a program on advanced technology for the highway (PATH, Kanafani and Parsons [1989], Broucke and Varaiya [1997]) was designed for controlling the platoons on interconnected highways. Five layers (i.e., network layer, link layer, coordination layer, regulation layer and physical layer) were designed, working with each other, to control the activities of vehicle platoons to improve the overall performance. Most of other hierarchical traffic control frameworks such as the dolphin framework (Tsugawa et al. [2000]) used the PATH as guidelines. Based on those architectures, Baskar et al. [2007] proposed an integrated hierarchical traffic control framework by combing the advantages and extending in various directions.

Apart from the application-oriented projects, there are also academical studies considering the hierarchical traffic control. For example, Head et al. [1992] proposed a hierarchical framework for real-time traffic control. They decomposed the signal control problem to three sub-problems based on different time and distance horizons: the route choice level, the traf-

fic flow and queue dynamics level, and the individual drivers level. However, limited by the hardware and software at that time, they only provided a conceptual framework without applications either in simulation or real-world. Later, they enriched this framework by adding specific algorithms for each level (Mirchandani and Head [2001], Mirchandani and Wang [2005]). They firstly estimated the traffic states (e.g., queue size and travel times), then predicted the link and platoon dynamics based on the estimated states and the real detector data. Finally, they optimized the network flow at the first stage to get the target phase timings, based on which they optimized the intersection flow performance at the second stage.

Recently, with the rise of faster computation machines and new technologies in vehicles and transportation systems such as AVs and V2X, the multi-scale traffic control, especially the SVCC, has been regaining the attention of researchers. A common feature of the advanced traffic-oriented vehicle control discussed in Section 2.1.2 is that they usually neglect the traffic signal control, making it a single-scale vehicle control problem. Studies on urban traffic signal control reviewed in Section 2.1.1, on the contrast, usually do not pay particular attention to vehicle control. Signal control and vehicle control are mutually dependent in reality: signal timing influences the movements of individual vehicles and thus the performances (such as emissions and fuel consumption) of the vehicles, while at the same time individual vehicle performances are the critical input to traffic control methods on how to best adjust signal timing. In the past, however, the information exchange between vehicles and signals were quite limited: signals detect the arrivals of vehicles (often as the aggregated number of vehicles in a certain time window) and adjust signal timing accordingly, while vehicles/drivers see the signal timing and adjust driving accordingly when they are close to the intersection. This makes it impossible to implement coupled signal and vehicle control. With CAVs, information between signals and (individual) vehicles can be exchanged in real time, which should be leveraged to further improve the traffic control performance, leading to the SVCC (Guler et al. [2014], Sun et al. [2017], Xu et al. [2017a], Yang et al. [2016], Yu et al. [2018]).

In general, the optimization model for SVCC is more complex, involving both linear and non-linear states, discrete and continuous decision variables, and other complex constraints, making the problem much more challenging to solve. To the best of our knowledge, this topic has just receiving attention in recent years and the literature is relatively sparse. Li et al. [2014] developed a signal control algorithm for automated vehicles at isolated signalized intersections. The method can simultaneously optimize vehicle trajectories and signal timing plans. By considering only two phases of the traffic signal, they used a simple enumeration method to select the optimum signal timing plan. They firstly determined the trajectory of the first vehicle, then calculated the trajectories of the following vehicles, and finally assigned the vehicles to different cycles by checking whether a vehicle can depart the intersection before the end of the green. A rolling horizon scheme was developed to implement the algorithm and to process newly arriving vehicles continually. Simulation results showed that the proposed algorithm can reduce the average travel time delay by 16.2-36.9% and increase the mobility by 2.7-20.2% compared to traditional actuated signal control methods. Xu et al. [2017a] proposed a cooperative control method to simultaneously optimize traffic signal timing and vehicle operations to improve transportation efficiency and vehicle fuel economy. The method considered multiple objectives: safety, mobility, and energy use. For this, the modeling/control framework of the study defined different priorities for different objectives: safety first, mobility second, and energy third. Accordingly, they proposed to solve safety by design, i.e., by applying the classical "dual-diagram" signal control design, improve mobility by optimizing signal phases and timing to minimize the total travel time of all vehicles, and save energy by optimizing the trajectory of each vehicle given its time budget to pass the intersection (obtained via solving the mobility objective). The proposed approach thus decomposed the complex control problem into three (much simpler) sub-problems based on the priorities of the three main objectives. Cooperating with each other, the three sub-problems were solved sequentially. Simulation results in MATLAB and VISSIM showed that the proposed method could improve traffic efficiency by 19.7% and fuel economy by 23.7% compared to a benchmark actuated signal control algorithm. Guo et al. [2019b]

proposed an algorithm using dynamic programming with shooting heuristic as a subroutine to simultaneously optimize the CAV trajectories and traffic signal's phases/timings. The method could also deal with the mixed traffic flow with both CAVs and HDVs. Numerical experiments showed that, compared with adaptive signal control, the proposed algorithm could reduce the average travel time by up to 35.72% and reduce fuel consumption by up to 31.5%.

The above studies focus on single intersection SVCC with 100% CAV penetration. For the mixed traffic flow scenario, Niroumand et al. [2020] proposed a joint optimization method for vehicle-group trajectories and signal timing. They categorized the mixed traffic flow into two types: CAVs and connected human-driven vehicles (CHVs) that are modeled by a linear car-following model. They firstly solved the joint optimization problem of all CAVs and the traffic signal, then solved the trajectory optimization problem for CHVs, and enforced the CHVs to follow their immediate front vehicle by introducing a "while" phase. The traffic signal was optimized on a slower time scale while the trajectories of vehicles were optimized on a faster time scale. Numerical experiments showed that the proposed algorithm could reduce the total delay by up to 96.2% compared with the actuated signal control. However, this study focused on single intersection and assumed that CHVs would follow the recommended trajectories. Tajalli and Hajbabaie [2021] developed a traffic signal timing and CAV trajectory co-optimization method in the mixed traffic flow. They modeled the joint optimization problem as an integrated MINLP and decomposed the intersection-level program into several lane-level sub-programs. The key to deal with mixed traffic flow in this study was to capture the interactions between CAVs and HDVs by car-following models. They also two time scales to model the signal control (i.e., slower-scale) and the vehicle trajectory control (i.e., faster-scale). The numerical experiments showed that the proposed methods reduced the average travel time and fuel consumption by up to 41% and 31%, respectively. However, due to the integrated formulation of both signal and vehicle control, the run time of the solution algorithm was relatively large. The above studies focused on single intersection scenario, for the scenario of multiple intersections and mixed traffic

flow, Yang et al. [2021] developed a cooperative driving framework for corridors considering the platooning behavior of CAVs and the cooperation among traffic signals. At the corridor scale, they formulated the corridor-level signal coordination problem as a linear programming problem which could generate optimal offsets for each intersection. At the intersection scale, they calculated the expected arrival time of all CAVs by solving a MILP problem. At the vehicle scale, they designed a state transition diagram to decide the driving modes and the corresponding driving strategies of each vehicle.

2.3 Summary and research gaps

As a summary, UTC is naturally a multi-scale problem in both temporal and spatial domains. Multi-scale UTC has been receiving more and more attention as the rise of CAVs and V2X technologies. However, studies on this topic are sparse. SVCC is the most typical multi-scale UTC problem and has been studied the most (but not well-studied). There are several limitations of current studies regarding the multi-scale SVCC: 1) The multi-scale features are not well-recognized, analyzed, or utilized; 2) Current methods are usually tailored for a single intersection or corridors with full CAV penetration. Studies of network-wide multi-scale SVCC with mixed traffic are rare; 3) For network-wide multi-scale SVCC, heuristic scale-separation methods are usually used to decompose the problem and improve the computation efficiency, which makes it challenging to analyze the control properties (such as stability) rigorously; 4) The computation efficiency, although reduced by decomposition, is still low due to the introduction of integer variables and non-linear terms in the models.

In this study, we aim to investigate the network-wide multi-scale SVCC problem with mixed traffic flow. To do this, we firstly develop a general modeling and control framework for the multi-scale UTC problem. Then, we use this framework to explicitly model and solve the SVCC problem under the environment of single intersection with full CAV penetration. Due to the relatively simpler scenario (i.e., single intersection and 100% CAV penetration), we are able to mathematically analyze the properties such as control stability. Based on the single intersection SVCC, we build a network-wide SVCC algorithm based on distributed control,

which can also work under mixed traffic flow condition. Distributed control is selected as the fundamental mechanism which brings relatively high computation efficiency (compared with centralized control) and enables the method to apply to not only corridors but also networks. In addition, utilizing the data generated by the proposed model-based methods, we develop an IL-enhanced method to further improve the computation efficiency. We also conduct various numerical experiments on each algorithm to demonstrate their performances.

Chapter 3

A GENERAL FRAMEWORK FOR MULTI-SCALE UTC

In this section, we aim to develop a general modeling framework for the multi-scale UTC problem. As illustrated in Figure 1.1, let k_m be the discrete time instant of scale (level) m with ΔT_m the duration of the discrete time step. Denote the total number of scales as M and assume $\Delta T_1 < \Delta T_2 < \cdots < \Delta T_M$, i.e., scale k_1 is the fastest scale and scale k_M is the slowest scale. Also denote $\mathbf{x}_m(k_m)$ the state of scale m (e.g., speed and position at the vehicle control scale), $\mathbf{u}_m(k_m)$ be the control variable of scale m (e.g., acceleration at the vehicle control scale). Note that $\mathbf{x}_m$ and $\mathbf{u}_m$ may be vector variables in case multiple agents exist at the scale. For scale m, we call scale $m+1$ its upper scale (if $m < M$) and scale $m-1$ its lower scale (if $m > 1$). A general formulation of the multi-scale UTC problem can be expressed in discrete-time as **A0**:

$$\min_{\mathbf{u}_1(k_1),\mathbf{u}_2(k_2),\ldots,\mathbf{u}_m(k_m)} \sum_{k_m}^{T} J(\mathbf{x}_1(k_1),\mathbf{x}_2(k_2),\ldots,\mathbf{x}_n(k_m),\mathbf{u}_1(k_1),\mathbf{u}_2(k_2),\ldots,\mathbf{u}_n(k_m),k_m) \quad (3.1)$$

subject to (for each scale m)

$$\mathbf{x}_m(k_m+\Delta t_m) - \mathbf{x}_m(k_m) = \Delta t_m \mathbf{f}_m(\mathbf{x}_m(k_m),\mathbf{u}_m(k_m),k_m) \quad (3.2a)$$

$$\mathbf{h}_m(\mathbf{x}_{m-1}(k_m),\mathbf{x}_m(k_m),\mathbf{x}_{m+1}(k_m),\mathbf{u}_m(k_m),k_m) \le 0 \quad (3.2b)$$

$$\mathbf{e}(\mathbf{x}_m(k_m=0),k_m=0) = 0 \quad (3.2c)$$

$$\mathbf{x}_m^{\text{low}} \le \mathbf{x}_m(k_m) \le \mathbf{x}_m^{\text{up}}, \quad \mathbf{u}_m^{\text{low}} \le \mathbf{u}_m(k_m) \le \mathbf{u}_m^{\text{up}}. \quad (3.2d)$$

Equation (3.1) shows the overall goal of the UTC problem, e.g., to improve the performance (such as minimizing congestion, fuel consumption, emissions or a combination of them) of all vehicles, intersections, corridors, and networks, by optimizing the control variables of each scale (e.g., acceleration of the vehicle scale, signal phases/timings of intersection

scale, signal coordination at the corridor level, and perimeter control between regions of an entire network). Model **A0** might also be a multi-objective optimal control problem, where the objective function can be separated to each scale's own objectives. There are constraints and dynamics for each scale and across different scales. First, there are "intra-scale" dynamics for each individual scale, e.g., the vehicle dynamics based on Newton equations for vehicle-scale control, which can be formulated as an ordinary differential Equation (3.2a). In addition, state and control variables at each scale may follow equality or inequality constraints, e.g., vehicles should follow the car-following rules to guarantee safety. Second and unique to this multi-scale modeling framework, there are "inter-scale" dynamics and interactions (and constraints) among scales. In particularly, the state and control of scale m are also affected by its upper and/or lower scales. For example, a vehicle should follow the traffic signals from its upper level. This can be formulated by Equation (3.2b). It is reasonable to assume that inter-scale dynamics/interactions are stronger between neighboring scales. To simplify the models and discussions, we only consider "inter-scale" dynamics between neighboring scales in this study (as expressed in (3.2b) and shown in Figure 3.1). Third, the initial conditions are fixed, which is shown in Equation (3.2c). Fourth, as shown in Equation (3.2d), the states of each scale should be constrained by the physical limits, e.g., the acceleration of a vehicle should be limited by the physical limits and the speed should be constrained by the speed limit.

A0 is clearly multi-scale: it is explicitly so in the temporal domain as indicated by the time index k_m, and implicitly so in the spatial domain as each scale only concerns the control on a certain geographic area (a vehicle, an intersection, a corridor, a sub-network, or a network). An alternative way to the multi-scale formulation **A0** is to formulate the entire problem using the fastest time scale (i.e., scale k_1). This way details of the problem can be maintained but the size of the problem (especially the temporal dimension) can be excessively large, preventing any efficient analysis and computation methods to deal with the resulting model.

To better understand the details of the proposed multi-scale control framework **A0**,

we consider a multi-scale UTC problem that contains (from lower to upper scales) vehicle control, intersection control, corridor control, sub-network control, and network control, as shown in Figure 1.1. Figure 3.1 describes the control problem of each scale, which also provides the possible considerations of state variables, control variables, intra-scale dynamics, and inter-scale dynamics. Simple bound constraints and initial conditions are ignored as these are relatively trivial to model.

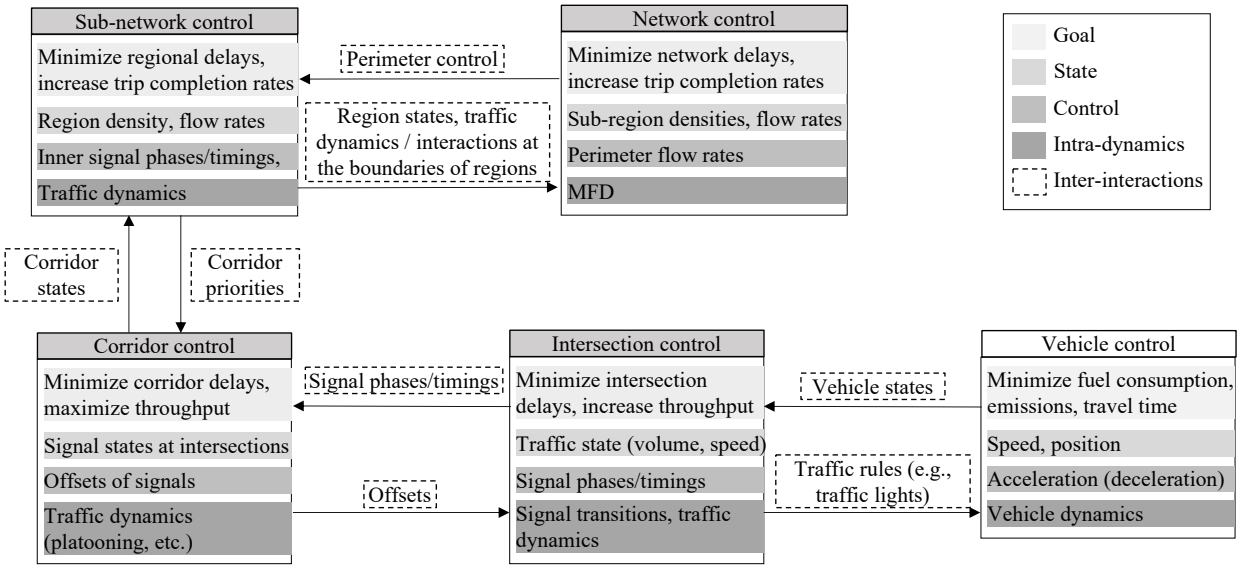

Figure 3.1: The multi-scale control problem for the UTC

In general, there are two approaches to apply **A0** to solve the multi-scale UTC problem: model-based approach and learning-based approach. There are also challenges for each approach, which we discuss in the following with some thoughts on possible research directions for solutions. In particular, we show how this dissertation address some of the challenges.

3.1 Model-based approaches

For model-based approaches, we need to build the explicit models of the studied problem based on **A0**. To this end, there are two key challenges that need to be carefully studied:

1) How to model (or approximate if necessary) the intra- and inter-scale traffic dynamics/interactions, and 2) How to analyze and solve the multi-scale problem.

For the first challenge, we have to note that many of the intra- and inter-scale traffic dynamics/interactions, as shown in Figure 3.1, have been well-studied in the literature (e.g., FD/MFD, car-following, platooning, etc.). In most cases, the resulting models can be explicitly described, which however can be complex and highly non-linear (e.g., some of the vehicle dynamics and car-following models (Li et al. [2017], Kesting et al. [2010])). Yet there are also intra- and inter-scale dynamics/interactions that we do not understand very well, e.g., the traffic interactions among the neighboring regions under MFD (Guo and Ban [2020], Ni and Cassidy [2019]), or the interactions among HDVs and CAVs in complex urban environments. Nevertheless, it can be expected that some of these dynamics/interactions can be complex or may not even be described by explicit functional forms. It is thus imperative to derive schemes to model, and more importantly, properly approximate the complex intra-/inter-scale dynamics/interactions for developing a tractable multi-scale framework for UTC. To this end, the newly emerging data-driven techniques may have great potentials, e.g., the DMD methods (Tu [2013], Proctor et al. [2016]) introduced in Section 2.2. Another possible way is to approximate the dynamics function by defining a function estimator (e.g., Taylor series approximation) and solving the optimization problem (constructed using data) to make the estimator functions as "close" to the original function as possible (see Chapter 1 in Dontchev and Rochafellar [2014]). In this way, the estimator function can provide useful guide when developing approximating schemes of the multi-scale model in **A0**. The third method is to integrate physical principles (models) into AI/ML models (the "physics-guided AI", Wang and Yu [2021]) to help reveal the underlying structure and understand the intrinsic dynamics/interactions in the multi-scale UTC problem. Some investigations on this has started in transportation (Yuan et al. [2020, 2021], Mo et al. [2021]), and in particular in the dynamic traffic assignment (DTA) field for approximating the complex dynamic network loading process of DTA (Wang et al. [2018], Bahuleyan and Vanajakshi [2017]).

This study focuses on the multi-scale SVCC problem. The intra-scale dynamics (e.g.,

the single vehicle dynamics and car-following behavior) and the inter-scale dynamics (e.g., traffic rules guided by the traffic signal such as right-of-way of each lane) are either have been well-studied or can be properly modeled using analytical formulations. Thus, we adopt the explicit modeling approach to capture the intra-and inter-scale traffic dynamics/interactions. Specifically, we model the traffic dynamics explicitly by formulating vehicle dynamics as Newton equations and describing the interactions among vehicle's using analytical car-following equations. We capture the interactions between vehicles and traffic signals by novel explicit formulations based on different time scale, and we model the integrated multi-scale SVCC problem as an explicit case of **A0** (see Section 4.2).

For the second challenge, as aforementioned, general theories, models, and methods to deal with a general multi-scale (both temporally and spatially) problem are still lacking currently, despite some recent advances (Ouyang et al. [2019], Zafar et al. [2019], Qiao et al. [2019], Xu et al. [2017b], Dada and Mendes [2011], Lee and Othmer [2010]). It is still very challenging to directly analyze and solve **A0**. For example, as discussed in Section 2.2.1, the singular perturbation methods cannot be applicable to multi-scale UTC due to the highly integrated dynamics across levels. Often, one has to resort to certain decomposition schemes to reformulate **A0** to multiple single-scale problems (one or multiple problems for each scale). Thus, there is a critical need now to develop more advanced multi-scale models, including analysis theories and computation methods, to analyze and solve **A0** directly. In addition, whether the multi-scale problem is solved directly or by decomposition schemes, some properties such as the control stability should be well considered.

In this study, as will shown in Section 4.2, solving and analyzing an integrated multi-scale problem directly is very hard, especially when there are no mature theories to guide the process. To address this issue, in this research, we propose a model predictive control (MPC) scheme by decomposing the integrated problem into well-designed sub-problems that can be solved more efficiently. Furthermore, we propose a stability analysis method for the MPC scheme, based on the consistency of critical states. There are two major steps for the stability analysis method: 1) Find critical states that need to be consistent (equal)

between different scales (usually two, e.g., one vehicle control problem and one signal control problem for the two-scale SVCC problem), and 2) Establish stability conditions by well designed objective functions, terminal conditions, and most importantly, the consistency of the critical states among different sub-problems. As we will show in Section 4.3, such a stability analysis method works well for the SVCC problem under full CAV penetration and single intersection. It can also be extended to the environment of multiple intersections with full CAV penetration (see Section 5). This methods may also work for the scenarios of mixed traffic flow and multiple intersections. Thus we hope that the proposed stability analysis method can provide a general framework, together with some useful insights, to motivate future research to study the stability issues of multi-scale UTC problems.

3.2 Learning-based approaches

If we models **A0** explicitly, there might be non-linear, non-smooth, non-convex, and integer terms included in the objective function, dynamic equations, and constraints, making it hard to be efficiently solved. By applying the proposed MPC scheme, we can simplify **A0** by selecting proper state representations, refining dynamic equations and constraints, and designing representative and effective objective functions, the computation and solution performance may be improved dramatically (see Section 4). However, when the size of the multi-scale UTC problem increases (e.g., from single intersection to multiple intersections), the computation burden also increases, making it expensive to apply the explicit model based methods to large traffic networks (see Section 5). There are several approaches to expedite the solution process and improve the efficiency. First, we can try to refine and simplify the model to make it easier to solve. For this, the simplicity and accuracy of the model need to be well balanced since they often conflict with each other. Second, we can adopt advanced computation techniques such as parallel and distributed computation (Bertsekas and Tsitsiklis [2015]). This approach does not really reduce the computation burden, but can solve the problem faster by the cooperation and coordination of multiple computers. Third, we can adopt some of the newly emerged learning-based algorithms, e.g., RL (Sutton and

Barto [2018]) and IL (Hussein et al. [2017], Schaal [1999]), to efficiently learn the optimal strategies for different scales. After being successfully trained under various traffic scenarios, the learned strategy could generate the control commands quickly (e.g., a neural network based strategy could calculate the commands by very fast forward computation).

Apart from the benefit of improving the efficiency, the learning-based method, especially the RL, is naturally an implicit formulation and direct solution process for the optimal control problem (Sutton et al. [1992], Bertsekas [2019]). The multi-scale UTC problem may be treated as a Markov decision process (MDP, Bellman [1957]), and theoretically, we can directly use the RL method to solve the multi-scale UTC problem by designing proper traffic state representations, reward functions, and learning algorithms. In fact, as shown in Section 2.1.1, there are already research efforts on applying RL methods to solve the traffic control problem, most of which focus on traffic signal control (e.g., Li et al. [2016a], Balaji et al. [2010], Abdulhai et al. [2003], Yau et al. [2017], Wei et al. [2019b], Zheng et al. [2021]) and AVs (e.g., Sallab et al. [2017], You et al. [2019], Xia et al. [2016]) separately. To the best of our knowledge, there are currently no studies discussing about applying the RL methods to solve the multi-scale UTC problem. The key challenge is that, compared with single-scale traffic signal control and vehicle control, the state and action (i.e., control variable) spaces of the multi-scale UTC problem are usually larger and more complex. For example, the actions consist of both signal's phases/timings and vehicle's acceleration/deceleration, while the former are usually discrete and the later are continuous. Such complex states and actions make it hard to develop efficient learning algorithms, and the large state/action spaces make it hard for the algorithm to explore and even fail to converge. The natural idea to address these issues is to design separate learning algorithms for each scale (e.g., vehicle agent, traffic signal agent, and perimeter control agent). To this end, apart from how to design efficient learning algorithms, how to connect the learning algorithm of each scale also needs to be carefully investigated. The general framework **A0** can provide insights for this issue. For example, when designing the state representations for a specific scale, we may only need to select the features from the current scale and its immediate neighboring scales.

Nevertheless, the RL methods may still suffer from exploration problem and even fail to converge due to the large state/action spaces of a specific scale (e.g., a large network consists of many intersections). IL methods can avoid such problems by utilizing demonstration data, i.e., the "expert" data (Li et al. [2020], Xiong et al. [2019]). IL methods aim to mimic expert's behavior by learning a mapping between observations and actions generated by the experts. However, IL methods may not perform well if there is not enough expert data.

In this study, we choose IL as the foundational learning method due to three reasons. First, RL usually suffer from exploration and convergence problems, making it hard to design the objective function and tune the parameters. Second, there is usually a large gap between the training performance of the RL methods and the real implementation performance, since RL is often trained in simulation environments due to the needs of exploration and extra work necessary for the transfer to real implementations. Third, one of the biggest weaknesses of IL is the lack of expert data because collecting such data is usually time-consuming and costly, especially for some manually calculated expert data. However, we have built the model-based methods, which can serve as the "expert" to efficiently generate learning data for IL. In this study, we will explore the idea of using model-based methods to generate training data for IL.

Chapter 4

MULTI-SCALE SVCC UNDER SINGLE INTERSECTION WITH FULL CAV PENETRATION

The goal of SVCC is to simultaneously optimize the vehicle trajectories and the traffic signal phases/timings to improve both vehicle performance (e.g., reduce fuel consumption) and the traffic performance (e.g., increase the throughput of the intersection). In real-world traffic systems, vehicles should be controlled with faster time scale (sub-seconds or milliseconds), while the control of signal timing phases can be much slower (e.g., a few seconds) to avoid the conflict and confusions caused by frequent phase changes. The multi-scale framework in **A0** then naturally applies.

In this section, we use the framework **A0** to model a two-scale SVCC problem under full CAV penetration and single intersection. Specifically, we first make several assumptions in Section 4.1 to clearly define the problem. We then present an explicit formulation of **A0** in Section 4.2. We show that such a problem is hard to be solved even after the simplification. Then, we present a model predictive control (MPC) scheme to decompose **A0** into a slower-scale signal control problem and a faster-scale vehicle control problem in Section 4.3. We show that such a multi-scale formulation can largely reduce the problem scale and make the SVCC solvable. We further prove the asymptotic stability of the proposed MPC scheme in Section 4.3.3 and conduct numerical experiments to demonstrate the performance of the MPC controller in Section 4.4.

4.1 Assumptions and notations

We firstly introduce the notations used in this study in Table 4.1 with most of the key parameters illustrated in Figure 4.1. Note that $i = 1, 2, ..., n_j$, $j = 1, 2, ..., J$, $l = 1, 2, ..., L$

where n_j is the number of vehicles on lane j, J is the total number of lanes around an intersection, and L is the number of candidate phases of the traffic signal.

Table 4.1: Notations

Symbol	Meaning	Type	Unit
$a_{i,j}(k_f)$	acceleration of vehicle i on lane j at time step k_f	continuous	m/s^2
$v_{i,j}(k_f)$	speed of vehicle i on lane j at time step k_f	continuous	m/s
$s_{i,j}(k_f)$	position of vehicle i on lane j at time step k_f	continuous	m
$f_{i,j}(k_f)$	fuel rate of vehicle i on lane j at time step k_f	continuous	mg/s
$g_{i,j}(k_s)$	whether the vehicle i on lane j has crossed the stop-line at time step k_s or not	binary	1/0
$\tilde{g}_{i,j}(k_s)$	whether the vehicle i on lane j has crossed the safety distance at time step k_s or not	binary	1/0
$p_l(k_s)$	whether the phase l at time step k_s is on or off	binary	1/0
$r_j(k_s)$	whether the lane j at time step k_s has the right-of-way or not	binary	1/0
τ_h	time headway	constant	s
d_0	safety distance	constant	m
$s_{j,\max}$	length of incoming lane j	constant	m
$s_{j,\text{safe}}$	safety tolerance distance of lane j	constant	m
$A^{J\times L}$	the mapping function from signal phase to right-of-way	constant	
ΔT_s	the step length of the slower-scale	constant	s
ΔT_f	the step length of the faster-scale	constant	s

To simplify the problem **A0** and clarify the scope of this study, we make the following assumptions. First, in this section, we consider the SVCC problem under single intersection scenario with the traffic signal controllable. Second, the internal dynamics of a vehicle, e.g., engine, powertrain, and tire dynamics, are neglected. The vehicle states are represented by

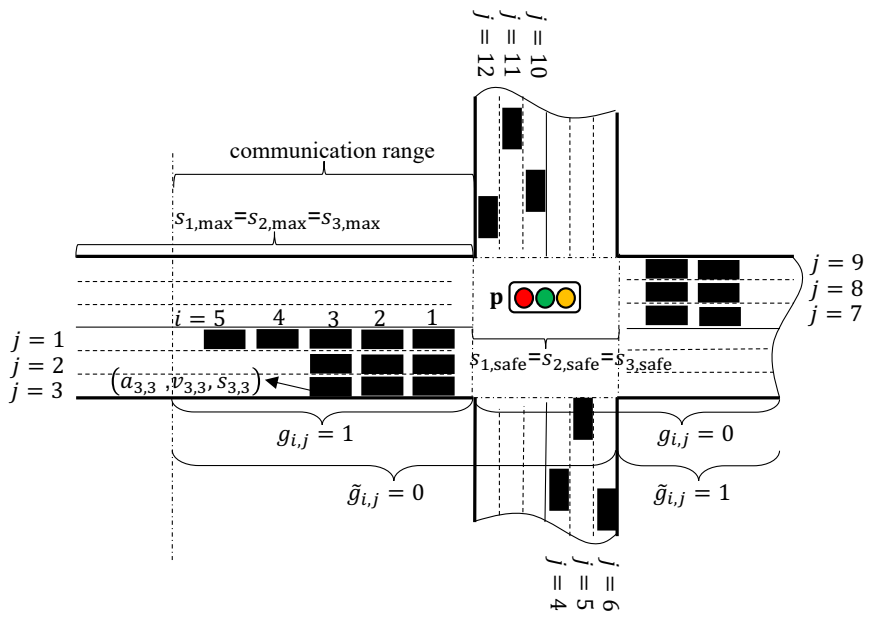

Figure 4.1: Illustration of the SVCC problem

the speed and position, and the control input is the acceleration. Third, we assume that all vehicles are CAVs, i.e., we can get the full information and control every vehicle. We will discuss how to release this assumption to mixed traffic flow in Section 5. Forth, only the longitudinal acceleration/deceleration of a vehicle is considered and the lateral lane-changing operations are neglected.

We redefine the notation in Section 3 a bit and let k_s and k_f be the time instants of slower (i.e., signal control) and faster (i.e., vehicle control) scale problems, respectively. In addition, the time intervals of k_s and k_f follow the following relationship:

$$\Delta T_s = c\Delta T_f, \quad c \in \mathbb{N}. \tag{4.1}$$

4.2 The integrated SVCC problem

4.2.1 Traffic signal modeling

Let $\mathbf{p}(k_s) = \begin{bmatrix} p_1(k_s) & p_2(k_s) & \cdots & p_L(k_s) \end{bmatrix}^T$ be the signal phase, $p_l(k_s)(l = 1, 2, ..., L)$ is a binary variable indicating whether the phase l is on ($p_l(k_s) = 1$) or off ($p_l(k_s) = 0$). For a typical four-leg intersection, normally there are eight phases; see Figure 4.4. Let $A^{J \times L}$ be the matrix that maps a signal phase to the right-of-way of corresponding lanes. Let $\mathbf{r}(k_s) = \begin{bmatrix} r_1(k_s) & r_2(k_s) & \cdots & r_J(k_s) \end{bmatrix}^T$ be the right-of-way vector where $r_j(k_s)$ is a binary variable indicating whether lane j at time k has the right-of-way ($r_j(k_s) = 1$) or not ($r_j(k_s) = 0$). We have the relationship:

$$A\mathbf{p}(k_s) = \mathbf{r}(k_s). \tag{4.2}$$

Considering that only one phase should be green at a time. We have the following constraint:

$$\sum_{l=1}^{L} p_l(k_s) = 1. \tag{4.3}$$

4.2.2 Vehicle modeling

The dynamics of a vehicle can be described by the following ordinary differential equations (in discrete format):

$$\begin{bmatrix} v_{i,j}(k_f + 1) \\ s_{i,j}(k_f + 1) \end{bmatrix} = \begin{bmatrix} v_{i,j}(k_f) + a_{i,j}(k_f)\Delta T_f \\ s_{i,j}(k_f) + v_{i,j}(k_f)\Delta T_f \end{bmatrix}. \tag{4.4}$$

Under the multi-vehicle scenario, there should be safety distances between consecutive vehicles, i.e., the vehicles should follow the car-following constraint:

$$s_{i,j}(k_f) - s_{i-1,j}(k_f) \geq \tau_h v_{i,j}(k_f) + d_0. \tag{4.5}$$

Note that we consider full CAV penetration and homogeneous vehicles in this section, τ_h and d_0 are set as the same for all CAVs for the sake of simplicity. When it comes to mixed traffic

flow or heterogeneous vehicles, the car-following parameters can be changed accordingly to reflect the real behavior.

The speed of a vehicle should be positive and follow the speed limit:

$$v_{i,j}(k_f) \in [0, v_{\max}]. \tag{4.6}$$

The acceleration of a vehicle should not exceed the physical limits:

$$a_{i,j}(k_f) \in [a_{\min}, a_{\max}]. \tag{4.7}$$

In addition, we introduce another constraint on $g_{i,j}(k_f)$ so that it can correctly reflect the position of each vehicle. If $g_{i,j}(k_f) = 0$, the vehicle has crossed the stop-line, if $g_{i,j}(k_f) = 1$, the vehicle is still on the incoming lane j. Given this, we have:

$$[1 - 2g_{i,j}(k_f)]\,[s_{i,j}(k_f) - s_{j,\max}] \geq 0. \tag{4.8}$$

In order to eliminate the non-linearity, we we rewrite Equation (4.8) as

$$s_{i,j}(k_f) - s_{j,\max} \leq M(1 - g_{i,j}(k_f)) \tag{4.9a}$$

$$s_{i,j}(k_f) - s_{j,\max} \geq -M g_{i,j}(k_f), \tag{4.9b}$$

where M is a sufficiently large constant value (i.e, "big M"). Equation (4.9) shows that when a vehicle has crossed the intersection, i.e., $s_{i,j}(k_s) - s_{j,\max} > 0$, $g_{i,j}(k_s) = 0$, otherwise, $g_{i,j}(k_s) = 1$. This is equivalent to (4.8). Note that here we only include the basic car following and vehicle safety constraints, without imposing additional vehicle related dynamics and intersections (e.g., platooning). This will allow the control framework to explore the optimal control strategies in a large solution space, which may help produce well-established strategies or phenomena such as vehicle platooning.

4.2.3 Interactions between traffic signal and vehicles

Vehicles should follow traffic rules. If the signal phase is red, the vehicle should not pass the intersection. We model such constraint by

$$[1 - r_j(k_s)]\,[s_{i,j}(k_f) - s_{j,\max}]\,[s_{i,j}(k_f) - s_{j,\max} - s_{j,\text{safe}}] \geq 0, \quad k_s = \lfloor \frac{k_f}{m} \rfloor. \tag{4.10}$$

The last two terms formulate a quadratic function with zeros at $s_{j,\max}$ and $s_{j,\max} + s_{j,\text{safe}}$. This function takes negative values in the interval $(s_{j,\max}, s_{j,\max} + s_{j,\text{safe}})$ and positive values elsewhere. If vehicle i at lane j has passed the intersection but hasn't pass the safety distance, i.e., $s_{i,j}(k_f) \in (s_{j,\max}, s_{j,\max} + s_{j,\text{safe}})$, $r_j(k_s)$ must be 1, i.e., lane j should have the right-of-way, to ensure constraint Equation (4.10). In addition, Equation (4.10) can naturally introduce yellow phase even we only explicitly formulate green and red phase: A lane's phase can not turn to red unless the last passed vehicle finished the safety distance.

In order to eliminate the non-linearity, we introduce another binary variable $\tilde{g}_{i,j}(k_f)$. Similar to $g_{i,j}(k_f)$, $\tilde{g}_{i,j}(k_f)$ is a position indicator: $\tilde{g}_{i,j}(k_f) = 1$ if the vehicle has passed the safety distance, i.e., $s_{i,j}(k_f) \geq s_{j,\max} + s_{j,\text{safe}}$, $\tilde{g}_{i,j}(k_f) = 0$ otherwise, i.e., $s_{i,j}(k_f) < s_{j,\max} + s_{j,\text{safe}}$. Same as Equation (4.9), $\tilde{g}_{i,j}(k_f)$ can be calculated by:

$$s_{i,j}(k_f) - s_{j,\max} - s_{j,\text{safe}} \leq M\tilde{g}_{i,j}(k_f) \tag{4.11a}$$

$$s_{i,j}(k_f) - s_{j,\max} - s_{j,\text{safe}} \geq -M(1 - \tilde{g}_{i,j}(k_f)). \tag{4.11b}$$

Then, Equation (4.10) can be reformulated as

$$g_{i,j}(k_f) + \tilde{g}_{i,j}(k_f) + r_j(k_s) \geq 1 \quad k_s = \lfloor \frac{k_f}{m} \rfloor. \tag{4.12}$$

Equation (4.12) shows that if $g_{i,j}(k_f) = 0$ and $\tilde{g}_{i,j}(k_f) = 0$ at time k_f, which indicates that vehicle i on lane j is located at the safety area, in order to make the left side larger or equal to 1, $r_j = 1$. If the vehicle locates at other positions, r_j could be 0 or 1. $k_s = \lfloor \frac{k_f}{m} \rfloor$ shows that, at a specific faster-scale time k_f, the corresponding slower-scale time is $\lfloor \frac{k_f}{m} \rfloor$. Therefore, as shown in Equation (4.12), a specific vehicle at the faster-scale should follow the traffic rule at the slower time $\lfloor \frac{k_f}{m} \rfloor$.

Noteworthy here is that due to the simplified assumptions, the intra- and inter-scale dynamics (4.8) and (4.10), although non-linear, are expressed in relatively simple forms. As shown above, they can be reformulated into their equivalent linear formulations by introducing "big M" and possibly extra binary variables. We can imagine that when more complex

intra-/inter-dynamics are applied, such equivalent reformulations may not exist or be done easily, in which case the data-driven methods as we discuss in Section 3.1 can be applied to develop schemes to approximate the complex dynamics.

4.2.4 Objectives

The objectives consists of two parts: 1) the total vehicle-time to pass the stop-line and 2) the total fuel consumption. The former is for traffic performance, and the latter is for vehicle performance. For the first part, we have

$$J_{\text{tt}} = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{k_f=0}^{T^f} w_1 g_{i,j}(k_f) \Delta T_f. \tag{4.13}$$

Note that $g_{i,j}(k_f) = 1$ if vehicle i on lane j has not crossed the intersection, and $g_{i,j}(k_f) = 0$ otherwise. J_{tt} in Equation (4.13) represents the accumulated vehicle-time of all vehicles before they cross the intersection. After having crossed the intersection, the vehicles are assumed to accelerate to the speed limit and then cruise.

For the fuel consumption, we have

$$J_{\text{fc}} = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{k_f=0}^{T^f} w_2 f_{i,j}(k_f) \Delta T_f. \tag{4.14}$$

Therefore, the objective can be expressed by

$$\begin{aligned} J &= w_1 J_{\text{tt}} + w_2 J_{\text{fc}} \\ &= \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{k_f=0}^{T^f} \left(w_1 g_{i,j}(k_f) + w_2 f_{i,j}(k_f) \right) \Delta T_f, \end{aligned} \tag{4.15}$$

where w_1 and w_2 are weights to scale different objectives.

4.2.5 A mixed integer non-linear programming (MINLP) formulation

The problem can be formulated as **A1**

$$\min_{p_l(k_s), a_{i,j}(k_f)} J$$

subject to

traffic signal related constraints : Equation (4.2), (4.3)

vehicle dynamics and related constraints : Equation (4.4) − (4.7)

vehicle position indicators : Equation (4.9), (4.11)

signal-vehicle coupled constraints/interactions : Equation (4.12).

A1 is a multi-scale (two-scale) formulation with k_s the slower-scale for signal control and k_f the faster-scale for vehicle control. Compared with the problem of modeling the whole problem at the same time scale k_f, the scale of **A1** is reduced, i.e., the signal-related decision and state variables $p_l(k_s), r_j(k_s)$ are much smaller. However, such a multi-scale formulation brings new challenges due to the involvement of the two time scales (k_s and k_f), which may not be solved directly. We next propose a model predictive control (MPC) scheme and decompose **A1** to two sub-problems, i.e., one slower problem mainly for signal control and one faster problem for vehicle control. In Section 4.3.3, we further prove the stability of such an MPC scheme.

4.3 An MPC-based decomposition scheme

The MPC scheme is shown in Figure 4.2. The MPC scheme is implemented in a rolling horizon manner. At a specific slower time step k_s, we formulate and solve the slower-scale problem for signal control (see Section 4.3.1) for forward T^s steps ($T^s = 6$ is used in the numerical section in this study). After solving the problem for T^s intervals, only the signal phase plan of the first slower interval $[k_s, k_s+1]$ is implemented and used. A new slower-scale problem will be formulated and solved at the next slower step $k_s + 1$ as well, and again, the first signal phase plan is used for the next interval. This process repeats itself until the end of all (slower) time intervals. Notice that we approximate the dynamics of vehicles at the slower-scale problem to reduce the computation effort, thus errors would be introduced due to the simplification. The MPC (and rolling-horizon implementation) can help correct and mitigate such errors.

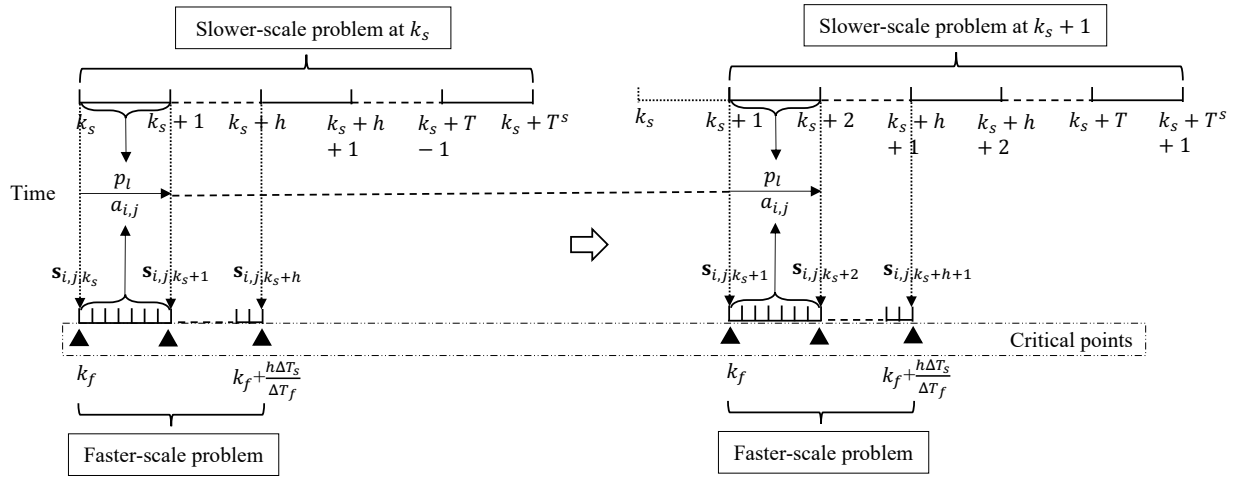

Figure 4.2: The decomposition and approximation framework

For the faster-scale problem, as shown in Figure 4.2, we utilize the first h (h is normally small, e.g., 2 or 3) predicted vehicle trajectory points generated by the slower-scale problem to construct the faster-scale problem (see Section 4.3.2). That is, we formulate the faster-scale vehicle control problem for the h slower steps, i.e., for the duration of $\Delta T_s * h$ (or $\Delta T_s/\Delta T_f * h$ faster-scale time steps). In particular, we ensure the faster-scale problem to produce positions of each vehicle at the corresponding slower time intervals be exactly the same as those generated by the slower-scale problem; see (4.22c) in **A3**. This is the so-called "consistency" of states between the two scales. These slower-scale time points (h in total) are called "critical time points" for which state consistency is required between the slower-scale and faster-scale. After solving the faster-scale problem, same to the slower-scale problem, we only implement the solutions of the faster problem for the next slower-scale time duration $[k_s, k_s + 1]$ (i,e., $\Delta T_s/\Delta T_f$ intervals of k_f). Under this MPC scheme, we essential decompose **A1** to two sub-problems that can be solved much more quickly. Furthermore, the stability of the MPC scheme can also be established based on the design of consistency of states as shown in Section 4.3.3.

Such an MPC scheme is not new for solving traffic control or modeling problems, e.g., it

has been widely used for solving DTA problems (Peeta and Mahmassani [1995], Peeta and Ziliaskopoulos [2001], Zhou and Mahmassani [2007], Mahmassani [2001], Ye et al. [2019], Guo et al. [2021]). What new here is that we propose to ensure the "consistency" of key traffic states between the two scales. For the signal control (slower) and vehicle (faster) scales studied in this study, the selected state is vehicle positions at the critical time points. That is, the slower-scale problem generates the position points for all vehicles at each critical time point, which need to be satisfied when solving the faster-scale problem. As we show later for the proof of Theorem 1 in Section 4.3.3, ensuring the consistency of the key traffic states (vehicle positions here) is essential for proving the stability of the MPC scheme.

To construct the MPC scheme and the two sub-problems, the key challenge is how to properly "separate" (decompose) the objective and constraints of the single multi-scale formulation **A1** into the two sub-problems with two time scales. For the objective function, similar to Xu et al. [2018], the slower problem focuses on designing signal plans to improve the traffic efficiency (i.e., J_{tt} in (4.15)), while fuel consumption is modeled at the vehicle-scale (i.e., J_{fc} in (4.15)). For constraints, naturally, signal related constraints in **A1** should be included in the slower-scale control problem and vehicle dynamics and related constraints should be included in the faster-scale vehicle control, while special attention should be paid to the coupled signal-vehicle control constraints (4.12). In addition, approximation is necessary during the decomposition due to the change of time scale or for simplification purposes. For example, since ΔT_s is relatively large (e.g., 5 seconds), a vehicle can accelerate or decelerate to almost any desired speed during the interval, no mater what the initial speed is. Therefore, for the slower-scale signal control problem, the first part in Equation (4.4) can be assumed to always satisfy and thus ignored. Next we present in detail how the two sub-problems are constructed.

4.3.1 Signal control: the slower-scale problem

As discussed above, the objective of the slower-scale problem **A1** is to improve efficiency, which can be expressed as

$$J = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} g_{i,j}(\hat{k}_s)\Delta T_s. \tag{4.16}$$

Also intuitively, instead of the acceleration $a_{i,j}(k_s)$, vehicle speed $v_{i,j}(k_s)$ can be used as the new decision variable for the signal-control problem. To reduce the complexity while describe reasonable dynamics, we further assume that the acceleration/deceleration is constant during the whole slower-scale interval. For example, if a vehicle tries to accelerate from $0m/s$ to $10m/s$, it is assumed that the vehicle adopts $2m/s^2$ as the acceleration. In this way, Equation (4.4) can be reduced to

$$s_{i,j}(\hat{k}_s) - s_{i,j}(\hat{k}_s - 1) = \frac{v_{i,j}(\hat{k}_s) + v_{i,j}(\hat{k}_s - 1)}{2}\Delta T_s. \tag{4.17}$$

The traffic rule constraint in **A1**, i.e., Equation (4.12), does not apply exactly. In **A1**, the time scale is small such that, for a vehicle passed the intersection, there must be at least one trajectory point locating at $[s_{j,\max}, s_{j,\max} + s_{j,\text{safe}}]$. However, this is not the case when the time scale increases. Therefore, we modify this constraint by

$$g_{i,j}(\hat{k}_s - 1) - g_{i,j}(\hat{k}_s) - r_j(\hat{k}_s) \leq 0. \tag{4.18}$$

Equation (4.18) shows that, if $g_{i,j}(k_s-1)-g_{i,j}(k_s) > 0$, i.e., the vehicle crossed the intersection at time step k_s, $r_j(k_s) = 1$. Further, (4.9) and signal/vehicle related constraints, i.e., (4.2), (4.3), (4.6), can directly apply. The slower-scale problem can then be formulated as **A2**

$$\min_{p_l(\hat{k}_s), v_{i,j}(\hat{k}_s)} \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} g_{i,j}(\hat{k}_s)\Delta T_s \tag{4.19}$$

subject to, for $\hat{k}_s \in [k_s, k_s + T^s - 1]$,

$$s_{i,j}(\hat{k}_s) - s_{i,j}(\hat{k}_s - 1) = \frac{v_{i,j}(\hat{k}_s) + v_{i,j}(\hat{k}_s - 1)}{2}\Delta T_s \tag{4.20a}$$

$$s_{i,j}(\hat{k}_s) - s_{i-1,j}(\hat{k}_s) \geq \tau_h v_{i,j}(\hat{k}_s) + d_0 \tag{4.20b}$$

$$-Mg_{i,j}(\hat{k}_s) \leq s_{i,j}(\hat{k}_s) - s_{j,\max} \leq M(1 - g_{i,j}(\hat{k}_s)) \tag{4.20c}$$

$$g_{i,j}(\hat{k}_s - 1) - g_{i,j}(\hat{k}_s) - r_j(\hat{k}_s) \leq 0 \tag{4.20d}$$

$$A\begin{bmatrix} p_1(\hat{k}_s) & p_2(\hat{k}_s) & \cdots & p_L(\hat{k}_s) \end{bmatrix}^T = \begin{bmatrix} r_1(\hat{k}_s) & r_2(\hat{k}_s) & \cdots & r_J(\hat{k}_s) \end{bmatrix}^T \tag{4.20e}$$

$$\sum_{l=1}^{L} p_l(\hat{k}_s) = 1 \tag{4.20f}$$

$$g_{i,j}(\hat{k}_s), p_l(\hat{k}_s), r_j(\hat{k}_s) \in \{0,1\}, v_{i,j}(\hat{k}_s) \in [v_{\min}, v_{\max}] \tag{4.20g}$$

$$s_{i,j}(k_s) = s^j_{i,k_s}, v_{i,j}(k_s) = v^j_{i,k_s}. \tag{4.20h}$$

In **A2**, signal phase p and vehicle position s are the state variables, while p and vehicle speed v are the control variables. s^j_{i,k_s} and v^j_{i,k_s} are the initial position and speed of vehicle i at lane j at time point k_s (i.e., obtained by solving **A2** at time k_{s-1}). Compared with problem **A1**, the scale and complexity of problem **A2** are largely reduced, making it solvable. The output of **A2** is the signal phase plan and the position of every vehicle at the slower-scale (i.e., larger) time interval. The signal phase plan can be used to control the signal, while the position of each vehicle is used as the guidance to design more detailed trajectory for individual vehicles.

4.3.2 Vehicle control: the faster-scale problem

The faster-scale problem is constructed for time $k_f = k_s * \frac{\Delta T_s}{\Delta T_f}$. Let $\mathbf{s}_{i,j,k_p}$ be the k_p ($k_p = 1, 2, \ldots, h$) reference position point (in the slower-time scale) for vehicle i at a specific lane j, which is given by solving the slower-scale problem as discussed above. And let $\mathbf{t}_{i,j,k_p}$ be the corresponding time points. The faster problem is to determine the trajectory for each vehicle to follow the position guidance from the slower-scale problem and satisfy the traffic rules. Since we only consider the longitudinal vehicle control, a specific vehicle can interact with the front and/or following vehicles on the same lane. Therefore, we formulate the faster-scale problem for each lane separately. For lane j, the faster-scale problem is formulated as an

non-linear program (denoted as **A3**):

$$\min_{a_{i,j}(\hat{k}_f)} \sum_{i=1}^{n_j} \sum_{\hat{k}_f=k_f}^{h\frac{\Delta T_s}{\Delta T_f}+k_f-1} f_{i,j}(\hat{k}_f)\Delta T_f \tag{4.21}$$

subject to, for $\hat{k}_f \in [k_f, k_f + h\frac{\Delta T_s}{\Delta T_f}]$,

$$v_{i,j}(\hat{k}_f) - v_{i,j}(\hat{k}_f - 1) = a_{i,j}(\hat{k}_f - 1)\Delta T_f \tag{4.22a}$$

$$s_{i,j}(\hat{k}_f) - s_{i,j}(\hat{k}_f - 1) = v_{i,j}(\hat{k}_f - 1)\Delta T_f \tag{4.22b}$$

$$s_{i,j}(\hat{k}_f) - s_{i-1,j}(\hat{k}_f) \geq \tau_h v_{i,j}(\hat{k}_f) + d_0 \tag{4.22c}$$

$$s_{i,j}(\mathbf{t}_{i,j,k_p}) - \mathbf{s}_{i,j,k_p} = 0, k_p = 1, 2, \ldots, h \tag{4.22d}$$

$$a_{i,j}(\hat{k}_f) \in [a_{\min}, a_{\max}], \quad v_{i,j}(\hat{k}_f) \in [v_{\min}, v_{\max}] \tag{4.22e}$$

$$s_{i,j}(k_f) = s_{i,k_f}, \quad v_{i,j}(k_f) = v_{i,k_f}. \tag{4.22f}$$

In **A3**, constraints (4.22a) - (4.22c) show that vehicles should follow the accurate vehicle dynamics and car-following safety constraints as in **A1**. Constraint Equation (4.22d) indicates that the vehicle positions generated by **A3** should be the same as the guidance trajectory points generated by the slower-scale problem **A2**. This is the exact mathematical expression of keeping consistency of the states between the two scales. The objective function of **A3** is to minimize the fuel consumption of all vehicles. The traffic efficiency (or mobility) objective is already guaranteed by (4.22d), i.e., the vehicle needs to be at certain position at certain time (predicted by the slower-scale problem).

Note that **A3** is a non-linear programming (NLP) problem with linear constraints and should obtain an optimal solution as long as its feasible set is not empty. The feasibility of **A3** is mostly guaranteed. The vehicle positions generated by **A2** are feasible since the optimal control variables $v_{i,j}(k_s)$ are feasible and the positions are calculated by Equation (4.20a). For **A3**, we can construct a feasible solution using constant acceleration $a_{i,j}$ between two consecutive slower points, i.e., $a_{i,j}(k_f) = (v_{i,j}(k_s+1) - v_{i,j}(k_s))/\Delta T_s$ for all $k_f \in [k_s, k_s + 1]$. However, there might be extreme situations such as vehicle accidents which make $\mathbf{s}_{i,j,h}$

not reachable (i.e., (4.22d) does not hold exactly). In the stability proof, we do need the assumption that **A3** is feasible.

We conclude this section by showing the dimensions of problems **A1**, **A2**, and **A3**. Here dimension means the number of binary variables, as this is the most critical to determine the computational burden to solve the three problems. The original problem **A1** is a mixed 0-1 integer non-linear programming problem. For **A1** at a specific slower time step, let n_{veh} be the number of vehicles within the communication range of the studied intersection, n^s_{var} be the number of binary variables in **A1**, and recall that T^s is the number of prediction steps, ΔT_s is the slower-scale step duration, and ΔT_f is the faster-scale step duration, the dimension of **A1** is roughly $n_{\text{veh}} n^s_{\text{var}} T^s \frac{\Delta T_s}{\Delta T_f}$. It requires exponential time to solve mixed integer programming problems in the worst case. Solving **A1** could be quite time consuming. After applying the MPC scheme, **A2** becomes a mixed 0-1 integer linear programming problem with a dimension of $n_{\text{veh}} n^f_{\text{var}} T^s$ where n^f_{var} is the number of binary variables in **A2**, and **A3** is a non-linear programming (with no binary variables) which can be solved in polynomial time. Compared with **A1**, the overall time complexity of **A2** and **A3** decreases dramatically. In this study, as will be discussed in the numerical section, we set $T^s = 6$, $\Delta T_s = 5s$, and $\Delta T_f = 0.2s$, and $n^s_{\text{var}} = 4$ and $n^f_{\text{var}} = 3$, unless noted otherwise. The dimensions of **A1** and **A2** are $600 n_{\text{veh}}$ and $18 n_{\text{veh}}$, respectively. Considering the worst case exponential time complexity of the mixed integer programming, the running time of **A2** can be largely reduced due to the dramatically reduced number of binary variables when compared with **A1**.

4.3.3 Stability analysis of the MPC scheme

We analyze and prove the stability of the MPC scheme in this section. We first introduce additional notation used in the analysis. Let a real-valued scalar function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$belongs to class $\mathcal{K}(\varphi \in \mathcal{K})$ if it is continuous, strictly increasing and $\varphi(0) = 0$. Let $x(k_s)$ be the state of **A2**. Note that for signal control, the state variable is also the control variable, thus does not have stability issues. In this section, $x(k_s)$ consists only of the vehicle states, i.e., $g_{i,j}(k_s)$ (which is also the mapping of $s_{i,j}(k_s)$ by Equation (4.9)).

Let $\mathbb{X}$ denote the state space. Let $\mathbb{X}_f(T^s) \in \mathbb{X}$ be the set of feasible states for problem **A2**. Let $\mathbb{X}_{T^s} \in \mathbb{X}$ denote the desired target set that contains the origin. Let $\mathcal{N}$ be a neighborhood of the origin $\mathcal{N} \in \mathbb{X}_f$. Let $\tilde{x}(k_s, t_s)$ denote the state vector of the newly arrived vehicles during the time step $[k_s, k_s + t_s]$, and let $\tilde{n}(k_s, t_s)$ denote the total number of $\tilde{x}(k_s, t_s)$. Assume $p^*(k_s) = [p^*(k_s|k_s), p^*(k_s + 1|k_s) \dots p^*(k_s + T^s - 1|k_s)]$ and $v^*(k_s) = [v^*(k_s|k_s), v^*(k_s + 1|k_s) \dots v^*(k_s + T^s - 1|k_s)]$ be the optimal control sequences for the signal and all vehicles at time step k_s, let $s^*(k_s) = [s^*(k_s|k_s), s^*(k_s + 1|k_s) \dots s^*(k_s + T^s - 1|k_s)]$ be the positions generated by the optimal control variables of **A2**, and let $g^*(k_s) = [g^*(k_s|k_s), g^*(k_s + 1|k_s) \dots g^*(k_s + T^s - 1|k_s)]$ be the corresponding vehicle position indicators. Note that all the above optimal control and state variables are for **A2** at time k_s. Denote $p^{\text{MPC}}(k_s) = p^*(k_s|k_s)$ and $v^{\text{MPC}}(k_s) = v^*(k_s|k_s)$. Let $J(x(k_s), p(k_s), v(k_s))$ be the objective function of **A2** and also define $l(x(\hat{k}_s|k_s), p(\hat{k}_s|k_s), v(\hat{k}_s|k_s))$:

$$J(x(k_s), p(k_s), v(k_s)) = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} g_{i,j}(\hat{k}_s) \Delta T_s, \tag{4.23}$$

$$l(x(\hat{k}_s|k_s), p(\hat{k}_s|k_s), v(\hat{k}_s|k_s)) = \sum_{j=1}^{J} \sum_{i=1}^{n_j} g_{i,j}(\hat{k}_s|k_s) \Delta T_s. \tag{4.24}$$

Here $\hat{k}_s$ indicates the step within the prediction horizon. Note that $l(x(\hat{k}_s|k_s), p(\hat{k}_s|k_s), v(\hat{k}_s|k_s))$ is bounded and monotonically nonincreasing with $\hat{k}_s$. We have

$$J(x(k_s), p(k_s), v(k_s)) = \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} l(x(\hat{k}_s|k_s), p(\hat{k}_s|k_s), v(\hat{k}_s|k_s)). \tag{4.25}$$

Denote the MPC value function at time step k_s as the minimum objective value

$$V^{\text{MPC}}(x(k_s)) = \inf_{p(k_s), v(k_s)} J(x(k_s), p(k_s), v(k_s)) = J(x(k_s), p^*(k_s), v^*(k_s)). \tag{4.26}$$

Denote the augmented MPC value function at time step k_s as

$$\tilde{V}^{\text{MPC}}(x(k_s), \tilde{x}(k_s, N)) = V^{\text{MPC}}(x(k_s)) + \tilde{n}(k_s, N) T^s \Delta T_s. \tag{4.27}$$

which is the sum of MPC value for vehicles at time step k_s and the objective value of the vehicles that will arrive in the next N steps, i.e., during $[k_s, k_s + N]$, assuming that all newly

arrived vehicles do not cross the intersection during the whole prediction horizon T^s. N is assumed to be a large integer compared with T^s. Note the augmented MPC value function (4.27) takes into account of the vehicles that will arrive in the future, and considers the "worst" case scenario, i.e., one of the newly arrived vehicles will pass the intersection by the end of the prediction horizon. It turns out that such an MPC value function is a proper Lyapunov function to establish the asymptotic stability of the MPC scheme for the two-scale SVCC problem; see the proof of Theorem 1.

We firstly show a Lemma in order to further prove the stability of the problem **A2**.

Lemma 1. *For **A2** constructed at time k_s, the prediction horizon T^s can be specified such that all vehicles at time step k_s can pass the intersection at the end of the prediction horizon, i.e., $g_{i,j}(T^s + k_s - 1) = 0$.*

Proof. Consider an optimal sequence of the controls:

$$\begin{aligned} p^*(k_s) &= [p^*(k_s|k_s), p^*(k_s+1|k_s) \dots p^*(k_s+T^s-1|k_s)] \\ v^*(k_s) &= [v^*(k_s|k_s), v^*(k_s+1|k_s), \dots v^*(k_s+T^s-1|k_s)]. \end{aligned} \tag{4.28}$$

obtained by solving **A2**. This optimal control sequence should clear all the vehicles "as soon as possible". This can be proved by contradiction. Assume there exits a time step $\hat{k}_s$ such that a vehicle (i,j) can pass the intersection, i.e., $g_{i,j}(\hat{k}_s) = 0$, with all the constraints satisfied. This actually constitutes a feasible control sequence, denoted as $v(\hat{k}_s|k_s)$. Assume that the optimal control variable $v^*(\hat{k}_s|k_s)$ prohibits the vehicle to cross the intersection, i.e., $g^*_{i,j}(\hat{k}_s) = 1$ (and all other variables stay the same). We can easily see that $v^*(\hat{k}_s|k_s)$ leads to a larger objective function (i.e., due to Equation (4.19)). That indicates the control sequence $v^*(k_s)$ is not optimal, which conflicts with the assumption that $v^*(\hat{k}_s|k_s)$ is optimal. Since the number of vehicles around the intersection at time k_s is finite, all vehicles can be cleared in a finite time. In fact, the roads around an intersection have physical length and we consider the vehicles on a range close to the intersection. Such a time should have a upper bound, which is denoted as $\hat{T}$. Let $T^s = 2\hat{T}$, we have $g_{i,j}(T^s - 1) = 0$. □

We then state the following theorem for the stability of the slower-scale MPC problem **A2**.

Theorem 1. *Assume **A3** is feasible, then:*

*(i) If **A2** is feasible at time k_s for state $x(k_s)$, **A2** is feasible at time k_s+1 for state $x(k_s+1)$ after applying the first elements of the optimal control sequences, i.e., $p^{MPC}(x(k_s))$ and $v^{MPC}(x(k_s))$ to the system.*

(ii) The origin of the MPC system is asymptotically stable in the Lyapunov sense.

Proof. (i) Since **A3** is feasible based on the assumption, **A**3 obtains an optimal solution as it is a linear program. The state consistency constraint (4.22d) can be satisfied after solving **A3** and $s^*(k_s+1|k_s)$ can be reached after applying the solutions of **A3** for the time period $[k_s, k_s+1]$. This enables us to construct a shifted sequence of controls based on the optimal control sequence at time step k_s (i.e., Equation (4.28)) as the feasible solution for next time step k_s+1):

$$\begin{aligned} p(k_s+1) &= [p^*(k_s+1|k_s), p^*(k_s+2|k_s), \ldots, p^*(k_s+\hat{T}|k_s), p(k_s+\hat{T}+1), \ldots, p(k_s+T^s)] \\ v(k_s+1) &= [v^*(k_s+1|k_s), v^*(k_s+2|k_s), \ldots, v^*(k_s+\hat{T}|k_s), v(k_s+\hat{T}+1), \ldots, v(k_s+T^s)]. \end{aligned} \tag{4.29}$$

where $\hat{T}$ is the same as the one in Lemma 1. To prove the feasibility, noticing that $s^*(k_s+1|k_s)$ is the initial condition for the problem at step k_s+1, if we firstly apply the 1 to $\hat{T}$ optimal control sequence of the problem at step k_s, all vehicles at time k_s should have passed the intersection at $\hat{T}$. If there are vehicles that haven't passed the intersection at time $\hat{T}$, those vehicles must be the newly arrived vehicles during step $[k_s, k_s+1]$. The number of newly arrived vehicles $\tilde{n}(k_s+1)$ is finite and usually small. Same to the arguments in Lemma 1, those newly arrived vehicles can be cleared using another $\hat{T}$ time steps. Therefore, we have $x(k_s+T^s|k_s+1) \in \{0\} = \mathbb{X}_{T^s}$, i.e., all vehicles at time step k_s+1 (both existing and newly arrived) can pass the intersection

at the end of the prediction horizon. This implies that problem **A2** is feasible for state $x(k_s+1)$.

(ii) Equation (4.24) shows that for **A2** constructed at k_s, $l(x(k_s|k_s), p(k_s|k_s), v(k_s|k_s)) \geq \Delta T_s$. Otherwise, if $l(x(k_s|k_s), p(k_s|k_s), v(k_s|k_s)) = 0$, all vehicles have passed the intersection at the current step k_s and there is no need for the signal and vehicle control. Considering that $x(k_s)$ is bounded, we can find a function $\alpha_1 \in \mathcal{K}$ such that $l(x(k_s|k_s), p(k_s|k_s), v(k_s|k_s)) \geq \alpha_1(\|x(k_s)\|)$ for all $x \in \mathbb{X}_f(T^s)$. For example, α_1 could be a linear function of the $L1$ norm of $x(k_s)$ with a slop less than ΔT_s. Thus, we have

$$\begin{aligned}\tilde{V}^{\mathrm{MPC}}(x(k_s), \tilde{x}(k_s, N)) &\geq V^{\mathrm{MPC}}(x(k_s)) \geq l(x(k_s), p^{\mathrm{MPC}}(x(k_s)), v^{\mathrm{MPC}}(x(k_s))) \\ &\geq \alpha_1(\|x(k_s)\|).\end{aligned} \tag{4.30}$$

For a neighborhood region of the origin $\mathcal{N} \in \mathbb{X}_f(T^s)$, since $l(x(\hat{k}_s|k_s), p(\hat{k}_s|k_s), v(\hat{k}_s|k_s))$ is bounded and non-increasing as $\hat{k}_s$ increases from k_s to $T^s + k_s - 1$, we can find a function $\tilde{\alpha}_2 \in \mathcal{K}$ such that $l(x^*(\hat{k}_s|k_s), p^*(\hat{k}_s|k_s), v^*(\hat{k}_s|k_s)) \leq \tilde{\alpha}_2(\|x^*(k_s|k_s)\|)$ for any optimal $p^*(k_s), v^*(k_s)$, initial state $x^*(k_s|k_s) \in \mathcal{N}$, and $\hat{k}_s = k_s, k_s+1, \ldots, T^s + \hat{k}_s - 1$. For example, let $\hat{L}$ be the upper bound of l which represents the production of the total physical vehicle capacity of the roads around the intersection and the slower time interval, α_2 could be a linear function of the $L1$ norm of $x(0|k_s)$with a slop larger than $\hat{L}\Delta T_s$. In addition, $\tilde{\alpha}_2(.)$ can be extended to satisfy $\tilde{n}(k_s, N)\Delta T_s \leq \tilde{\alpha}_2(\|\tilde{x}(k_s, N)\|)$ by increasing the slope based on the upper bound of the number of arrived vehicles during $[k_s, k_s + N]$, which should be a finite value. Thus, we have

$$\begin{aligned}\tilde{V}^{\mathrm{MPC}}(x(k_s), \tilde{x}(k_s, N)) &= V^{\mathrm{MPC}}(x(k_s)) + \tilde{n}(k_s, N)T^s\Delta T_s \\ &= \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} l(x(\hat{k}_s|k_s), p(\hat{k}_s|k_s), v(\hat{k}_s|k_s)) + \tilde{n}(k_s, N)\Delta T_s \\ &\leq 2T^s\tilde{\alpha}_2(\|x(k_s), \tilde{x}(k_s, N)\|).\end{aligned} \tag{4.31}$$

Therefore, we have $\tilde{V}^{\mathrm{MPC}}(x(k_s)) \leq \alpha_2(\|x(k_s)\|)$ with $\alpha_2(.) = 2T^s\tilde{\alpha}_2(.)$. By optimality,

we have

$$\begin{aligned}
&\tilde{V}^{\mathrm{MPC}}(x(k_s+1),\tilde{x}(k_s+1,N)) - \tilde{V}^{\mathrm{MPC}}(x(k_s),\tilde{x}(k_s,N)) \\
&= J(x(k_s+1),p^*(k_s+1),v^*(k_s+1)) - J(x(k_s),p^*(k_s),v^*(k_s)) \\
&\quad + (\tilde{n}(k_s+1,N) - \tilde{n}(k_s,N))\, T^s \Delta T_s \\
&\leq J(x(k_s+1),p(k_s+1),v(k_s+1)) - J(x(k_s),p^*(k_s),v^*(k_s)) \\
&\quad + (\tilde{n}(k_s+1,N) - \tilde{n}(k_s,N))\, T^s \Delta T_s.
\end{aligned} \tag{4.32}$$

As discussed earlier (the paragraph after Equation (4.27)), N is a large integer, we have $\tilde{n}(k_s+1,N) - \tilde{n}(k_s,N) \to -\tilde{n}(k_s,1)$. Thus, we have

$$\begin{aligned}
&\tilde{V}^{\mathrm{MPC}}(x(k_s+1),\tilde{x}(k_s+1,N)) - \tilde{V}^{\mathrm{MPC}}(x(k_s),\tilde{x}(k_s,N)) \\
&\leq J(x(k_s+1),p(k_s+1),v(k_s+1)) - J(x(k_s),p^*(k_s),v^*(k_s)) - \tilde{n}(k_s,1) T^s \Delta T_s.
\end{aligned} \tag{4.33}$$

Notice that the control sequences for k_s+1 is constructed in Equation (4.29) by shifting the control sequences at k_s and all the vehicles at step k_s have crossed the intersection, the states and objective values for the vehicles that are already included at k_s do not change along the prediction horizon. Recall that $\tilde{x}(k_s,1)$ denotes the state vector of the newly arrived vehicles during the time step $[k_s, k_s+1]$. The control sequences at time step k_s+1, i.e., $p(k_s+1)$ and $v(k_s+1)$, are constructed by Equation (4.29), where vehicles at time step k_s should be all cleared at $\hat{T}$. We have

$$\begin{aligned}
&J(x(k_s+1),p(k_s+1),v(k_s+1)) - J(x(k_s),p^*(k_s),v^*(k_s)) \\
&= \sum_{\hat{k}_s=k_s+1}^{T^s+k_s} l(x(\hat{k}_s|k_s+1),p(\hat{k}_s|k_s+1),v(\hat{k}_s|k_s+1)) \\
&\quad - \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} l(x(\hat{k}_s|k_s),p^*(\hat{k}_s|k_s),v^*(\hat{k}_s|k_s)) \\
&= \sum_{\hat{k}_s=k_s+1}^{\hat{T}+k_s} l(x(\hat{k}_s|k_s),p^*(\hat{k}_s|k_s),v^*(\hat{k}_s|k_s)) \\
&\quad + \sum_{\hat{k}_s=k_s+1}^{T^s+k_s} l(\tilde{x}(\hat{k}_s|k_s+1),p(\hat{k}_s|k_s+1),v(\hat{k}_s|k_s+1))
\end{aligned} \tag{4.34}$$

$$
\begin{aligned}
&- l(x(k_s|k_s), p^{\mathrm{MPC}}(k_s|k_s), v^{\mathrm{MPC}}(k_s|k_s)) \\
&+ \sum_{\hat{k}_s=k_s+1}^{\hat{T}+k_s} l(x(\hat{k}_s|k_s), p^*(\hat{k}_s|k_s), v^*(\hat{k}_s|k_s)) \\
&= -l(x(k_s|k_s), p^{\mathrm{MPC}}(k_s|k_s), v^{\mathrm{MPC}}(k_s|k_s)) \\
&+ \sum_{\hat{k}_s=k_s+1}^{T^s+k_s} l(\tilde{x}(\hat{k}_s|k_s+1), p(\hat{k}_s|k_s+1), v(\hat{k}_s|k_s+1)).
\end{aligned}
$$

where $l(\tilde{x}(\hat{k}_s|k_s+1), p(\hat{k}_s|k_s+1), v(\hat{k}_s|k_s+1))$ is the objective value for the newly arrived vehicles during $[k_s, k_s+1]$ at time step $\hat{k}_s$. Recall that $\tilde{n}(k_s, 1)$ denotes the total number of vehicles arrived at that duration, the upper bound of $l(\tilde{x}(\hat{k}_s|k_s+1), p(\hat{k}_s|k_s+1), v(\hat{k}_s|k_s+1))$ at any time step $\hat{k}_s$ is $\tilde{n}(k_s, 1)\Delta T_s$, i.e., all newly arrived vehicles haven't cross the intersection at $\hat{k}_s$. Thus, we have

$$
\sum_{\hat{k}_s=k_s+1}^{T^s+k_s} l(\tilde{x}(\hat{k}_s|k_s+1), p(\hat{k}_s|k_s+1), v(\hat{k}_s|k_s+1)) \leq \tilde{n}(k_s, 1)T^s\Delta T_s. \tag{4.35}
$$

Combine Equation (4.33) (4.34) (4.35), we have

$$
\begin{aligned}
&\tilde{V}^{\mathrm{MPC}}(x(k_s+1), \tilde{x}(k_s+1, N)) - \tilde{V}^{\mathrm{MPC}}(x(k_s), \tilde{x}(k_s, N)) \\
&\leq J(x(k_s+1), p(k_s+1), v(k_s+1)) - J(x(k_s), p^*(k_s), v^*(k_s)) - \tilde{n}(k_s, 1)T^s\Delta T_s \\
&= -l(x(k_s|k_s), p^{\mathrm{MPC}}(k_s|k_s), v^{\mathrm{MPC}}(k_s|k_s)) \\
&\quad + \sum_{\hat{k}_s=k_s+1}^{T^s+k_s} l(\tilde{x}(\hat{k}_s|k_s+1), p(\hat{k}_s|k_s+1), v(\hat{k}_s|k_s+1)) - \tilde{n}(k_s, 1)T^s\Delta T_s \\
&\leq -l(x(k_s|k_s), p^{\mathrm{MPC}}(k_s|k_s), v^{\mathrm{MPC}}(k_s|k_s)) + \tilde{n}(k_s, 1)T^s\Delta T_s - \tilde{n}(k_s, 1)T^s\Delta T_s \\
&= -l(x(k_s|k_s), p^{\mathrm{MPC}}(k_s|k_s), v^{\mathrm{MPC}}(k_s|k_s)) \\
&< 0.
\end{aligned} \tag{4.36}
$$

Therefore, $\tilde{V}^{\mathrm{MPC}}$ is a Lyapunov function for problem **A2**. Based on the Lyapunov stability theorem, Theorem 1 holds.

□

4.4 *Numerical experiments*

In this section, we first illustrate the settings of the numerical experiment including the simulation platform, the intersection geometry and volume scenarios, and the fuel consumption model. Then, we conduct sensitivity analysis of key MPC parameters to test the performance of the scheme. We next show the overall performance (average fuel consumption, delay, and queue length) and detailed trajectory analysis of the proposed multi-scale model and a benchmark method.

4.4.1 Experiment settings

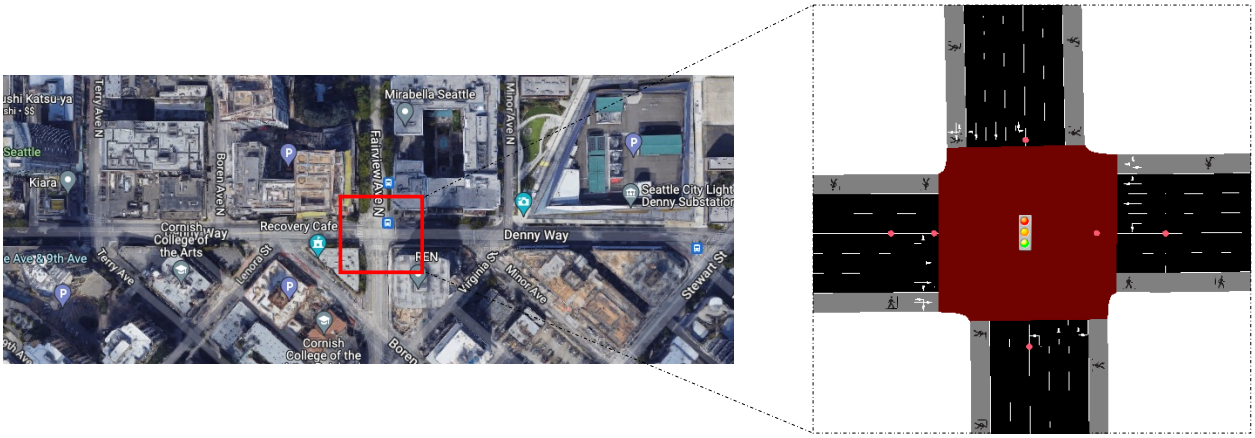

Figure 4.3: The intersection at the Fairview Avenue and the Denny Way, Seattle

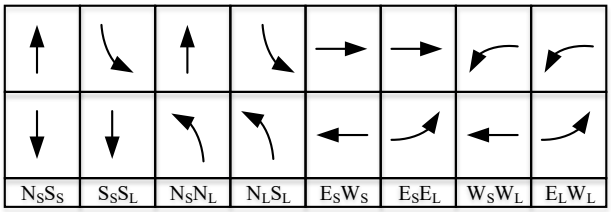

Figure 4.4: The candidate signal phases

We set up the simulation model based on a real-world intersection at the Fairview Avenue and the Denny Way (one of the major signalized intersections in downtown Seattle, USA) and use it to test the proposed multi-scale control model. The intersection is shown in Figure 4.3. We made two modifications in building the simulation model. First, as this study focuses on SVCC, the pedestrians are neglected (we will discuss how to consider pedestrians in Section 7). Second, for the original intersection, the lengths of the four legs vary from $50m$ to $250m$. To simplify the geometry to better test CAVs, we set the length of each leg the same ($300m$ is used here) and the CAV communication range as $200m$. After the revision, the intersection becomes a typical four-leg intersection with four approaches (each with three incoming lanes including a dedicated left-turn lane) and twelve movements. There are eight candidate phases for the studied intersection, which is shown in Figure 4.4 (see also Roess et al. [2004]). As an illustration, $\mathrm{N_S S_S}$ represents the "Northbound & Southbound Straight" phase.

Inspired by Vellamattathil Baby et al. [2020], we use the same third degree polynomial function of velocity and acceleration as the fuel consumption model, which is fitted based on the experimental data for Ford Focus (Ford [2021]):

$$\begin{aligned} f_{i,j}(k_f) = {} & 0.2736 + 0.05993 v_{i,j}(k_f) + 0.3547 a_{i,j}(k_f) - 0.005804 v_{i,j}^2(k_f) \\ & + 0.01787 v_{i,j}(k_f) a_{i,j}(k_f) + 0.06633 a_{i,j}^2(k_f) + 0.0001888 v_{i,j}^3(k_f) \\ & + 0.001959 v_{i,j}^2(k_f) a_{i,j}(k_f) + 0.02447 v_{i,j}(k_f) a_{i,j}^2(k_f) - 0.04892 a_{i,j}^3(k_f). \end{aligned} \tag{4.37}$$

We compare the proposed multi-scale method with the traditional actuated signal control method. Actuated signal control relies on loop detectors to obtain information of surrounding vehicles, and extend the green time if there are incoming vehicles under the current phase, until the green time reaches the maximum green time or an arrival gap is detected. It can adjust the green time for each phase based on real-time traffic conditions using a simple logic, making it the most popular method in real-world advanced traffic signal control. Hereafter in this study, the actuated signal control and the proposed method are denoted as Actuated and Multiscale, respectively. Note that this study is mainly to propose the multi-scale control

framework, its application to SVCC, and the MPC scheme. Thus we only compare it with actuated signal control to illustrate that the proposed framework can produce reasonable performance. The sequence order of the Actuated signal control is $N_S S_S \rightarrow N_L S_L \rightarrow E_S W_S \rightarrow E_L W_L$.

For each leg around the intersection, the real volumes are around 250 veh/hour for normal hours and 400 veh/hour for peak hours. We consider these values as the medium volume scenario, and vary the values to create the low and high volume scenarios (which are shown in Table 4.2). In addition, we create asymmetric volume scenarios by increasing the eastbound and northbound volumes and reducing the westbound and southbound volumes to test the performance under unbalanced volumes. The simulation period is set to 30 minutes. In order to capture dynamic volumes, the volumes increase from minimum to maximum linearly through the simulation period.

Table 4.2: Volume settings

	symmetric						asymmetric					
	low		medium		high		low		medium		high	
	min	max	min	max	min	max	min	max	min	max	min	max
WE straight	100	200	250	400	400	600	250	400	400	600	550	800
WE left/right turns	33	67	83	133	133	200	83	133	133	200	183	267
SN straight	100	200	250	400	400	600	250	400	400	600	550	800
SN left/right turns	33	67	83	133	133	200	83	133	133	200	183	267
EW straight	100	200	250	400	400	600	125	200	200	300	275	400
EW left/right turns	33	67	83	133	133	200	41	67	67	100	91	133
NS straight	100	200	250	400	400	600	125	200	200	300	275	400
NS left/right turns	33	67	83	133	133	200	41	67	67	100	91	133

We integrate Python, General Algebraic Modeling System (GAMS), and Simulation of Urban Mobility (SUMO) as the simulation platform. SUMO is used to build the simulation

environment and visualize the traffic states, GAMS is used to solve the slower-scale problem **A2** and faster-scale problem **A3**, and Python is used to bridge the SUMO and GAMS. The Solving Constraint Integer Programs (SCIP) is used to solve the slower-scale problem in GAMS, and the Interior Point OPTimizer (IPOPT) is used to solve the faster-scale problem.

4.4.2 Sensitivity analysis of the MPC scheme

First, it should be noted that the original multi-scale problem **A1** for the single intersection SVCC cannot be solved in acceptable time (on a computer with Intel(R) Core(TM) i9-9900K CPU and 32GB RAM) due to the large number of binary variables. This is also the reason why we propose the MPC-based scheme in this study. There are essentially two key parameters that might influence the control performance of the MPC scheme: the prediction horizon of the slower-scale problem (T^s) and the number of slower steps used by the faster-scale problem (h). Based on our observation, parameter h has a minor effect on the control performance, as long as $h >= 2$. This is because the positions have been determined by the slower-scale problem at those critical time points, leaving less freedom for the faster-scale problem to increase the performance by using a long optimization horizon. The performance gain when h increase from 1 to 2 is achieved by the fact that the faster-scale problem can design better trajectories (especially the speed profiles) if more than one critical points are given. In this study, we will use $h = 3$.

For T^s, it is intuitive that the longer the prediction horizon is, the better the control performance may be. However, longer prediction horizon also increases the size of **A2** and leads to longer computation time. Thus, a balance is needed between the performance and the computation cost when choosing a proper T^s. We next show the control performance of the proposed algorithm under different prediction horizon T^s in Figure 4.5. This is tested under the symmetric volume setting, different line types represent different volume levels, i.e., low, middle, and high. The performance is evaluated by the average fuel consumption, average waiting time, average queue length under the symmetric volumes, and the run time of the overall algorithm. We collect the fuel consumption calculated by the SUMO internal

fuel consumption model (the external model shows similar results thus we only present the fuel calculated by the SUMO model here for brevity). The waiting time of a vehicle is calculated by the total time when the speed is less than $0.1m/s$ during the trip. The queue length of a lane at a specific time instant is calculated as the position of the last vehicle with speed less than $0.1m/s$. For the run time, we use the run time of $T^s = 4$ as the base for each volume level, and use the relative increment of the run time under each prediction horizon to evaluate the run time performance.

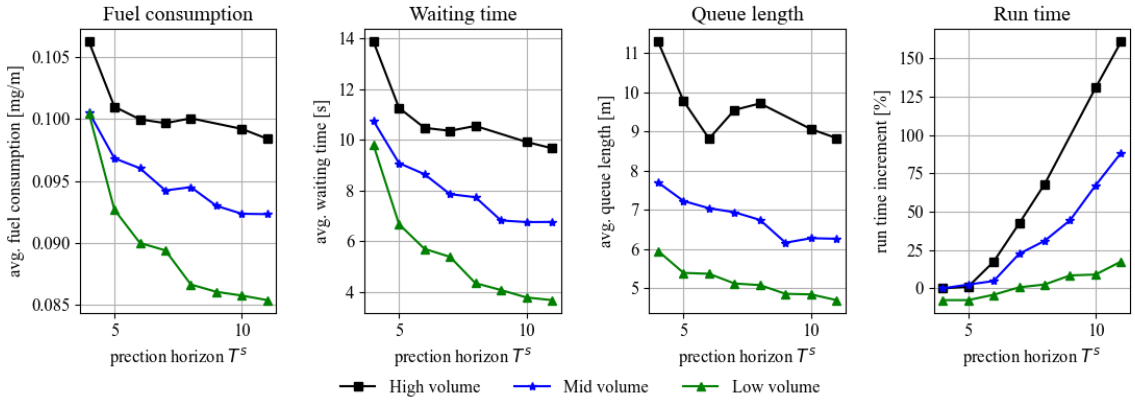

Figure 4.5: Control performance under different prediction horizons, symmetric volumes

As shown in Figure 4.5, in general, for all three volume levels, as the prediction horizon T^s increases, the overall control performance improves regarding the fuel consumption, waiting time, and queue length. However, the run time also increases with T^s. In particular, the control performance improves dramatically when T^s increases from 4 to 6, then improves relatively slowly with possible fluctuations (e.g., for queue lengths). For each volume level, the run time does not change too much as T^s increases from 4 to 5, then increases a bit from 5 to 6, and increases nearly linearly after that. Therefore, $T^s = 6$ can be regarded as an elbow point of the algorithm, which can best balance the control performance and the computation time. We further test the asymmetric volume settings and similar trends can be observed, with more fluctuations of the performances as T^s increases. Based on the

numerical results, we set $T^s = 6$ in the numerical experiments in this study.

4.4.3 Overall performance analysis

Table 4.3 shows the performance of the two control methods. We use the same performance measures as shown above, plus the average time loss (i.e. delays). The time loss of a vehicle is calculated as the time lost due to travelling at speed below the maximum speed. In addition, apart from the fuel consumption calculated by the SUMO internal fuel model, we also calculate the fuel consumption by the external model Equation (4.37). We note the external model as Avg. fuel I and the SUMO internal fuel model as Avg. fuel II. It turns out that, the two fuel consumption calculation methods, despite the discrepancy in absolute values, produce similar values and trends. All changes in the Multiscale methods are calculated compared with the Actuated method.

Table 4.3 shows the performance of the three methods considering average fuel consumption, average waiting time, average time loss, and average queue length under the symmetric volumes. We calculate the fuel consumption by two different methods: the external model Equation (4.37) (noted as Avg. fuel I), and SUMO internal fuel model (noted as Avg. fuel II). The waiting time of a vehicle is calculated by the total time when the speed is less than $0.1m/s$ during the trip. The time loss of a vehicle the time lost due to travelling at speed below the maximum speed. The queue length of a lane at a specific time instant is calculated as the position of the last vehicle with speed less than $0.1m/s$. All changes in Intelligent and Multiscale methods are calculated compared with the Actuated method.

Volume type	Metrics	Symmetric volumes			Asymmetric volumes		
		Actuated	Multiscale		Actuated	Multiscale	
			value	change		value	change
low	Avg. fuel I (mg/m)	0.11	0.08	-27.27%	0.11	0.08	-27.27%
	Avg. fuel II (mg/m)	0.11	0.09	-18.18%	0.12	0.09	-25.00%
	Avg. waiting time (s)	12.02	6.15	-48.84%	13.48	6.65	-50.67%

	Avg. time loss (s)	23.58	11.85	-49.75%	26.02	12.84	-50.65%
	Avg. queue length [m]	6.47	5.24	-19.01%	8.21	5.87	-28.50%
medium	Avg. fuel I (mg/m)	0.11	0.08	-27.27%	0.11	0.08	-27.27%
	Avg. fuel II (mg/m)	0.12	0.09	-25.00%	0.13	0.10	-22.71%
	Avg. waiting time (s)	16.31	7.89	-51.62%	17.44	8.50	-51.26%
	Avg. time loss (s)	29.57	15.02	-49.21%	31.54	15.9	-49.59%
	Avg. queue length [m]	10.19	6.63	-34.94%	12.30	7.83	-36.34%
high	Avg. fuel I (mg/m)	0.13	0.09	-35.22%	0.13	0.09	-30.77%
	Avg. fuel II (mg/m)	0.18	0.10	-45.31%	0.17	0.10	-40.89%
	Avg. waiting time (s)	41.31	10.89	-73.64%	33.96	11.06	-67.43%
	Avg. time loss (s)	28.18	19.22	-31.80%	50.96	19.62	-61.50%
	Avg. queue length [m]	28.19	9.81	-65.20%	26.56	10.43	-60.73%

Table 4.3: Experiment results

For the low volume scenario, compared with the Actuated method, the Multiscale method can improve the performance considering the fuel consumption, waiting time, time loss, and queue length. For the medium volume scenario, the level of performance improvement of the Multiscale method is similar to that the low volume scenario, but with small increment for all metrics. For the high volume scenario, the performance improvement becomes even higher. For example, the fuel-saving performance improves by 35.22% and 45.31%, respectively, by the two fuel consumption calculation methods. Same pattern can also be observed in the table for waiting time, time loss, and queue length. Another interesting observation is that although we use the external fuel consumption model (Equation (4.37)) in the multi-scale framework due to its calculation efficiency, the generated control strategies perform fairly well when evaluated by either the external fuel consumption model or the internal SUMO fuel consumption model. Similar trends can also be found under the asymmetric volume

scenarios from Table 4.3; we omit the detailed discussions for brevity.

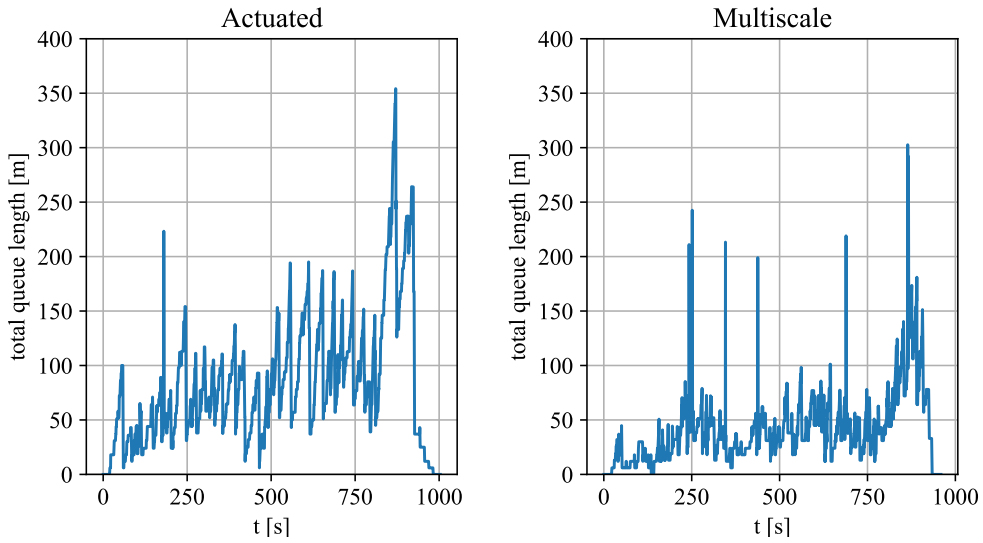

Figure 4.6: Total queue length around the intersection

To have a deeper understanding of the control performance, we show the total queue length around the intersection during the whole simulation in Figure 4.6. The total queue length is calculated by summing up the queue length of all the 12 incoming lanes around the intersection. It is shown that, under Multiscale control, the overall queue length is less than that under Actuated control. Another observation is that the Multiscale method tends to generate more large fluctuations. But the absolute fluctuation values are still less (or at the same level) than normal values under the Actuated control. Nevertheless, the fluctuations usually vanish within a very short time and the total queue length will become normal. In addition, the Multiscale method can clear the intersection sooner than the Actuated method. The good traffic control performance of the Multiscale method comes from two aspects. First, using the Multiscale method, the vehicles can be better controlled by the faster-scale vehicle control problem. However, there is no vehicle control for the Actuated signal control. Second, the Multiscale usually generates better signal plans compared with

the Actuated method by selecting signal phases that are more adaptive to real-time traffic conditions. To show this, we plot the duration distributions of each phase (i.e., Figure 4.4) in Figure 4.7, and the usage proportion of each phase in Figure 4.8. The phase number in these two figures are the index of the phases in Figure 4.4, starting from the leftmost (indexed as 1) to rightmost (indexed as 8). For the duration distribution, we collect the duration of each phase whenever it is used, and show the results using box plot. For the usage proportion, we calculate the total duration of each phase and divide it with the total simulation time. It can be shown from these two figures that the Multiscale method is more flexible than the Actuated signal control. First, the Multiscale method utilizes all the 8 phases to manage the traffic based on corresponding traffic conditions. On the contrast, the Actuated signal control is limited by the design and can only use 4 phases to manage the traffic. Second, the Multiscale method could use more flexible phase duration such as 5 seconds to deal with minor traffic (but could have longer waiting time). The Multiscale method could terminate a phase even there are vehicles coming for the current phase to serve other phases which could provide larger benefits. The Actuated method, however, could be easily triggered by incoming vehicles and fail to serve other phases adaptively.

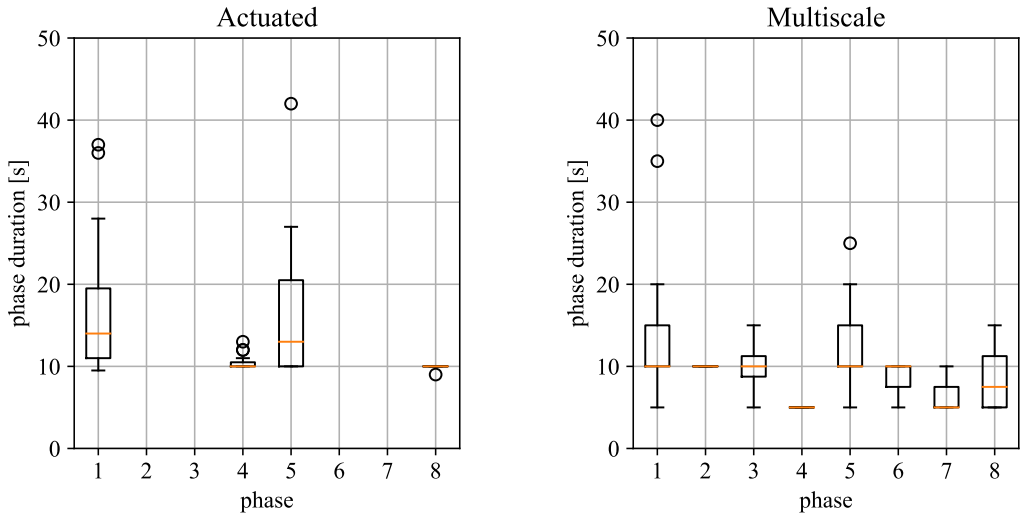

Figure 4.7: Duration of each phase

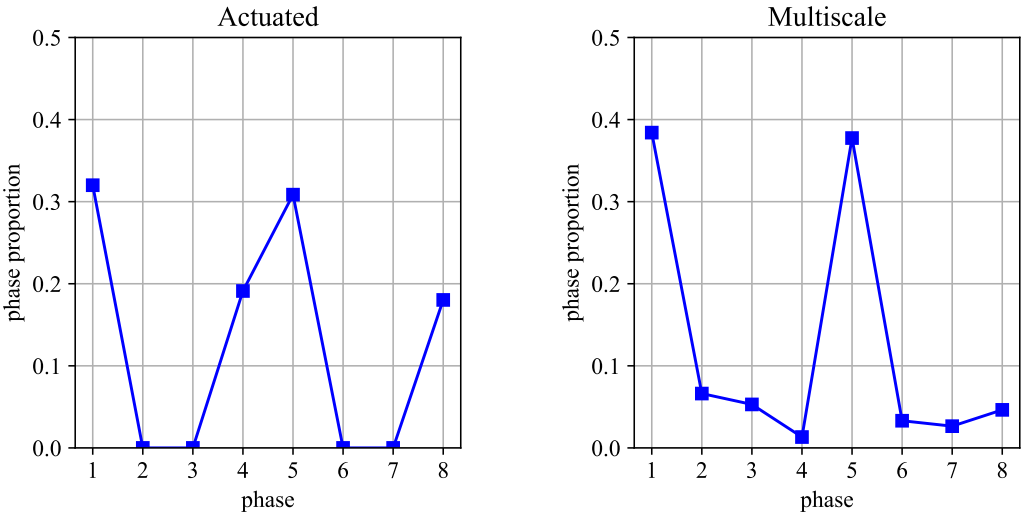

Figure 4.8: Usage proportion of each phase

In summary, compared with the Actuated method, the Multiscale method can reduce the fuel consumption, waiting time, time loss, and queue length under all volume scenarios

tested in the table, and the reductions increases as the volume increases.

4.4.4 Vehicle trajectory analysis

In order to better understand the performance of different control methods, we firstly show the aggregated trajectories of all WE straight vehicles under the high symmetric volume in Figure 4.9. For each vehicle, the time when it enters the communication range is set as zero. We plot every vehicle's trajectory in the same figure and refer it as the "aggregated trajectories," which can help clearly show the overall performance of all vehicles. The figure shows that the Multiscale method can largely reduce the delay and the maximum queue length of WE straight vehicles. In fact, we can observe similar trends for the vehicles of other movements, which are omitted here for brevity. Compared with the Actuated method, the Multiscale method can better balance the signal phases and timings for different movements, thus reducing the travel times and queue lengths.

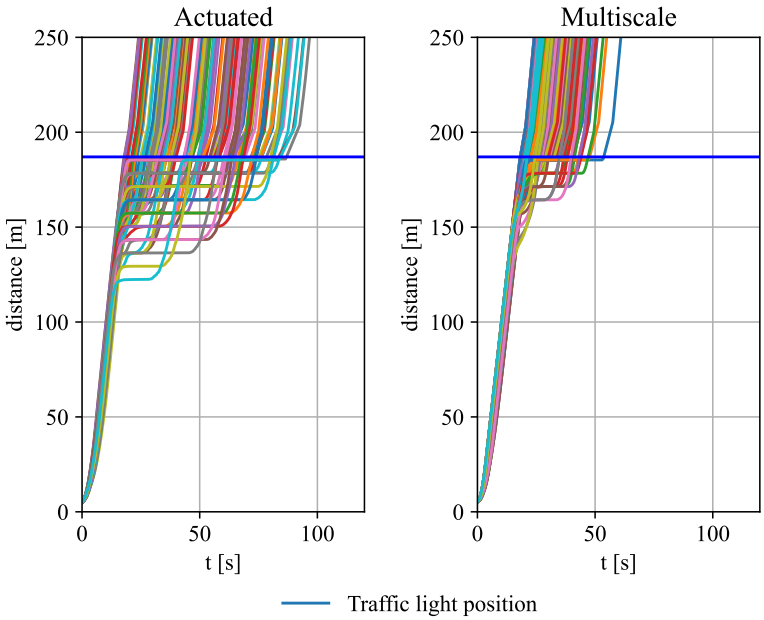

Figure 4.9: Trajectories of all WE straight vehicles for different control methods

Another observation is that, although we did not explicitly include any vehicle platoon model in the Multiscale method (in either **A1** or **A2**), there are platoon patterns showed in the results. To see this, we plot every single trajectory of the WE straight vehicles during the first 600 seconds under the Multiscale control in Figure 4.10. In this figures, the red dotted lines indicate that the signal phase for the WE straight vehicles are red (thus the blank intervals indicate green phase). Note that for the Multiscale method, any phase in the 8 candidate phases (Figure 4.4) can be selected. There are some green phases without any trajectory, e.g., around $80s$. This is because there are two phases that the WE vehicles can pass the intersection, i.e., $\mathrm{E_S W_S}$ and $\mathrm{E_S E_L}$, the blank green phases are used for left-turning vehicles. We can see that, from time to time, vehicles did arrive in platoons, e.g., around $t = 120s, 180s, 500s, 550s$, etc. The Multiscale method designs signal timing in such a way that vehicles in the same platoon often passes the intersection (and waits in the same queue if needed) during the same green time. This implies that the method can somehow capture the arriving patterns of the vehicles, and the arrival platoons are usually maintained when passing the intersection.

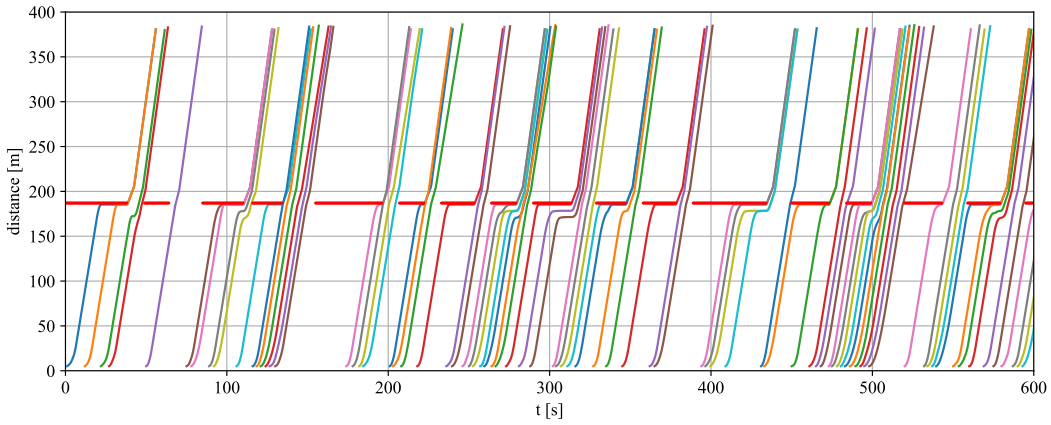

Figure 4.10: Trajectories of WE straight vehicles during first 600 seconds for the Multiscale method

We then show how the Multiscale method reduce the fuel consumption of individual vehicles. Observing that most fuel-savings are achieved during the deceleration process before the stop line, we show in Figure 4.11 the speed and instantaneous fuel rate (calculated by the default SUMO fuel consumption model) of two selected vehicles controlled by the Actuated and Multiscale methods respectively. The negative values of the horizontal axis indicate the time before the vehicle completely stops. It can be shown from the speed profile that, under the Multiscale method, the vehicle decelerates much earlier than the vehicle under the Actuated method (which is the default car-following model in SUMO). During the deceleration period, the fuel consumption is nearly zero. Under the Actuated method, the vehicle maintains a relatively high speed with fluctuation and decelerate more abruptly compared with the vehicle under the Multiscale method, which leads to more fuel consumption.

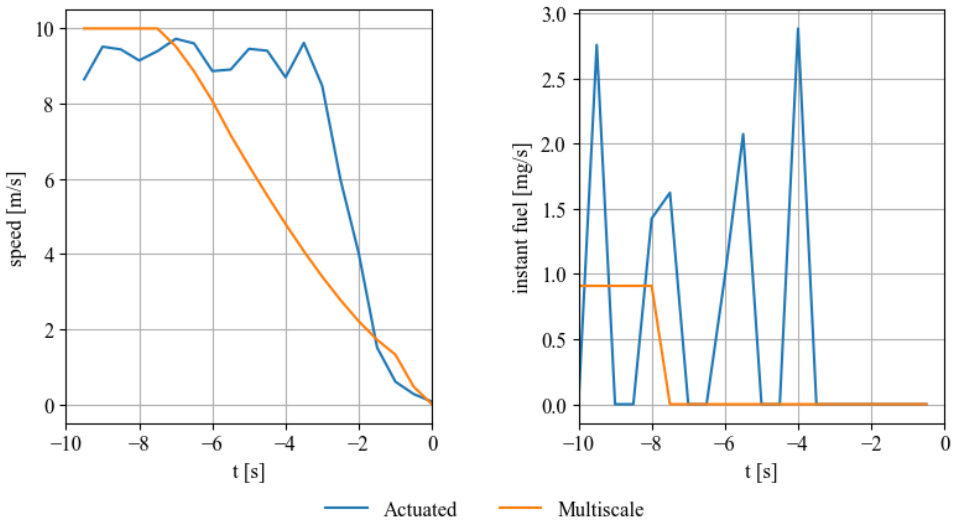

Figure 4.11: Trajectories of two vehicles for different control methods

There are only two vehicles illustrated in Figure 4.11. To have a better understanding of the general behavior, we collect the trajectories of all WE straight vehicles with stops before

the stop-line and plot the heat-map of the trajectories in Figure 4.12. The precision of the time (i.e., horizontal axis) is $0.5s$ and the precision of the speed is $1m/s$. The color of a cell in Figure 4.12, say the i-th time window (column) and the j-th speed (row), represents the "density" of trajectories of that cell, i.e., the number of trajectories locates in the cell divided by the total number of trajectories locates at the i-th time window (across all speed windows). The sum of cell values of each time window is thus 1. A high value of a cell (with certain speed) indicates that, at a specific time window, there are more trajectories with the speed at the corresponding speed window.

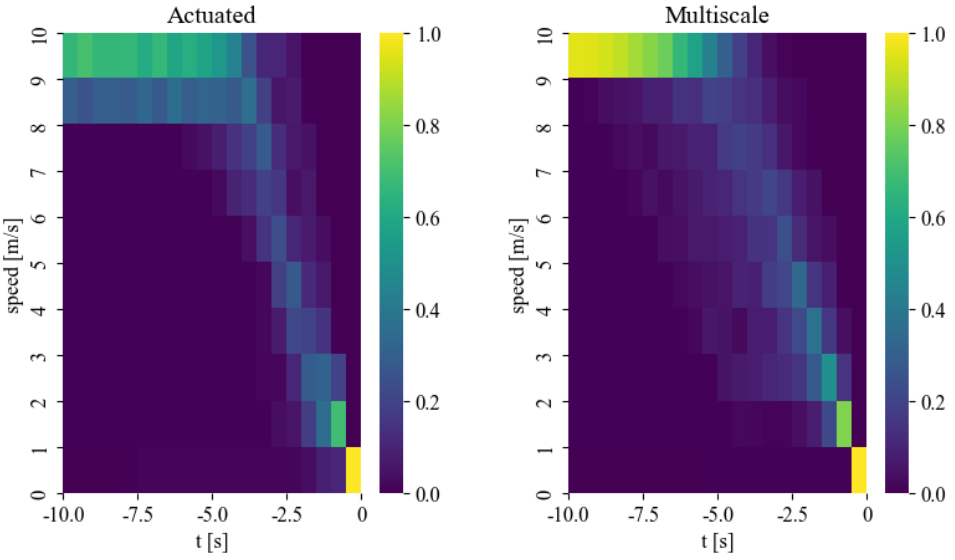

Figure 4.12: Heatmap of the vehicle speed before stopping of all WE straight vehicles for different control methods

There are several observations from Figure 4.12. First, the upper right boundaries of the Actuated and Multiscale methods are similar, indicating that the most radical deceleration processes generated by the Multiscale method is similar to those generated by the Actuated method. Second, under the Actuated method, the deceleration processes are quite concentrated, which is similar to the one in Figure 4.11. However, under the Multiscale method,

there are more trajectories locate at the left side compared with the Actuated method, indicating that there are more cases where the vehicles decelerate earlier to save fuel. To have a more intuitive understanding, we plot the similar heat-map for the instantaneous fuel consumption rate in Figure 4.13. It is shown that the fuel consumption rate under the Actuated method varies a lot before the stop, while the fuel consumption rate under the Multiscale method concentrates on smaller values, which is consistent to the findings from Figure 4.11 and Figure 4.12.

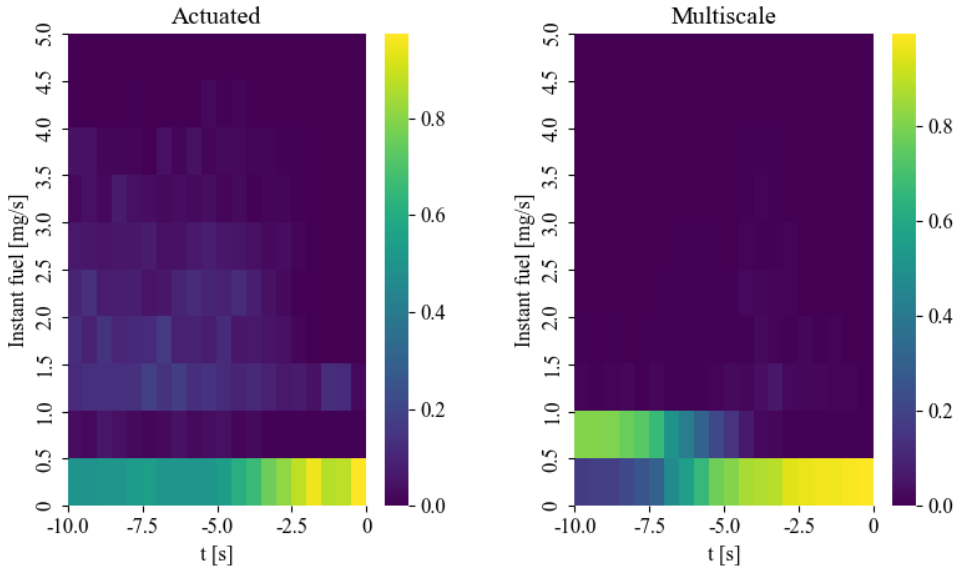

Figure 4.13: Heatmap of the instant fuel consumption rate before stopping of all WE straight vehicles for different control methods

4.5 Summary

In this section, we modeled the SVCC problem as a two-scale problem. In order to efficiently solve the formulated problem, we decomposed the integrated two-scale control problem to two separate sub-problems (i.e., the faster-scale and slower-scale problems) based on the scale and features of the SVCC problem. We also analyzed the stability of the corresponding

MPC control scheme developed based on the slower-scale problem. The numerical experiment results showed that, compared with traditional actuated signal control, the proposed method could largely reduce the delay as well as the fuel consumption.

Chapter 5

MULTI-SCALE SVCC UNDER MULTIPLE INTERSECTIONS WITH MIXED TRAFFIC FLOW

In practice, urban traffic system usually consists of multiple intersections, i.e., on a traffic network. Network-wide UTC is different from the single intersection control due to the need to coordinate among different intersections. In addition, while CAVs are currently under rapid developments and testing, it is expected that large-scale deployment of CAVs will take some time and it will take even a longer time for the whole vehicle fleet to switch to all CAVs or a high penetration of CAVs. Therefore, for a considerable period of time in the future, we will likely have to deal with UTC with mixed traffic flow of CAVs and human-driven vehicles (HDVs) on urban traffic networks. Section 4 focuses on single intersection and full CAV penetration. This section aims to extend the multi-scale SVCC in Section 4 to network-wide multi-scale SVCC with the mixed traffic flow.

As reviewed in Section 2, the literature on network-wide multi-scale SVCC with mixed traffic flow is quite sparse. Researchers usually tackle this problem separately by focusing on only one scale, i.e., either network-wide traffic signal control with CAVs (usually under full CAV penetration, see Li and Ban [2020], Li et al. [2016b], Timotheou et al. [2014], Al Islam and Hajbabaie [2017], Zheng et al. [2021]) or CAV control with mixed traffic flow (Hu et al. [2018], Guo et al. [2021], Gong and Du [2018], Wang et al. [2020b]). Methods for the first topic, i.e., network-wide traffic signal control with CAVs, can be broadly categorized as model-based methods and learning-based methods. Each type of the methods can be further categorized as centralized and distributed approaches. In general, model-based methods are usually more robust and stable and learning-based methods often have higher computation efficiency; centralized methods have the potential to guarantee global optimality and dis-

tributed methods can better balance modeling capability and computational efficiency. Note that most of the existing studies regarding network-wide traffic signal control assume 100% CAV penetration in order to obtain the knowledge of the traffic states with a few exceptions (Yang et al. [2021]). For the second topic, i.e., CAV control with mixed traffic flow, studies can be categorized into to two types: flow-based traffic assignment and vehicle-based control. The first approach aims to assign the CAV flows at least at the link level to achieve certain goals such as improve the network-wide traffic performance. The second approach is more microscopic, aiming to design CAV control algorithms with the considerations of surrounding non-CAVs to improve CAVs' and the traffic performance.

In the following of this chapter, we firstly adopt the distributed control as the fundamental scheme for the network-wide control and use the information sharing technique to achieve the coordination among different intersections (see Section 5.2.1). Under 100% CAV penetration, we prove the control stability of the proposed network-wide multi-scale SVCC algorithm. Then, in Section 5.2.2, we use CAV information to estimate HDV information and develop a "safety check" technique to control the CAVs in the mixed traffic flow. Together, we propose a network-wide SVCC framework with mixed traffic flow in Section 5.2.3. Finally, in Section 5.3, we test the proposed multi-scale SVCC algorithms under various traffic scenarios (i.e., a corridor, a synthetic traffic network, and a real-world traffic network) to demonstrate the performance of the proposed algorithm.

5.1 Assumptions and notations

We firstly introduce the notation used in this section in Table 5.1 with most of the key parameters illustrated in Figure 5.1. The notations are based on the notations for single intersection (see Section 4.1). The main difference is that we add an intersection number m to each variable to indicate different intersections. Note that $i = 1, 2, ..., n_j^m$, $j = 1, 2, ..., J^m$, $l = 1, 2, ..., L^m$, where n_j^m is the number of vehicles on lane j of intersection m, J^m is the total number of lanes around intersection m, and L^m is the number of candidate phases of the traffic signal at intersection m. $m = 1, 2, ..., M$, where M is the number of intersections

of the traffic network. In addition, k_s is the time interval index of the slower-scale (i.e., signal control) and k_f is the time interval index of the faster-scale (i.e., vehicle control) .

Table 5.1: Notations

Symbol	Meaning	Type	Unit
$a^m_{i,j}(k_f)$	acceleration of vehicle i on lane j of intersection m at time step k_f	continuous	m/s^2
$v^m_{i,j}(k_f)$	speed of vehicle i on lane j of intersection m at time step k_f	continuous	m/s
$s^m_{i,j}(k_f)$	position of vehicle i on lane j of intersection m at time step k_f, measured from the entering position of the communication range (i.e., the left most vertical dash line in Figure 5.1 for the EB vehicle)	continuous	m
$f^m_{i,j}(k_f)$	fuel rate of vehicle i on lane j of intersection m at time step k_f	continuous	mg/s
$g^m_{i,j}(k_s)$	whether the vehicle i on lane j of intersection m has crossed the stop-line at time step k_s ($g^m_{i,j} = 0$) or not ($g^m_{i,j} = 1$)	binary	1/0
$p^m_l(k_s)$	whether the phase l of intersection m at time step k_s is on ($p^m_l = 1$) or off ($p_l = 0$)	binary	1/0
$r^m_j(k_s)$	whether the lane j of intersection m at time step k_s has the right-of-way ($r^m_j = 1$) or not ($r^m_j = 0$)	binary	1/0
τ_h	time headway	constant	s
d_0	safety distance	constant	m
$G = (V, E)$	the traffic network where V is the set of intersections and E is the set of roads that connect the intersections		
$A^{J\times L}_m$	the mapping function from signal phase to right-of-way of intersection m	constant	
$s_{j,\max}$, $s_{j,\text{safe}}$	length and the safety distance of incoming lane j	constant	m
ΔT_s, ΔT_f	the step lengths of the slower-scale and faster-scale	constant	s
T^s, T^f	the number of slower-scale and faster-scale steps	constant	

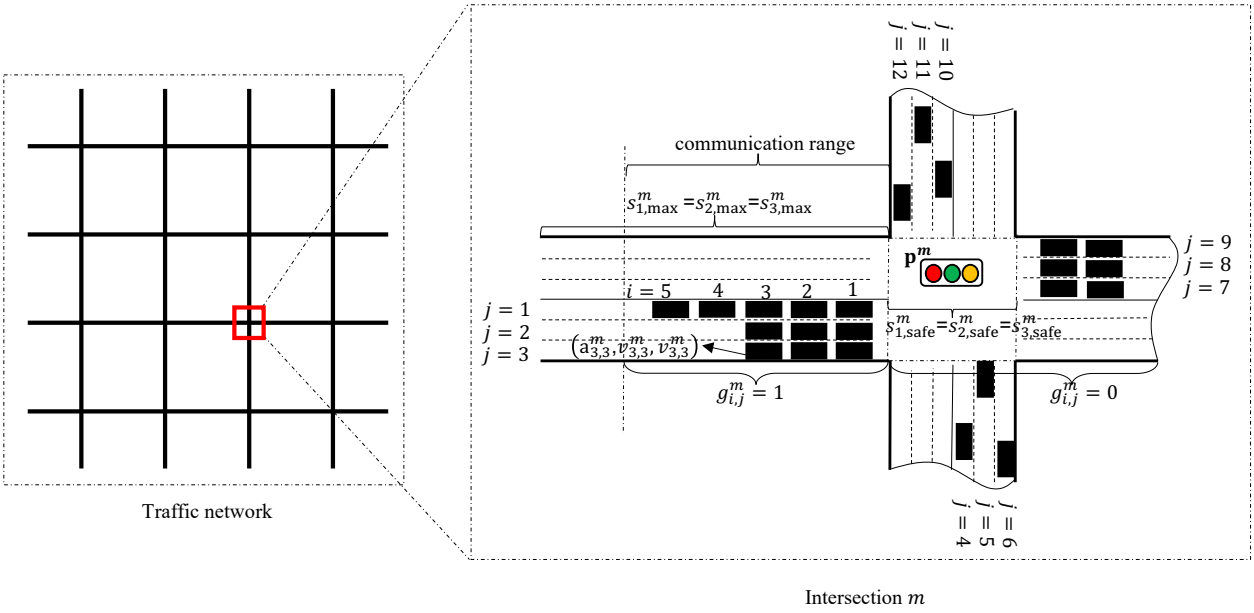

Figure 5.1: Illustration of the notations for the environment of multiple intersections

We make several major assumptions to make the studied problem clear and solvable. First, we assume that information, such as traffic signal plans and surrounding vehicles' states, can be shared among intersections. This is feasible nowadays or in the near future as the rise of CAVs, especially V2I and I2I communication techniques.

Second, we assume that CAVs can be fully observed and controlled, whereas HDVs cannot be controlled. In addition, we need to clearly define the observability of HDVs. One way is to assume that HDVs are fully observable, indicating that we can obtain the information, i.e., position, speed, and acceleration, of all HDVs, but cannot control them. This assumption may be viable when the CV technology reaches full or very high market penetration, which is not practical under the current situation. Inspired by Yang et al. [2021], another way is to assume that we can obtain partial information of HDVs by road sensors like loop detectors and the on-board sensors of CAVs. Considering that modeling the accurate detection process of a CAV is complex, we assume that a CAV can only obtain the information of its front and rear vehicles. In addition, we assume that the number of HDVs can be detected by loop

detectors. In fact, such information can also be collected by other sensors and even on-board sensors of CAVs after complex post-processing. Figure 5.2 illustrates the assumptions.

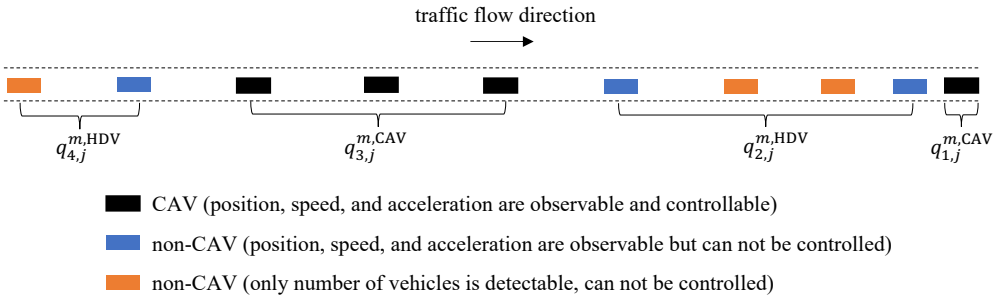

Figure 5.2: Illustration of the assumptions at intersection m, lane j

5.2 Methodology

In this section, we focus on developing the multi-scale control model for the scenario of network UTC with mixed traffic flow of both CAVs and HDVs. The general multi-scale framework in Chapter 3 still applies, and our focus here is on developing the MPC scheme for network UTC with mixed flow. For this, we adopt the distributed control idea and use information sharing techniques to address the traffic signal control problem with multiple intersections on a traffic network. As shown in Figure 5.3, we assume that there are communications among intersections. In particular, for a specific intersection (e.g., the intersection marked by the solid rectangle), the information of neighboring intersections (e.g., intersections marked by dash rectangles) that are directly connected to it is shared. The shared information enable each intersection to generate optimal signal phases considering the incoming traffic flow. We show the detailed methods in Section 5.2.1. For the mixed traffic flow scenario, the fundamental idea is to estimate the states of HDVs first, build the slower-scale problem (i.e, signal control) by treating HDVs as CAVs, build the faster-scale problem only for CAV platoons, and apply the control commands to CAVs with safety check to avoid potential conflicts caused by the uncontrollability of HDVs. We illustrate the detailed methods

in Section 5.2.2. Finally, in Section 5.2.3, we present an integrated framework that combines the two scenarios (i.e., multiple intersections on a network and mixed traffic flow).

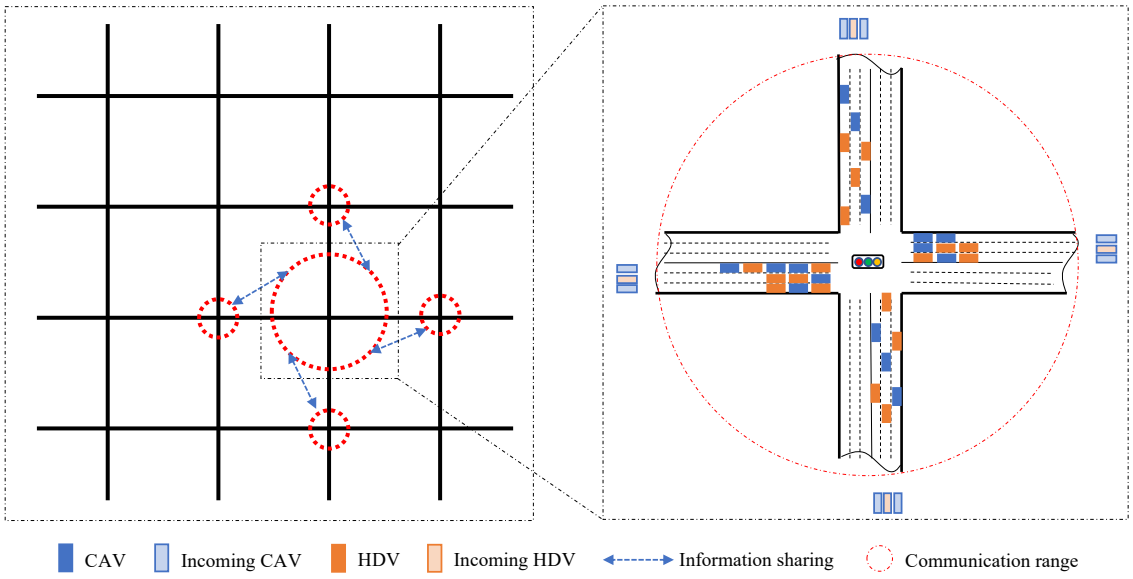

Figure 5.3: An illustration of the scenario and methods

5.2.1 Multiple intersections with full CAV penetration

Information sharing

We focus on extending the original MPC scheme and the two sub-problems, **A2** and **A3**, which were designed for a single intersection and 100% CAVs. Here we discuss how to extend them to multiple intersections on a corridor or a network. In fact, if the distance between two consecutive intersections is long enough, the single intersection SVCC should be able to perform well by operating distributively without coordination, since the impact of upstream flows has already been considered in the multi-scale problem when the flows approach the downstream intersection. If the distance is short, the exit flows of the upstream intersections under current slower-scale time step may enter the communication zones of the downstream intersections during the prediction horizon. If each intersection operates independently, those

flow "overlaps" are not considered, which merits the coordination among intersections.

Sharing information of each single intersection with the neighboring intersections is one of the most promising methods to consider signal coordination (Timotheou et al. [2014], Al Islam and Hajbabaie [2017], Mehrabipour and Hajbabaie [2017]). For a specific intersection, the solution of the slower-scale signal control problem gives the critical trajectory points of all nearby vehicles under a certain future time horizon, which can be used by the neighboring intersections. In this way, a single intersection is able to know the information of oncoming traffic flows, even those flows have not arrived in the communication range. Although the trajectory points are predicted and the prediction error increases as the prediction horizon increases, we can apply the MPC scheme to update the trajectory points step by step so that the errors can be mitigated.

At time step k_s, we can obtain the critical trajectory points (and the corresponding time points) for all vehicles around the intersection m during the whole prediction horizon $[k_s, k_s + T^s]$ by solving the problem **A2** (will be reformulated as **A4**; see below). We can then collect the arrival times (i.e., the times when entering the communication range of intersection m) of the vehicles coming from all neighboring intersections. Apart from the vehicles coming from neighboring intersections, there are two other types of vehicles that need to be considered for the study intersection m: the vehicles that have been already within the communication range, and the vehicles that haven't arrive the communication range but have been on the upstream roads of intersection m. The first type vehicles should be considered during the whole prediction horizon since they are already physically on the surrounding lanes of the intersection. For the second type, we can calculate the arrival times by assuming that they travel with the speed limit. Thus, we can find the arrival times of all vehicles that need to be considered in **A2**, which is denoted as $k_{i,j}^{m,s}$, representing the arrival time step of vehicle i on lane j at intersection m under the slower time scale. In order to limit the scale of the slower-scale problem, we constraint the arrival times by $k_{i,j}^{m,s} \leq T^a$, i.e., we only consider the vehicles that will arrive in the next T^a slower time steps. Correspondingly, we can calculate the arrival position and speed of each vehicle: $s_{i,j}^{m,s}$, $v_{i,j}^{m,s}$.

Due to the consideration of all the incoming vehicles from neighboring intersections, we need to reformulate the slower-scale problem **A2** for intersection m as **A4** below:

$$\min_{p_l^m(\hat{k}_s), v_{i,j}^m(\hat{k}_s)} \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} g_{i,j}^m(\hat{k}_s) \Delta T_s \tag{5.1}$$

subject to

$$s_{i,j}^m(\hat{k}_s) - s_{i,j}^m(\hat{k}_s - 1) = \frac{v_{i,j}^m(\hat{k}_s) + v_{i,j}^m(\hat{k}_s - 1)}{2} \Delta T_s, \quad \forall \hat{k}_s \geq k_{i,j}^{m,s} \tag{5.2a}$$

$$s_{i,j}^m(\hat{k}_s) - s_{i-1,j}^m(\hat{k}_s) \geq \tau_h v_{i,j}^m(\hat{k}_s) + d_0, \quad \forall \hat{k}_s \geq k_{i-1,j}^{m,s} \tag{5.2b}$$

$$-M g_{i,j}^m(\hat{k}_s) \leq s_{i,j}^m(\hat{k}_s) - s_{j,\max}^m \leq M(1 - g_{i,j}^m(\hat{k}_s)), \quad \forall \hat{k}_s \geq k_{i,j}^{m,s} \tag{5.2c}$$

$$g_{i,j}^m(\hat{k}_s - 1) - g_{i,j}^m(\hat{k}_s) - r_j(\hat{k}_s) \leq 0, \quad \forall \hat{k}_s \geq k_s \tag{5.2d}$$

$$A \begin{bmatrix} p_1^m(\hat{k}_s) & p_2^m(\hat{k}_s) & \cdots & p_L^m(\hat{k}_s) \end{bmatrix}^T = \begin{bmatrix} r_1^m(\hat{k}_s) & r_2^m(\hat{k}_s) & \cdots & r_J^m(\hat{k}_s) \end{bmatrix}^T, \quad \forall \hat{k}_s \geq k_s \tag{5.2e}$$

$$\sum_{l=1}^{L^m} p_l^m(\hat{k}_s) = 1, \quad \forall \hat{k}_s \geq k_s \tag{5.2f}$$

$$g_{i,j}^m(\hat{k}_s), p_l^m(\hat{k}_s), r_j^m(\hat{k}_s) \in \{0, 1\}, \quad \forall \hat{k}_s \geq k_s \tag{5.2g}$$

$$s_{i,j}^m(k_s) = s_{i,j}^{m,s}, v_{i,j}^m(ks) = v_{i,j}^{m,s}. \tag{5.2h}$$

The main difference between **A4** and the single intersection slower-scale problem **A2** is that **A4** considers the incoming vehicles from neighboring intersections, with their arrival times denoted as $k_{i,j}^{m,s}$. Note that in **A4**, the prediction horizon T^s is larger or equal to the arrival time limit T^a, i.e., $T^s \geq T^a$. After solving **A4**, we get the critical trajectory points for all CAVs by taking into account of the vehicles that will arrive during the whole prediction horizon. We can then formulate and solve the faster-scale problem **A3** (no change), and apply MPC scheme as shown in Figure 4.2.

Stability analysis of the MPC scheme

In this section, we analyze the stability of the MPC scheme on a network of intersections and 100% CAV penetration. We firstly introduce additional notation used in the analysis. Let a real-valued scalar function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{K}(\varphi \in \mathcal{K})$ if it is

continuous, strictly increasing and $\varphi(0) = 0$. Let $x^m(k_s)$ be the state of problem **A4** for intersection m at time step k_s. Considering that based on the objective function (5.1), the stability of the studied MPC represents the ability of the controller to serve all vehicles (i.e., all vehicles should be able to pass the intersection), in this section, $x^m(k_s)$ consists only of the vehicle states in **A4**, i.e., $g^m_{i,j}(k_s)$ (which is equivalent to vehicle positions, $s^m_{i,j}(k_s)$, due to Equation (5.2c)). Let $\tilde{x}^m(k_s, t_s)$ denote the state vector of the newly arrived vehicles at intersection m that can be took into account when constructing **A4** during the time step $[k_s, k_s + t_s]$, and let $\tilde{n}^m(k_s, t_s)$ denote the total number of those newly arrived vehicles. Assume $p^{m,*}(k_s) = [p^{m,*}(k_s|k_s), p^{m,*}(k_s + 1|k_s) \dots p^{m,*}(k_s + T^s - 1|k_s)]$ and $v^{m,*}(k_s) = [v^{m,*}(k_s|k_s), v^{m,*}(k_s + 1|k_s) \dots v^{m,*}(k_s + T^s - 1|k_s)]$ be the optimal control sequences for the signal and all vehicles around intersection m at time step k_s, let $s^{m,*}(k_s) = [s^{m,*}(k_s|k_s), s^{m,*}(k_s+1|k_s) \dots s^{m,*}(k_s+T^s-1|k_s)]$ be the positions generated by the optimal control variables of **A4**, and let $g^{m,*}(k_s) = [g^{m,*}(k_s|k_s), g^{m,*}(k_s+1|k_s) \dots g^{m,*}(k_s+T^s-1|k_s)]$ be the corresponding vehicle position indicators. Note that all the above optimal control and state variables are for **A4** at time k_s. Denote $p^{m,\mathrm{MPC}}(k_s) = p^{m,*}(k_s|k_s)$ and $v^{m,\mathrm{MPC}}(k_s) = v^{m,*}(k_s|k_s)$. Let $J(x^m(k_s), p^m(k_s), v^m(k_s))$ be the objective function of **A4** and also define $l(x^m(\hat{k}_s|k_s), p^m(\hat{k}_s|k_s), v^m(\hat{k}_s|k_s))$:

$$J(x^m(k_s), p^m(k_s), v^m(k_s)) = \sum_{j=1}^{J} \sum_{i=1}^{n_j} \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} g^m_{i,j}(\hat{k}_s)\Delta T_s, \tag{5.3}$$

$$l(x^m(\hat{k}_s|k_s), p^m(\hat{k}_s|k_s), v^m(\hat{k}_s|k_s)) = \sum_{j=1}^{J} \sum_{i=1}^{n_j} g^m_{i,j}(\hat{k}_s|k_s)\Delta T_s. \tag{5.4}$$

Here $\hat{k}_s$ indicates the step within the prediction horizon. Note that $l(x^m(\hat{k}_s|k_s), p^m(\hat{k}_s|k_s), v^m(\hat{k}_s|k_s))$ is bounded and monotonically non-increasing with $\hat{k}_s$. We have

$$J(x^m(k_s), p^m(k_s), v^m(k_s)) = \sum_{\hat{k}_s=k_s}^{T^s+k_s-1} l(x^m(\hat{k}_s|k_s), p^m(\hat{k}_s|k_s), v^m(\hat{k}_s|k_s)). \tag{5.5}$$

Denote the MPC value function at time step k_s as the minimum objective value:

$$V^{\mathrm{MPC}}(x^m(k_s)) = \inf_{p^m(k_s), v^m(k_s)} J(x^m(k_s), p^m(k_s), v^m(k_s)) = J(x^m(k_s), p^{m,*}(k_s), v^{m,*}(k_s)). \tag{5.6}$$

Denote the augmented MPC value function at time step k_s as

$$\tilde{V}^{\mathrm{MPC}}(x^m(k_s), \tilde{x}^m(k_s, N)) = V^{\mathrm{MPC}}(x^m(k_s)) + \tilde{n}^m(k_s, N) T^s \Delta T_s. \tag{5.7}$$

which is the sum of MPC value for vehicles at time step k_s and the objective value of the vehicles that will arrive in the next N steps, i.e., during $[k_s, k_s + N]$, assuming that all newly arrived vehicles do not cross the intersection during the whole prediction horizon T^s. N is assumed to be a large integer compared with T^s.

We firstly show a Lemma in order to further prove the stability of the problem **A4**.

Lemma 2. *For **A4** constructed for intersection m at time k_s, the prediction horizon T^s ($T^s \geq T^a$) can be specified such that all vehicles at time step k_s can pass the intersection at the end of the prediction horizon, i.e., $g^m_{i,j}(T^s + k_s - 1) = 0$.*

Proof. Consider an optimal sequence of the controls:

$$\begin{aligned} p^{m,*}(k_s) &= [p^{m,*}(k_s|k_s), p^{m,*}(k_s+1|k_s) \dots p^{m,*}(k_s + T^s - 1|k_s)] \\ v^{m,*}(k_s) &= [v^{m,*}(k_s|k_s), v^{m,*}(k_s+1|k_s), \dots v^{m,*}(k_s + T^s - 1|k_s)], \end{aligned} \tag{5.8}$$

obtained by solving **A4**. This optimal control sequence should clear all the vehicles "as soon as possible". This can be proved by contradiction. Assume there exits a time step $\hat{k}_s$ such that a vehicle (i, j) can pass the intersection, i.e., $g_{i,j}(\hat{k}_s) = 0$, with all the constraints satisfied. This actually constitutes a feasible control sequence, denoted as $v(\hat{k}_s|k_s)$. Assume that the optimal control variable $v^{m,*}(\hat{k}_s|k_s)$ prohibits the vehicle to cross the intersection, i.e., $g^{m,*}_{i,j}(\hat{k}_s) = 1$ (and all other variables stay the same). We can easily see that $v^{m,*}(\hat{k}_s|k_s)$ leads to a larger objective function (i.e., due to Equation (4.19)). That indicates the control sequence $v^{m,*}(k_s)$ is not optimal, which conflicts with the assumption that $v^{m,*}(\hat{k}_s|k_s)$ is optimal. Since the number of vehicles that are physically around the intersection at time k_s

is finite, and the number of incoming vehicles that will arrive in time interval $[k_s, k_s + T^a]$ is also finite, all vehicles can be cleared in a finite time, which is denoted as $\hat{T}$. Let $T^s = 2\hat{T}$, we have $g_{i,j}(T^s - 1) = 0$. □

We then state the following theorem for the stability of the slower-scale MPC problem **A4**.

Theorem 2. *If the following assumptions hold:*

*(1) The state consistency between **A3** and **A4** holds.*

(2) The communication ranges of each intersection are exactly the lengths of corresponding incoming roads, i.e., the vehicle control is valid everywhere within the studied network.

(3) The route of each vehicle is determined and fixed at the entrance of the traffic network.

Then:

*(i) If **A4** for intersection m is feasible at time k_s for state $x^m(k_s)$, **A4** for intersection m is feasible at time $k_s + 1$ for state $x^m(k_s + 1)$ after applying the first solutions of the optimal control sequences, i.e., $p^{m,MPC}(x^m(k_s))$ and $v^{m,MPC}(x^m(k_s))$ to the intersection.*

(ii) The origin of the MPC system for every intersection m is asymptotically stable in the Lyapunov sense, and thus, the traffic network is asymptotically stable.

Proof. (i) Based on assumption (2), all vehicles within the network, including the vehicles coming from intersection m's neighboring intersections, can be controlled. In addition, assumption (1) shows that the state consistency constraint (4.22d) can be satisfied after solving and applying the solutions of the faster-scale problem **A3**. Therefore, $s^{m,*}(k_s + 1|k_s)$ generated by **A4** can be reached after applying the solutions of **A3** for the time period $[k_s, k_s + 1]$. This enables us to construct a shifted sequence of controls

based on the optimal control sequence at time step k_s (i.e., Equation (5.8)) as the feasible solution for next time step $k_s + 1$):

$$\begin{aligned} p^m(k_s+1) = [&p^{m,*}(k_s+1|k_s), p^{m,*}(k_s+2|k_s), \ldots, \\ &p^{m,*}(k_s+\hat{T}|k_s), p^m(k_s+\hat{T}+1), \ldots, p^m(k_s+T^s)] \\ v^m(k_s+1) = [&v^{m,*}(k_s+1|k_s), v^{m,*}(k_s+2|k_s), \ldots, \\ &v^{m,*}(k_s+\hat{T}|k_s), v^m(k_s+\hat{T}+1), \ldots, v^m(k_s+T^s)]. \end{aligned} \tag{5.9}$$

where $\hat{T}$ is the same as the one in Lemma 2. To prove the feasibility, noticing that $s^{m,*}(k_s+1|k_s)$ is the initial condition for the problem at step k_s+1, if we firstly apply the 1 to $\hat{T}$ optimal control sequence of the problem at step k_s, all vehicles at time k_s, including the vehicles that are physically on the surrounding lanes and the vehicles coming from neighboring intersections (with arrival time less than or equal to T^a), should have passed the intersection at $\hat{T}$. If there are vehicles that haven't passed the intersection at time $\hat{T}$, those vehicles must be the newly arrived vehicles during step $[k_s, k_s+1]$. The number of newly arrived vehicles is finite and usually small. Same to the arguments in Lemma 2, those newly arrived vehicles can be cleared using another $\hat{T}$ time steps. Therefore, all vehicles at time step k_s+1 (both existing and newly arrived) can pass the intersection at the end of the prediction horizon. This implies that problem **A4** is feasible for state $x^m(k_s+1)$.

(ii) The stability proof for intersection m in the traffic network is similar to the proof for single intersection. The key is to prove that the augmented MPC value function $\tilde{V}^{\mathrm{MPC}}$ is a proper Lyapunov function for problem **A4**, based on Lemma 2 and Theorem 1(i). We omit the details here for the sake of brevity. Based on assumption (3), i.e., the route of each vehicle is determined and fixed at the entrance of the traffic network (which is usually the case in real-world scenarios), there are no "cycle" routes. Thus, vehicles can finally pass all intersections in there routes. This indicates that, given finite traffic demands, the network can be finally cleared, i.e., the MPC scheme is asymptotically

stable for the traffic network.

□

5.2.2 Mixed traffic flow of CAVs and HDVs

A4 and **A3** assume full CAV penetration, i.e., every vehicle is observable and controllable. In this section, we show how to extend this framework to the environment of mixed traffic flow, i.e., limited CAV penetration.

Slower-scale problem: signal control

There are three steps to formulate the slower-scale problem: First, separate the mixed traffic flows into CAV flows and HDV flows. Second, estimate the position of each individual HDV. Third, formulate the slower-scale problem **A4** by treating all vehicles as CAVs. Note that the techniques developed in Section 5.2.1 (i.e., the information sharing) is still used here to achieve the coordination among different intersections.

Separating the mixed traffic flows is simple. For each lane, we know the position, speed, and acceleration of each CAV and their immediately neighboring HDVs. We can group the vehicles on each lane into "platoons" with the same vehicle type, i.e., CAVs or HDVs. As shown in Figure 5.2, the mixed flows on lane j can be separated into $\tilde{n}_j^m$ platoons: $q_{\tilde{i},j}^{m,\mathrm{CAV}}$ or $q_{\tilde{i},j}^{m,\mathrm{HDV}}$ where $\tilde{i} = 1, 2, \ldots, \tilde{n}_j^m$ and CAV/HDV indicates the type of the $\tilde{i}$th platoon.

There are various methods to estimate the states (positions and speeds) of the HDVs within a HDV flow. Simple methods such as the linear interpolation can be very efficient, but are usually low in accuracy. Complex methods such as deep neural networks (Gurghian et al. [2016]) can achieve high accuracy, but usually require huge resources (e.g., data collection, model training, etc.) and increase the model complexity. In this study, considering that the goal of the slower-scale problem is to provide the faster-scale problem with CAVs' critical trajectories, the initial positions of HDVs do not necessarily need to be very accurate. Thus we use the simple yet efficient linear interpolation method to estimate the HDV flows. We know the position, speed, and acceleration of the HDVs that are directly adjacent to the

CAVs, i.e., the blue vehicles in Figure 5.2, from CAVs' detection systems. In addition, it is assumed that we know the number of HDVs within each HDV platoon. We then use linear interpolation to estimate the states of those HDVs. Specifically, assume there are $\hat{n}^m_{\tilde{i},j}$ HDVs in the HDV platoon $q^{m,\text{HDV}}_{\tilde{i},j}$. Let $s^{m,\text{HDV}}_{\tilde{i},j,1}$ and $s^{m,\text{HDV}}_{\tilde{i},j,\hat{n}^m_{\tilde{i},j}}$ be the positions of the first and last HDVs of the HDV flow $q^{m,\text{HDV}}_{\tilde{i},j}$, respectively. Let $v^{m,\text{HDV}}_{\tilde{i},j,1}$ and $v^{m,\text{HDV}}_{\tilde{i},j,\hat{n}^m_{\tilde{i},j}}$ be the speeds of the first and last HDVs of the HDV flow $q^{m,\text{HDV}}_{\tilde{i},j}$, respectively. We have

$$s^{m,\text{HDV}}_{\tilde{i},j,\hat{i}}(k_s) = s^{m,\text{HDV}}_{\tilde{i},j,\hat{n}^m_{\tilde{i},j}} + \frac{\hat{i}-1}{\hat{n}^m_{\tilde{i},j}-1}\left(s^{m,\text{HDV}}_{\tilde{i},j,1} - s^{m,\text{HDV}}_{\tilde{i},j,\hat{n}^m_{\tilde{i},j}}\right) \quad \hat{i} = 2,3,\ldots,\hat{n}^m_{\tilde{i},j}-1 \tag{5.10a}$$

$$v^{m,\text{HDV}}_{\tilde{i},j,\hat{i}}(k_s) = v^{m,\text{HDV}}_{\tilde{i},j,\hat{n}^m_{\tilde{i},j}} + \frac{\hat{i}-1}{\hat{n}^m_{\tilde{i},j}-1}\left(v^{m,\text{HDV}}_{\tilde{i},j,1} - v^{m,\text{HDV}}_{\tilde{i},j,\hat{n}^m_{\tilde{i},j}}\right) \quad \hat{i} = 2,3,\ldots,\hat{n}^m_{\tilde{i},j}-1. \tag{5.10b}$$

Having estimated HDV states (i.e., their positions and speeds), we treat HDVs as CAVs and formulate the slower-scale problem **A4**. After solving **A4**, we use the signal phase of the first slower interval as the signal phase for the next slower time step. Meanwhile, the trajectories of CAVs are used as the critical trajectories for the faster-scale vehicle control problem, which is discussed in the next section.

Faster-scale problem: vehicle control

For each CAV platoon $q^{m,\text{CAV}}_{\tilde{i},j}$, we can formulate and solve the faster-scale problem **A3** (as shown in the Appendix A) using the information generated by the slower-scale problem **A4**, i.e., critical trajectory points and the corresponding time points. The faster problem of flow $q^{m,\text{CAV}}_{\tilde{i},j}$ determines the detailed trajectory for each vehicle to satisfy the traffic rules. However, different from the full CAV penetration scenario, the trajectories generated by **A3** might not be feasible under mixed traffic flow. This is because we only consider CAV platoons when constructing **A3**, the leading CAV of a CAV platoon might be too close to the HDV in front of it, violating the safety constraint. One way to remedy this issue is to integrate the HDV dynamics in **A3**. However, there are three challenges of this approach. First, it will increase the complexity of **A3**. Second, the initial states of HDVs estimated by the linear interpolation are not very accurate, which will introduce more errors to the model, making

the idea of integrating HDV dynamics less meaningful. Third, the mathematical model of HDV may not well capture the real dynamics of HDVs since human driving behaviors are highly dynamics and have large variations. As a result, the solutions obtained from such a complex model might still be infeasible.

We use two techniques to overcome this issue mainly to ensure safety. First, the MPC scheme (i.e., Figure 4.2) can naturally adjust to the changing traffic situations. We solve the slower-scale problem **A4** at time step k_s by formulating T^s slower steps, and solve the faster-scale problem **A3** at the same time point by formulating $h\Delta T_s$ time duration with a smaller step length ΔT_f, i.e., a faster step k_f. Then, we apply the first $\Delta T_s/\Delta T_f$ solutions of the faster-scale problem to the first slower time step, i.e., $[k_s\Delta T_s, k_s\Delta T_s + \Delta T_s]$. Finally, we move to the next slower time step k_s+1 and do the same process again. Changes during the slower time step $[k_s\Delta T_s, k_s\Delta T_s + \Delta T_s]$ can be captured by the newly formulated problem at next slower time step k_s+1. Second, we enhance the safety by adding a safety check process before applying the solutions generated from **A3** to the CAVs. We check the commands from the first to the last vehicle in a CAV platoon if there is a HDV in front of the leading CAV. If the leading CAV is too close to the front HDV, i.e., it violates the car-following safety constraint, we use the intelligent driver model (IDM, Kesting et al. [2010]) to calculate an acceleration and apply it to the leading CAV. Then, we check other vehicles in the CAV platoon. If a CAV violates the car-following safety constraint, we calculate an alternative acceleration based on states of the front CAV using the IDM, and apply it to the current CAV. The acceleration of the IDM is calculated by

$$a_{i,j}^m = a_{\max}\left(1 - \left(\frac{v_{i,j}^m}{v^*}\right)^{\delta} - \left(\frac{d^*}{s_{i,j}^m - s_{i-1,j}^m}\right)^2\right) \tag{5.11a}$$

$$d^* = d_{\min} + v_{i,j}^m h_{\text{IDM}} + \frac{v_{i,j}^m(v_{i-1,j}^m - v_{i,j}^m)}{2\sqrt{a_{\max}a_{\text{comf}}}}, \tag{5.11b}$$

where $a_{\max}$ is the physical maximum acceleration, a_{comf} is the comfortable deceleration, v^* is the desired speed (which is usually the speed limit), δ is a constant acceleration exponent which is usually set to 4, $d_{\min}$ is the minimum car-following distance, and h_{IDM} is the safe

time headway.

5.2.3 The integrated framework

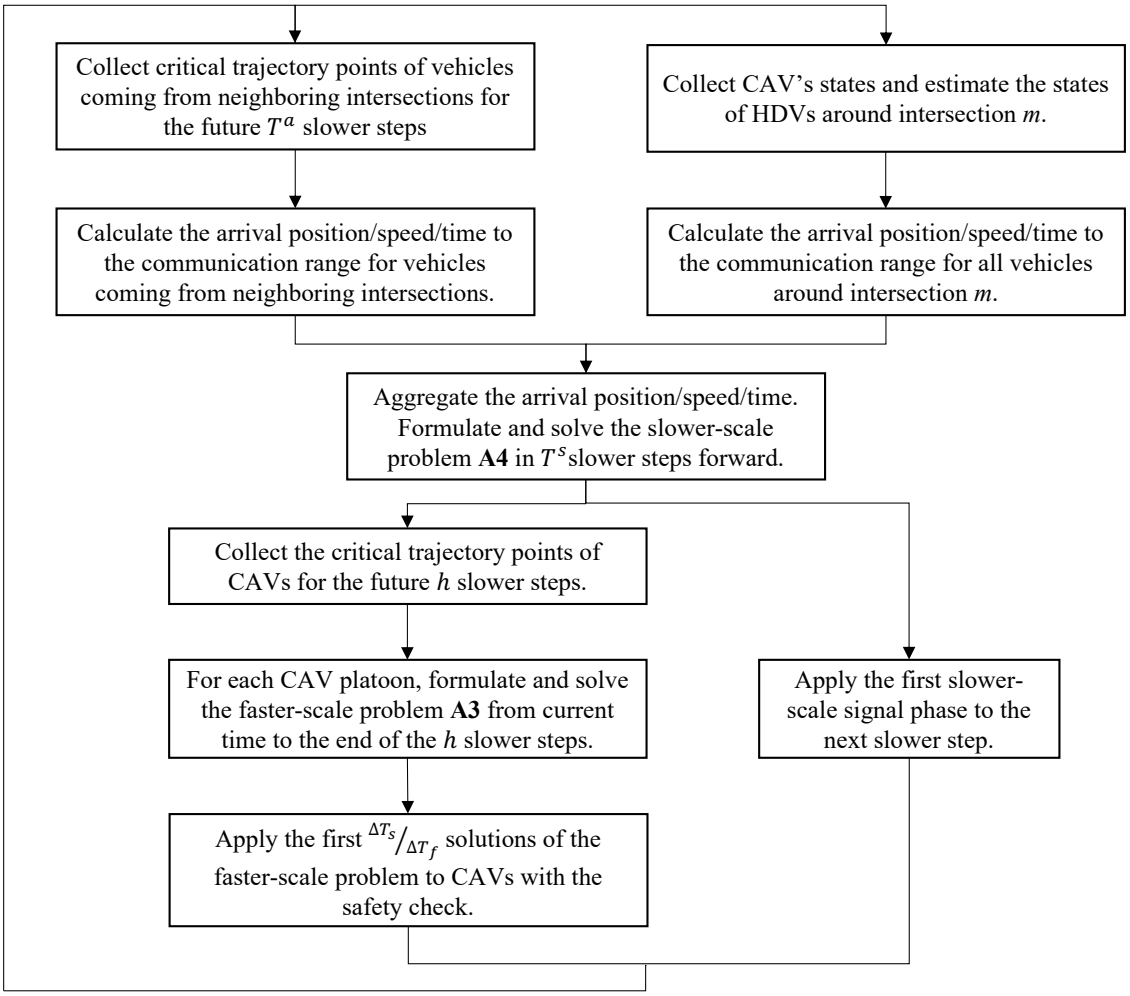

Figure 5.4: SVCC for mixed traffic flow and multiple intersections (any intersection m)

Combining the methods proposed in the last two subsections, the overall framework for network UTC with mixed traffic flow is shown in Figure 5.4. At a specific slower time step k_s, for any intersection m, we firstly collect the critical trajectory points of the vehicles coming from the neighboring intersections that are directly linked to m, using the solutions generated

at $k_s - 1$ (in slower time step). Based on the critical points, we can calculate the arrival position/speed/time to the intersection's communication range of all the neighboring vehicles. Meanwhile, we collect the CAVs' information and estimate the HDVs' states for the vehicles around the intersection, and estimate the corresponding arrival position/speed/times. Then, we aggregate the arrival time/position/speed for all vehicles that will arrive the intersection's communication range within the prediction horizon as $\left(k_{i,j}^{m,s}, v_{i,j}^{m,s}, s_{i,j}^{m,s}\right)$, formulate and solve the slower-scale problem **A4**. We apply the solution of **A4**, i.e., the optimal signal timing, for the next slower time step. Meanwhile, we collect the first h critical trajectory points of CAVs generated by solving **A4**. Then, at each faster-scale time step, for each CAV platoon, we formulate and solve the faster-scale problem **A3** from current faster-scale time to the end of the $h * \Delta T_s/\Delta T_f$ time steps. We apply the first $\Delta T_s/\Delta T_f$ solutions (i.e., accelerations of all CAVs) generated by solving each **A3** to every CAV, under the safety check principle. When we have finished one slower time step by moving with faster-scale time interval, we move to the next slower time step $k_s + 1$.

5.3 Numerical experiments

In this section, we test the proposed SVCC algorithm under mixed traffic flow for a single intersection, and expand the scenario to a corridor, a synthetic 4 by 4 network, and a real-world 4 by 6 network.

5.3.1 Experiment settings

Common settings

Same as in Chapter 4, we use the third degree polynomial function (Equation (4.37)) of speed and acceleration as the fuel consumption model (Vellamattathil Baby et al. [2020]). The slower-scale and faster-scale time steps are also the same as in Chapter 4, i.e., $5s$ and $0.5s$, respectively. Since we have tested the prediction horizon in Chapter 4, we use the same tuned prediction horizon $T^s = 4$ to balance the computation burden and performance gain.

Similarly, we set $h = 3$.

To compare the performances, we build and test two other algorithms: fixed-time traffic signal control and actuated traffic signal control. Note that it is hard to design fixed-time signal plans for traffic networks, we only use the actuated signal control for the comparison when it comes to the synthetic 4 by 4 network and the real-world 4 by 6 network. Nevertheless, actuated signal control is the most commonly used signal control method (besides fixed time control). For both the fixed-time and actuated signal control methods, vehicles are controlled by the default car-following model in the simulation software, i.e., SUMO. Hereafter in this section, we use Multiscale, Actuated, Fixed-time to denote the proposed multi-scale SVCC, the actuated signal control, and the fixed time signal control, respectively.

As in Chapter 4, we integrate Python, General Algebraic Modeling System (GAMS), and SUMO as the simulation platform. For the model-based algorithm, GAMS is used to solve the slower-scale problem **A4** and the faster-scale problem **A3**. SUMO is used to simulate different traffic scenarios and generate data. Python is used as the integration platform to communicate between SUMO and GAMS.

Settings for the single intersection, corridor, and synthetic 4 by 4 network scenarios

We use the same settings in Chapter 4, i.e., Figure 4.3 and 4.4, as the testing scenario for the single intersection. For the single intersection scenario, we test two volume settings: symmetric and asymmetric volumes. For the symmetric setting, the straight volumes of each of the four legs follows the same volume demand, which increases linearly from 400 veh/hour to 600 veh/hour (the testing period is set as 15 minutes) to mimic the normal hours and peak hours. For the asymmetric setting, the eastbound and northbound volumes increase from 550 veh/hour to 800 veh/hour while the westbound and southbound volumes remain the same as the symmetric setting. For both symmetric and asymmetric volume settings and for each leg, the left and right turn volumes are set as one-third of the straight volumes. By testing the symmetric and asymmetric volumes, the ability of dealing with different traffic scenarios can be revealed.

Based on the single intersection, we build a 5-intersection corridor and a 4 by 4 network, which is shown in Figure 5.5. For the volumes, we also test symmetric and asymmetric settings, but at a network level. We firstly categorize the origin-destination (OD) to two types: main and minor. The main ODs are those pairs directly connected by a straight road. For example, the main road of the 5-intersection corridor and the four south-north straight roads and the four west-east straight roads in the 4 by 4 network. The minor ODs includes those pairs that are not directly connected by straight roads. For example, for the left bottom corner origin in the 4 by 4 network (i.e., the south incoming road of the left bottom intersection), apart from the main destination (i.e., the north outgoing road of the left upper intersection), there are destinations to the outgoing roads around the boundaries of this network, which are referred to as the minor destinations. For symmetric volumes, we set the volumes for all main ODs as 180 veh/hour to 240 veh/hour (also increasing linearly during the simulation) and the volumes for all minor ODs as the one-sixth of the main ODs. For asymmetric volumes, we use the same volume generation strategy but volumes coming from west and south directions are the same as the symmetric case, but volumes coming from east and north directions are half as the symmetric case.

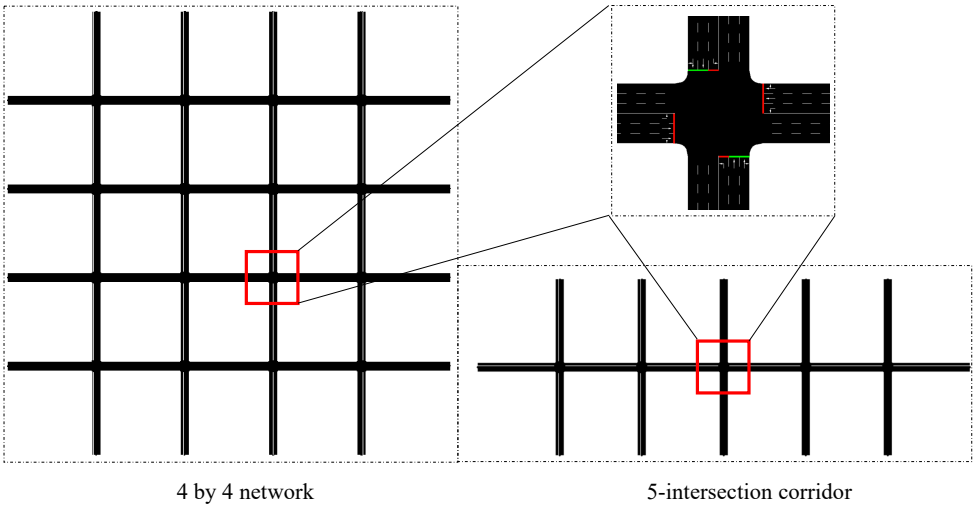

Figure 5.5: Corridor and network scenarios

Settings for the real-world 4 by 6 network scenario

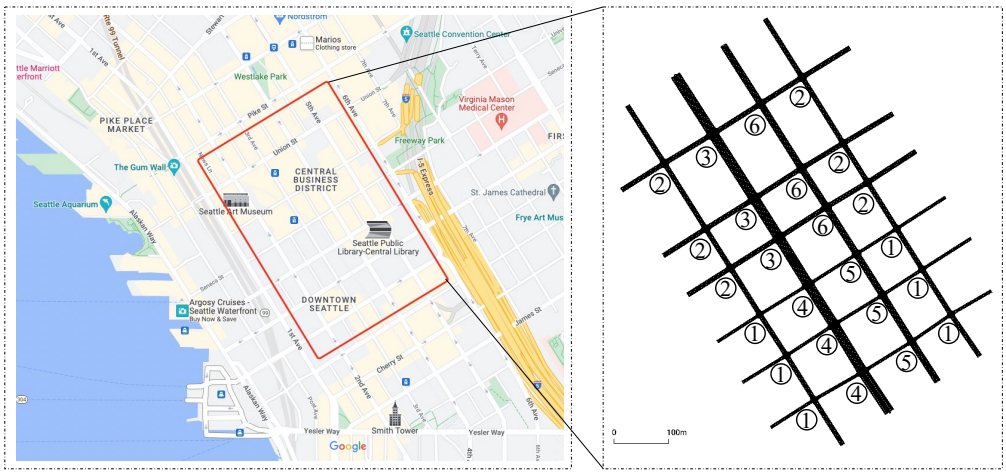

Figure 5.6: A 4 by 6 Seattle downtown network

We build a real-world 4 by 6 network based on the downtown area of Seattle, United States, which is shown in Figure 5.6. There are different road types (e.g., one-way and two-way roads) in this network, leading to various intersection geometries. The numbers in circles in the right-side SUMO networks represent the types of the corresponding intersections. As shown in Figure 5.7, we categorize the intersections in this network into six types:

- Type 1: two one-way roads with two and three incoming lanes, respectively.
- Type 2: two one-way roads both with three incoming lanes.
- Type 3: one one-way road with three incoming lanes and one two way road with three incoming lanes.
- Type 4: one one-way road with two incoming lanes and one two way road with three incoming lanes.

- Type 5: two one-way roads with two and four incoming lanes, respectively.
- Type 6: two one-way roads with three and four incoming lanes, respectively.

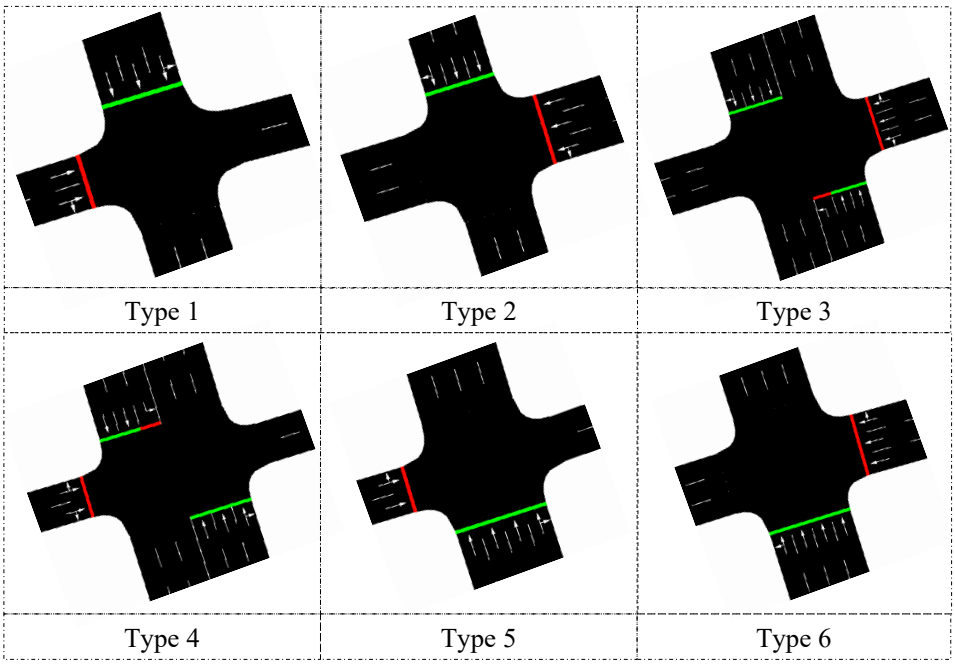

Figure 5.7: Six intersection types in Seattle downtown network

We extract the real-world OD volume data from the downtown Seattle SUMO simulation network, which has been built and calibrated by the authors' research group. The number of OD pairs is large and we omit the detailed OD volumes here for the sake of brevity. To have an intuitive understanding, the maximum volume is 332 veh/hour from the right upper north incoming road to the right bottom south outgoing road.

5.3.2 Experiment results

In this section, we test and analyze the experimental results of the proposed SVCC method. We firstly discuss the single-intersection scenario to provide a basic understanding of the

method's performance. Then, we show the results for the 5-intersection corridor, the synthetic 4 by 4 network, and the real-world 4 by 6 Seattle downtown network to further illustrate the performance.

Single-intersection

Figure 5.8 shows the average waiting time, time loss, and queue length under different CAV penetrations and different control types: Multiscale, Actuated, and Fixed-time. The waiting time (in seconds) is defined as the number of seconds a vehicle has a speed of less than $0.1m/s$. The time loss is defined as the time lost due to travelling at speed below the maximum speed. The queue length is calculated using the end of the last standing vehicle. For the Actuated and Fixed-time methods, the penetration of CAVs does not affect the performance because these two methods do not differentiate CAVs and HDVs.

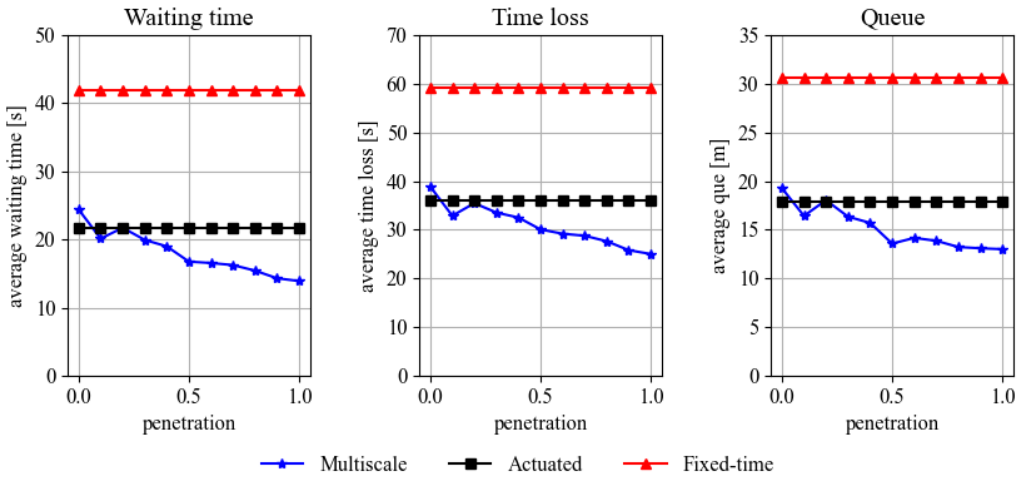

Figure 5.8: Average waiting time, time loss, and queue length under different CAV penetrations

It can be shown from Figure 5.8 that the Actuated method outperforms the Fixed-time method, which is expected since the Actuated method is more adaptive to real-time traffic

states with the help of loop detectors. The average waiting time, time loss, and queue length of the Actuated method is 47.62%, 38.98% and 43.55% lower (i.e., better) than the three indexes of the Fixed-time method, respectively. The performance of the Multiscale method increases as the CAV penetration increases. The higher the CAV penetration is, the more information the Multiscale can use to calculate the best operation strategies, usually leading to improved performance. The Multiscale method outperforms the Fixed-time under all CAV penetrations, and the performance gain of waiting time, time loss, and queue length ranges from 40.48%, 43.48%, 38.71% (at 0 penetration) to 66.67%, 63.77%, 58.06% (at 100% penetration), respectively. Compared with the Actuated signal control, the Multiscale method performs a little worse at 0 CAV penetration, but better as the CAV penetration increases. At 100% CAV penetration, the Multiscale method performs 36.36%, 30.56%, 22.86% better than the Actuated signal control method regarding the waiting time, time loss, and queue length, respectively.

Figure 5.9 shows the average fuel consumption of CAVs, HDVs, and the mixed traffic flow under different CAV penetrations and different control types. Note that the average fuel consumption of the Actuated and Fixed-time methods does not change for the three sub-figures because CAVs and HDVs are not differentiated in these two methods. Similar trends as the above analysis of waiting time, time loss, and queue length can be found from Figure 5.9. Actuated signal control outperforms the Fixed-time signal control since the actuated signal control usually extends a green phase when newly incoming vehicles are detected so that vehicles can avoid decelerating and stopping, which can help reduce the fuel consumption of idling and re-accelerating. Compared with the Fixed-time signal control, the average fuel consumption of the mixed traffic flow (i.e., both CAVs and HDVs) of the Actuated signal control is 27.78% less. The fuel-saving performance of the Multiscale method increases as the increase of CAV penetration. Compared with the Fixed-time method, the Multiscale method can save fuel from 22.22% (at 0 CAV penetration) to 38.89% (at 100% CAV penetration). Similarly, compared with the Actuated signal control, the Multiscale method performs slightly worse when the CAV penetration is less than 20%, but performs

much better when the CAV penetration increases. The maximum fuel-saving gain of the Multiscale method compared with the Actuated method is 18.52% when CAV reaches full penetration.

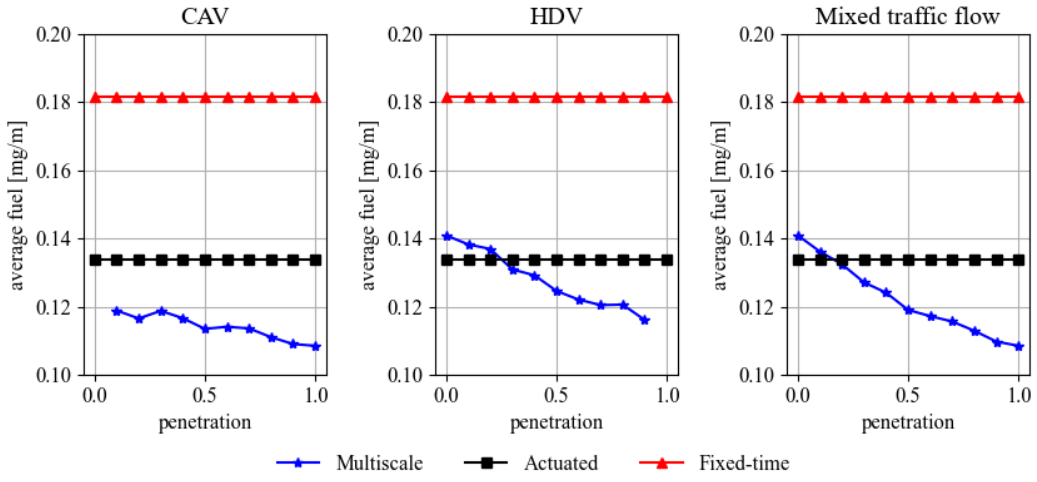

Figure 5.9: Average fuel consumption of CAV, HDV, and the mixed traffic flow under different CAV penetrations

5-intersection corridor, synthetic 4 by 4 network, and real-world 4 by 6 network

Table 5.2 and Table 5.3 show the performance (i.e., average waiting time, time loss, queue length, and fuel consumption of the mixed traffic flow) of different control methods under three different scenarios: a 5-intersection corridor, a synthetic 4 by 4 network, and a real-world 4 by 6 Seattle downtown network. For the first two synthetic networks, since we have the freedom to vary the volume demands, we test two different volumes, i.e., symmetric and asymmetric, for each of the scenarios. In Table 5.2 and Table 5.3, "sym.", "asym.", and "imp." represent symmetric volume, asymmetric volumes, and improvement, respectively. In addition, imp. I and imp. II represent the improvements of the Multiscale method compared with the Actuated and Fixed-time methods, respectively. There are two values for each performance index for the Multiscale method, where the first one is the value under 0%

CAV penetration and the second one is the value under 100% CAV penetration. Note that it is hard to design fixed time signal plan for a network, and considering that the Actuated method usually outperforms the Fixed-time method, we only compare the performances of the Multiscale and Actuated methods for the two network scenarios.

There are several observations from the two tables. First, it is clearly shown that both Actuated and Multiscale perform much better the Fixed-time method. Compared with the Fixed-time method, the maximum performance gains of the Multiscale method regarding average waiting time, time loss, queue length, and fuel consumption can reach $79.34\%, 68.90\%, 66.27\%$ and 53.54%, respectively. Second, Multiscale also outperforms Actuated under all scenarios. Although the Multiscale method performs slightly worse than the Actuated method under low CAV penetration in the single-intersection scenario, when the size of the traffic scenario increases to corridors or networks, the Multiscale method performs better than the Actuated method, even under 0% CAV penetration. Compared with the Actuated method, the minimum performance gains of the Multiscale method (under 0% CAV penetration) regarding average waiting time, time loss, queue length, and fuel consumption is $3.84\%, 4.49\%, 4.95\%$ and 0.00%, respectively, and the maximum gains can reach $55.57\%, 48.33\%, 31.41\%$ and 28.42%, respectively.

Table 5.2: Average waiting time and time loss under different scenarios

Scenario	Volume	Method	Performance indexes	
			avg. waiting time (s)	avg. time loss (s)
5-intersection corridor	sym.	Actuated	29.66	51.18
		Fixed-time	77.21	104.65
		Multiscale value	28.52 - 17.68	48.88 - 34.37
		Multiscale imp. I	3.84% - 40.39%	4.49% - 32.84%
		Multiscale imp. II	63.06% - 77.10%	53.29% - 67.16%
	asym.	Actuated	27.46	47.51

		Fixed-time		68.44	89.7
		Multiscale	value	23.67 - 14.14	41.03 - 27.90
			imp. I	13.80% - 48.51%	13.64% - 41.28%
			imp. II	65.41% - 79.34%	54.26% - 68.90 %
Synthetic 4 by 4 network	sym.	Actuated		73.29	118.46
		Multiscale	value	59.06 - 39.39	97.54 - 65.59
			imp.	19.42% - 46.25%	17.66% - 44.63%
	asym.	Actuated		67.52	109.15
		Multiscale	value	51.00 - 30.00	84.20 - 56.40
			imp.	24.47% - 55.57%	22.86% - 48.33%
Real-world 4 by 6 Seattle network	real	Actuated		19.06	44.78
		Multiscale	value	17.79 - 10.51	40.12 - 33.23
			imp.	6.66% - 46.75%	10.41% - 25.79%

Table 5.3: Average queue length and fuel consumption under different scenarios

Scenario	Volume	Method	Performance indexes	
			avg. queue length (m)	avg. fuel (mg/s)
5-intersection corridor	sym.	Actuated	13.34	0.135
		Fixed-time	28.31	0.226
		Multiscale value	12.68 - 9.55	0.135 - 0.105
		imp. I	4.95% - 28.41%	0.00% - 22.22%
		imp. II	55.21% - 66.27%	40.27% - 53.54%
	asym.	Actuated	11.3	0.130
		Fixed-time	19.95	0.208
		Multiscale value	9.94 - 8.01	0.125 - 0.097

			imp. I	12.04% - 29.12%	3.85% - 25.38%
			imp. II	50.18% - 59.85%	39.90% - 53.37%
Synthetic 4 by 4 network	sym.	Actuated		15.27	0.164
		Multiscale	value	13.03 - 10.13	0.151 - 0.114
			imp.	14.67% - 33.66%	7.93% - 30.49%
	asym.	Actuated		12.6	0.156
		Multiscale	value	10.33 - 8.47	0.141 - 0.107
			imp.	18.02% - 32.78%	9.62% - 31.41%
Real-world 4 by 6 Seattle network	real	Actuated		9.04	0.144
		Multiscale	value	8.80 - 7.00	0.140 - 0.110
			imp.	2.65% - 22.57%	2.78% - 23.61%

To have a more intuitive understanding of different control methods, we show the spatial diagrams of all west-to-east (WE) straight vehicles in the 5-intersection scenario, with the Multiscale control (Figure 5.10) and the Actuated signal control (Figure 5.11). WE straight vehicles are those entering from the west most road, travelling through the whole corridor, and exiting from the right most road, see Figure 5.5 for the corridor. The two figures are made using the trajectories under the symmetric volume setting, i.e., the volumes of west-to-east straight vehicles and east-to-west straight vehicles are the same. In the two figures, the horizontal and vertical axes represent time and distance, respectively. The red dotted lines indicate that the signal phase for the WE straight vehicles are red (thus the blank intervals indicate green phase), and the blue lines represent the trajectories of each vehicle. Note that for the Multiscale method, any phase in the 8 candidate phases (Figure 4.4) can be selected. There are some green phases without any trajectory, e.g., around $80s$ at the fourth intersection (i.e., $800m$). This is because there are two phases that the WE vehicles can pass the intersection, i.e., $\mathrm{E_S W_S}$ and $\mathrm{E_S E_L}$, the blank green phases are used for left-turning vehicles. For the Actuated method, we set the cycle as $\mathrm{N_S S_S} \rightarrow \mathrm{N_L N_L} \rightarrow \mathrm{E_S W_S} \rightarrow \mathrm{E_L W_L}$. The minimum green time is $10s$ and the maximum green time is $45s$.

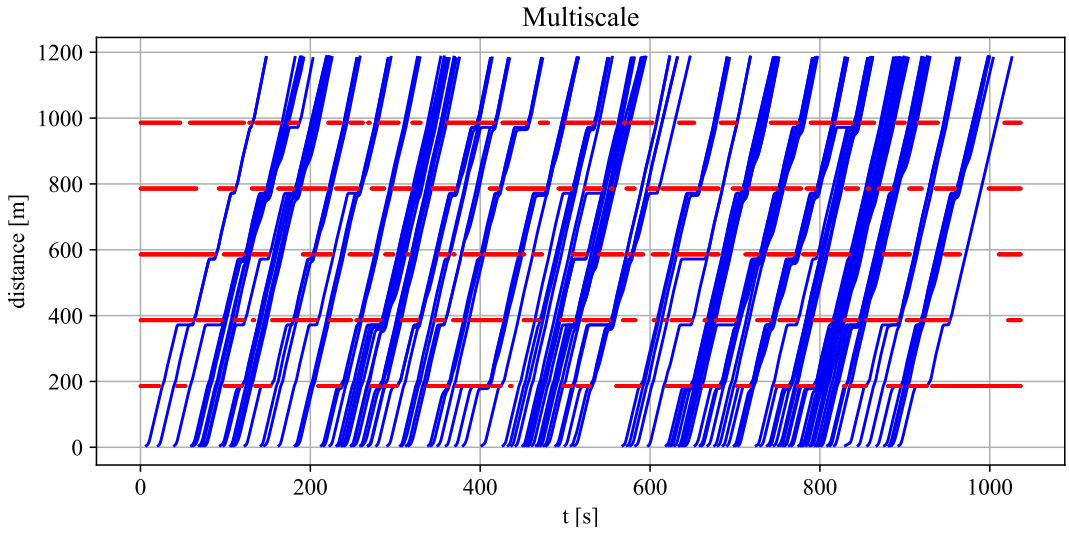

Figure 5.10: Spatial diagram of all WE straight vehicles in the 5-intersection corridor scenario with the Multiscale control method

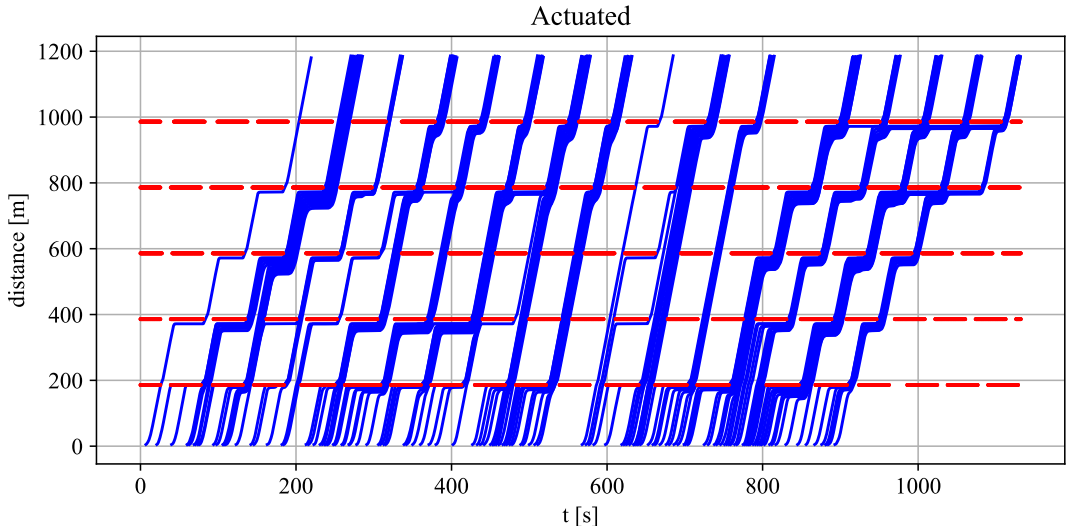

Figure 5.11: Spatial diagram of all WE straight vehicles in the 5-intersection corridor scenario with the Actuated control method

From Figure 5.10 and 5.11, it can be shown that the Multiscale method can clear the corridor sooner than the Actuated method (which can be shown from the ending time of the two figures). Considering that the volume settings are the same, the Multiscale method shows better control performance. One of the main advantages of the Multiscale method is it generates more "green waves" for vehicles than the Actuated method. With the green waves, vehicles can pass the corridor with fewer stops, resulting in lower delays and lower fuel consumption. On the contrast, due to the lack of coordination, the Actuated signal control generate less green waves but relatively more congestion, especially when the volume is higher (e.g., the right part of Figure 5.11).

In summary, under the scenarios of a 5-intersection corridor, a synthetic 4 by 4 network, and a real-world 4 by 6 Seattle downtown network, the proposed model-based Multiscale method performs best when compared with the Actuated and the Fixed-time methods. In addition, the performance of the Multiscale method increases as the CAV penetration increases.

5.4 Summary

In this section, based on the multi-scale SVCC algorithm for single intersection with full CAV penetration (i.e., Chapter 4), we developed a multi-scale SVCC algorithm for multiple intersections with mixed traffic flow. We developed a distributed control method for each individual intersection which can achieve coordination among different intersections through information sharing technique. We also extended the MPC scheme for single intersection to traffic networks and analyzed the control stability of the MPC using the state consistency between the slower-scale and faster-scale problems. In addition, we estimated HDV's information based on CAV's states and designed a safety check mechanism to guarantee safety. Finally, we combined the two techniques as an integrated framework that can deal both network-wide traffic control and the mixed traffic flow condition. Numerical experiments on various traffic scenarios showed that the proposed algorithm could largely improve the traffic performance compared with traditional actuated signal control and fixed-time control

methods.

Chapter 6

LEARNING-ENHANCED SVCC

As discussed in Section 2 and Section 3.2, there are essentially two types of methods in terms of the learning-based UTC: RL and IL. We choose IL as the foundational learning method in this study due to three reasons. First, RL usually suffer from exploration and convergence problems, making it hard to design the objective function and tune the parameters. Second, there is usually a large gap between the training performance of the RL methods and the real implementation performance, since RL is often trained in simulation environments due to the needs of exploration and extra work necessary for the transfer to real implementations. Third, one of the biggest weaknesses of IL is the lack of expert data because collecting such data is usually time-consuming and costly, especially for some manually calculated expert data. However, we have built the model-based methods, which can serve as the "expert" to efficiently generate learning data for IL. In this paper, we will explore the idea of using model-based methods to generate training data for IL, which we call it learning-enhanced SVCC model. As we show in more detail later in the numerical section, such an IL based on model-generated data has some interesting features such as certain level of transferability.

IL techniques aim to mimic expert's behavior by learning a mapping between observations and actions that generated by the experts (i.e., the demonstrations) for certain tasks. IL is not a new idea but is drawing increasing attention recently as the rise of AI and the developments in computing and sensing technologies. Based on the learning methods, IL can be roughly categorized into three types: behavioural cloning (BC), inverse reinforcement learning (IRL), and direct policy learning (DPL) (Le Mero et al. [2022]). BC is a supervised IL approach which creates a direct mapping using the observation-action pairs in the expert's

dataset (Torabi et al. [2018]). However, the policy trained on expert's data may introduce errors if there are new observations when it is applied to new tasks, and most importantly, such errors may be compounded, leading to the cascading error problem (Bagnell [2015]). DPL and IRL were developed to remedy the limitations of BC. IRL aims to firstly learn a reward function through the demonstrations, and use forward RL techniques to learn a good policy based on the learned reward function. Such reward functions can be feature-based (Ng et al. [2000], Abbeel and Ng [2004] or model-free Ho and Ermon [2016], Fu et al. [2017], Kuefler et al. [2017]). However, feature-based methods may suffer from reward function ambiguity issues (Ng et al. [2000]), and all those methods need RL in the loop to learn the optimal policies after learning the reward function. DPL chooses another approach to overcome the limitations of BC. DPL enriches the demonstration dataset by requesting the experts to evaluate the policy during the application. When new states are observed, or the actions generated by the initial policy is not good, the experts will help generate good actions and the new state-action pairs will be added to the dataset. The policy will be re-trained on the new dataset (Ross et al. [2011], Pan et al. [2020]). The major drawback of DPL is that having an expert to evaluated current policies and generate good actions based on current observations is expensive. However, as mentioned previously, we have built the model-based methods for the multi-scale SVCC, which can serve as the "expert" in this case. This can help train the IL model, which may then be applied to UTC on other (and similar) networks. In this paper, we use one of the DPL methods, the data aggregation (DAGGER, Ross et al. [2011]) algorithm, as the learning framework.

6.1 Methodology

Considering that the policy itself is trained in a supervised learning manner, there are various methods to represent the policy. For example, linear regression (Montgomery et al. [2021]), logistic regression (Menard [2002]), support vector machine (Noble [2006], decision tree Song and Ying [2015]), and neural networks (Hecht-Nielsen [1992]). In this paper, we use the deep neural networks to build the policy due to their strong representational and generalization

capacities. The policy structure is shown in Figure 6.1. For each incoming lane of intersection m, we firstly discretize it to N cells (5-meter is used as the cell length in this paper), and build three vectors using the discretized lane representation, i.e., the position, speed, and waiting time vectors. Including waiting times into the states can avoid the situation that a small traffic flow (e.g., on the side street) might wait very long time when the volume of the conflicting movements is high. For each vehicle on a specific lane, whether it is a CAV (which we know all the information) or an HDV (whose information is estimated), we calculate the cell index based on its position, and set the corresponding position vector cell as 1. The corresponding speed and waiting time cells are filled with normalized values: the speed is normalized to $[0, 1]$ linearly based on the speed limit (i.e., $\tilde{v}_{i,j}^m = v_{i,j}^m / v_{\max}$), the waiting time is also normalized to $[0, 1]$ linearly using a predefined waiting time bound (i.e., $\tilde{t}_{i,j}^m = wt_{i,j}^m / wt_{\max}$). Then, we collect the vector states for all incoming lanes and aggregate them into three matrices representing position, speed, and waiting time, respectively. The three matrices can be understood as a 3-channel image, and we choose the ResNet (He et al. [2016]) as the policy network. ResNet is a deep CNN (LeCun et al. [1998]) that uses residual learning to overcome the degradation problem in deep neural networks and thus achieve efficient learning. The output of the ResNet is the signal timing plans for the next two slower time steps. The first will be used to control the traffic signals, while the second is used to estimate the vehicle positions by assuming that the signal timing plans for the next two steps follow the results generated by ResNet.

IL is used to control the traffic signals given traffic states (i.e., to solve the slower-scale problem). To achieve efficient SVCC, we further decouple the signal and vehicle controls: we do not require the state consistency of the vehicle trajectories, i.e., the slower-scale signal control algorithm will not generate the critical trajectories for the faster-scale vehicle control problem. Although this dampens the ability to analyze and guarantee stability, such a further simplification is necessary for a learning-enhanced method because it is very hard for the deep neural networks to generate trajectories for every vehicle. For the vehicle control problem, we design a rule-based algorithm with given signal information. In fact, the fuel-

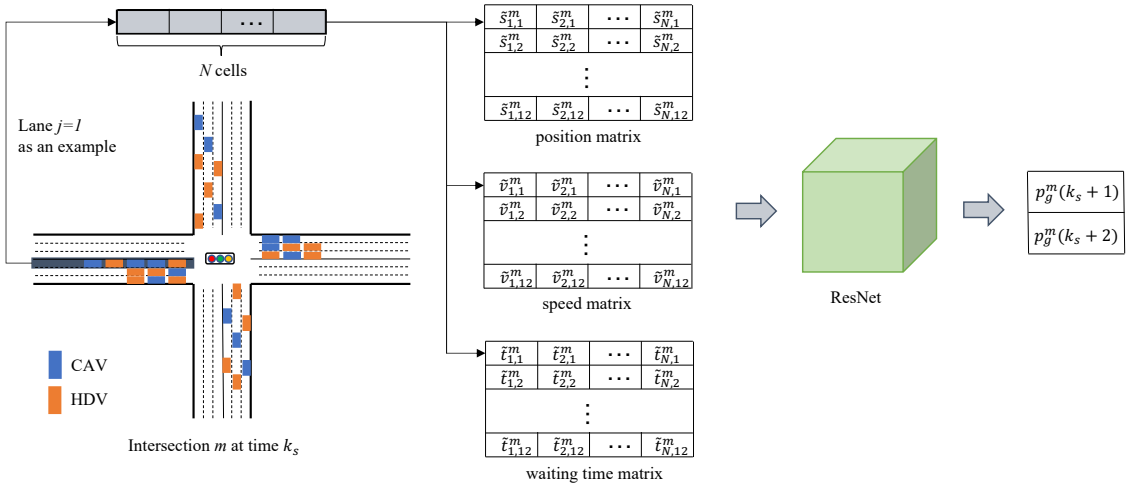

Figure 6.1: The ResNet based learning framework

efficient and safe longitudinal driving strategies for road vehicles have pretty clear patterns. In urban area, a driving process can be segmented into several sub-processes, each with similar patterns. Usually, each sub-process includes acceleration (e.g., starting from the stop line), cruise (e.g., after reaching speed limit), and deceleration (e.g., stopping before a stop line with a red traffic signal ahead). Such a three-stage driving strategy has been recognized in literature (Lin et al. [2020]). Among these three stages, the deceleration stage is usually the main stage that fuel can be saved. Gliding to decelerate is the most fuel-efficient deceleration strategy while most drivers do not follow such a strategy. Therefore, we design a rule-based driving algorithm for all the CAVs, which is shown in Figure 6.2.

For a specific CAV on a specific lane at a specific intersection, we firstly check whether the current lane has the right of way or not. Then, we check whether the current CAV is the first CAV on this lane or not. There are four resulted conditions:

- If the current lane has the right of way and the current CAV is the first CAV on the current lane, we further check if the speed limit $v_{\max}$ has been reached or not. If the CAV is driving with the speed limit, we set the reference acceleration $a_{\text{ref}} = 0$, i.e., the

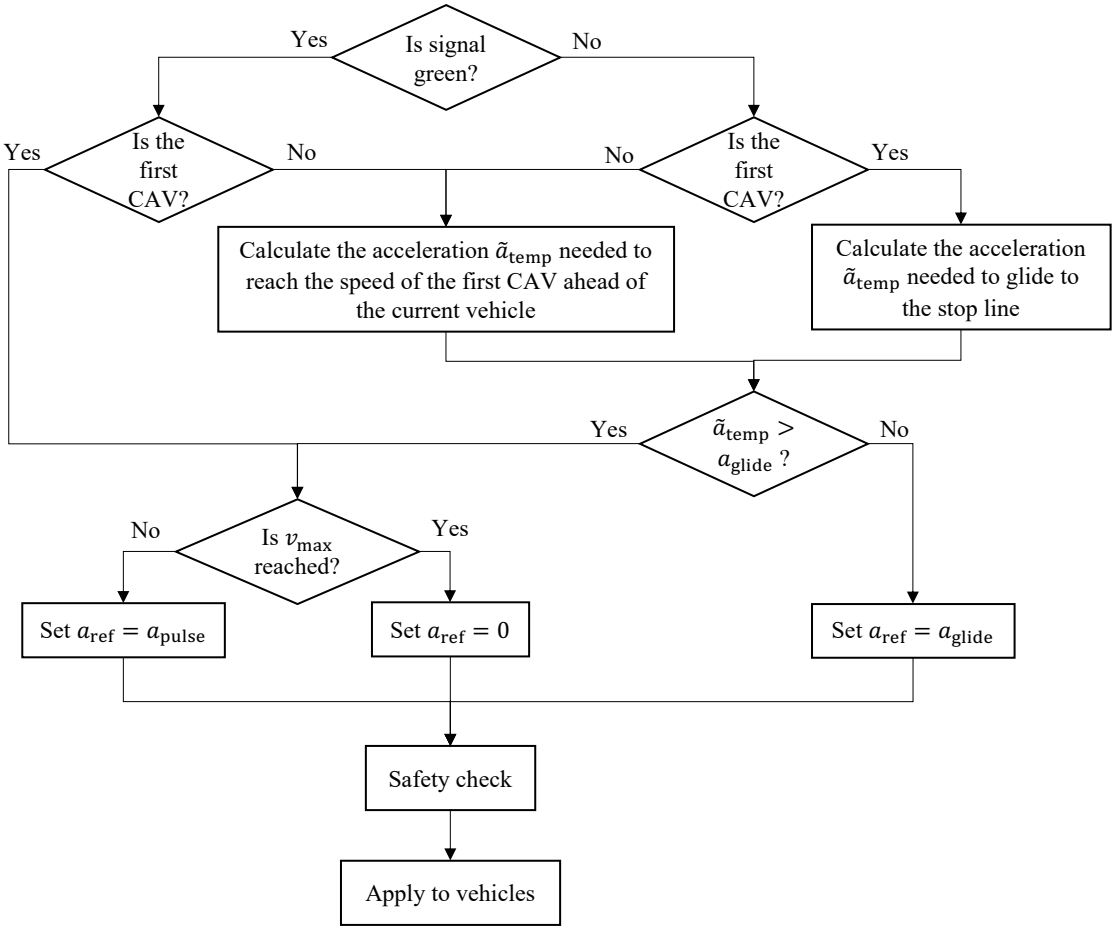

Figure 6.2: Flowchart of the rule-based vehicle control

CAV is controlled to maintain the speed limit to pass the intersection. If the speed limit has not been reached, we set the reference acceleration $a_{\text{ref}} = a_{\text{pulse}}$, where a_{pulse} is a pre-defined economical acceleration value which will be used for all acceleration process.

- If the current lane has the right of way and the current CAV is not the first CAV, i.e., there is at least one CAV ahead. We will calculate the acceleration a_{temp} needed to reach the speed of the first front CAV for the current CAV. Then, if $a_{\text{temp}} > a_{\text{glide}}$ where a_{glide} is a pre-defined economical deceleration value which will be used for all

deceleration process, we further check if the current CAV is driving on the speed limit or not. As the first scenario, if the speed limit has been reached, we set the reference acceleration as $a_{\text{ref}} = 0$. Otherwise, we set $a_{\text{ref}} = a_{\text{pulse}}$, i.e., there is still space for the current CAV to gain more speed. If $a_{\text{temp}} \leq a_{\text{glide}}$, we set $a_{\text{ref}} = a_{\text{glide}}$, i.e., the current CAV should begin to decelerate to match the speed of the front CAV.

- If the current lane does not have the right of way and the current CAV is the first CAV, i.e., vehicles on this lane should stop before the stop line, we firstly calculate the deceleration value a_{temp} needed to glide to the stop line (with final speed 0). Then, as the above scenario, we further check whether $a_{\text{temp}} > a_{\text{glide}}$ or not. If $a_{\text{temp}} > a_{\text{glide}}$ (e.g., the first CAV is far from the stop line), we believe that there is still some space for the CAV to gain speed thus we follow the rule as described above to handle the speed limit. Otherwise, we set $a_{\text{ref}} = a_{\text{glide}}$.

- If the current lane does not have the right of way and the current CAV is not the first CAV, the scenario is the same as the second one. The current CAV should refer to the front CAV to decide whether it should accelerate or decelerate.

After generating the reference accelerations, we use the same safety check method introduced in Section 5.2.2 to avoid unsafe situations due to the existence of HDVs. Finally, we apply the adjusted acceleration to the current CAV.

Integrating the IL-based traffic signal control and the rule-based vehicle control, the pseudo-code of the learning-enhanced SVCC method is shown in Algorithm 1.

Algorithm 1 Learning-enhanced SVCC

Initialize demonstration dataset $\mathcal{D} \leftarrow \emptyset$

for i = 0:10 **do**

Set CAV penetration as i/10.

Run the model-based multi-scale SVCC algorithm and collect dataset $\mathcal{D}_i$.

Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.

end for

Train the initial policy π_1 on $\mathcal{D}$.

for j = 1:N **do**

for i = 0:10 **do**

Set CAV penetration as i/10.

Use policy π_j to control traffic signals and use the rule-based method to control CAVs to run the simulation. Meanwhile, use the model-based method as a shadow to calculate the optimal actions for states encountered, and collect dataset $\mathcal{D}_i$.

Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.

end for

Train policy π_{j+1} on $\mathcal{D}$.

end for

return Best policy on validation.

6.2 Numerical experiments

The settings for the test scenarios are the same as in Chapter 5, where we tested the model-based SVCC algorithm under a corridor, a 4 by 4 synthetic network, and a 4 by 6 real-world Seattle downtown network. In this section, we show the experiments on the learning-enhanced SVCC algorithm for the synthetic 4 by 4 network and the real-world 4 by 6 network. Note that the learning-enhanced algorithms are directly trained and updated on the tested networks, i.e., we have to train different learning-enhanced algorithms given different traffic networks. Considering that the basic units (i.e., single intersections) of different traffic networks are usually the same, we also explore the transferability of the proposed learning-enhanced algorithm by training each intersection type first and applying them to traffic networks that consist of multiple types of intersections.

Specifically, the communication range is set as $100m$, and the cell length for discretization is set as $5m$, i.e., each lane can be represented by a 3 (position, speed, and waiting time) $\times 20$ matrix. The batch size when training the ResNet is set as 64 and the learning rate is set

as 0.0001. We compare the learning-enhanced method with the Actuated and the model-based Multiscale control methods. Hereafter in this paper, we use Multiscale, L-enhanced, Actuated, Fixed-time to denote the model-based multi-scale SVCC, the learning-enhanced SVCC, the actuated signal control, and the fixed time signal control, respectively.

In the first part of this section, we show the experimental results of Algorithm 1 on the synthetic 4 by 4 network and the real-world 4 by 6 Seattle downtown network, respectively. For this part, Algorithm 1 is directly applied to the two networks, i.e., we train and update the policies for each network separately. It might become very expensive when the size of the network increases. Considering that the basic units (i.e., single intersections) of different networks are usually the same and the proposed L-enhanced method is also intersection-based, in the second part, we explore a new learning strategy by training each intersection type first (in a small network) and applying them to traffic networks that contain these types of intersection. In this way, the learned policies can be easily applied to a new network and thus the proposed learning-enhanced method may present certain transferability property (note that this is not a "transfer learning" (Pan and Yang [2009], Weiss et al. [2016]) model).

6.2.1 Performance analysis of the learning-enhanced SVCC method on the synthetic 4 by 4 and real-world 4 by 6 networks.

We firstly show the training performance of the synthetic 4 by 4 network in Figure 6.3. We use the average waiting time as an example to show the performance (other indexes such as average time loss and queue length are in the same trend and omitted here for brevity). The horizontal axis in the two sub-figures is the number of round, i.e., j in Algorithm 1. The more round, the more data that we can use to train the policies. The performance of Actuated and Multiscale do not change along with the training round because there is no need to train them. The data points in Figure 6.3 are calculated from 110 testing episodes (10 episodes with different random seeds for each CAV penetration which ranges from 0% to 100% CAV penetration with a step of 10%) after each training round. Round 0 here represents the initial policies trained using the data generated purely by the model-based Multiscale method. For

other rounds, we use the learned policies to control the traffic signals and the rule-based method to control the vehicles. Meanwhile, we use the Multiscale method as a "shadow" method to generate training data. As shown in Figure 6.3, the L-enhanced method can learn a good policy after the first round iteration: the average waiting time decreases dramatically and the standard deviation also drops significantly. After that, the performance stabilize around a certain level with small fluctuations, indicating that the L-enhanced algorithm converges.

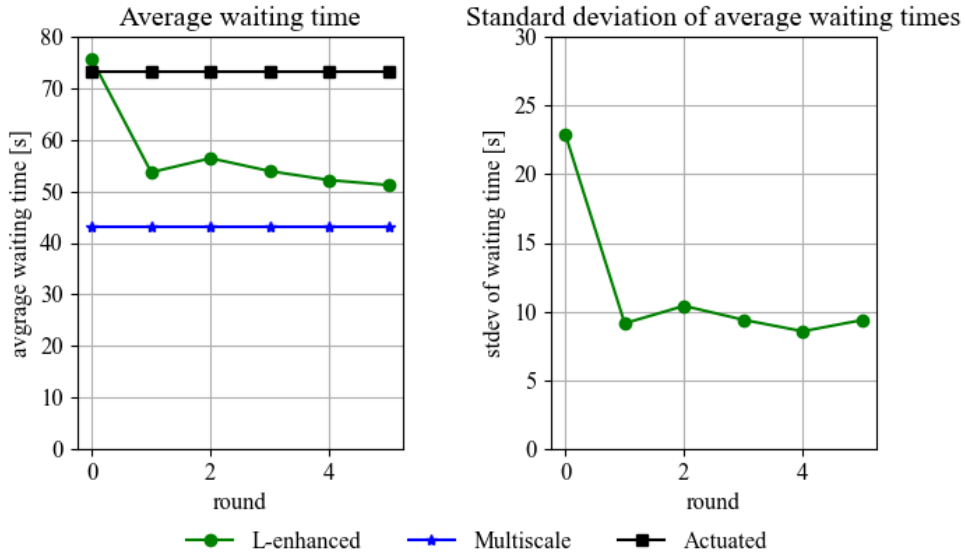

Figure 6.3: Training performance of the synthetic 4 by 4 network under symmetric volume

Then, we show the performance of the learned policy on the 4 by 4 network in Figure 6.4. The average waiting time, queue length, and fuel consumption of the mixed traffic flow along different CAV penetrations are shown. It shows that the L-enhanced method outperforms the Actuated method under all CAV penetrations and performs close to the Multiscale method. This is reasonable because the L-enhanced method tries to mimic the policies of the Multiscale method, i.e., the best performance that the L-enhanced method can achieve is the model-based Multiscale's performance. The main advantage of the L-enhanced

method to the Multiscale method is the computation efficiency. Based on our observation, it takes about four hours for a 15 minutes simulation for the model-based Multiscale method on the synthetic 4 by 4 network, while the time can be reduced to 5 minutes when we apply the L-enhanced method. In other words, the L-enhanced method achieves similar performance as the Multiscale method but with much higher computation efficiency.

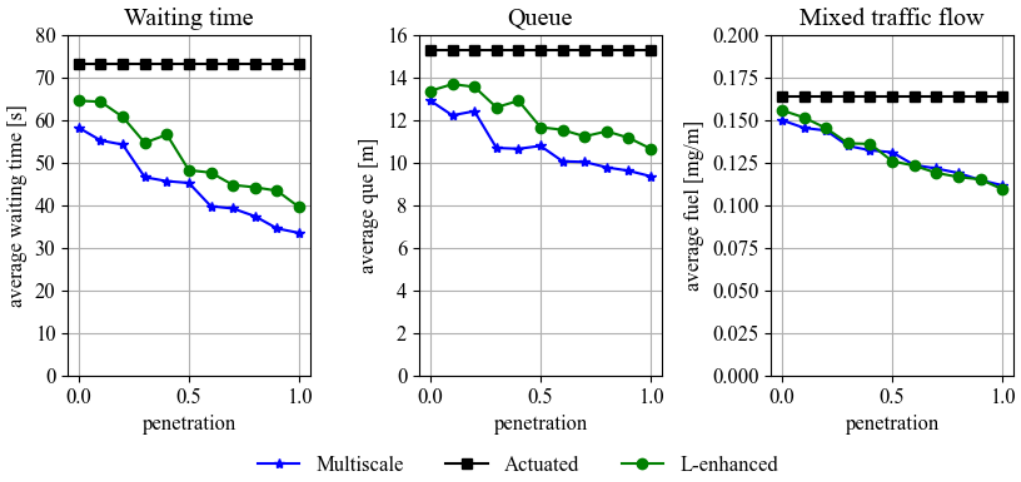

Figure 6.4: Performance of the L-enhanced method after learning on the 4 by 4 network

As shown in Figure 6.5, the performance of the L-enhanced method on the real-world 4 by 6 Seattle downtown network has similar trends with the performance on the synthetic 4 by 4 network, indicating that the L-enhanced method can be applied to different networks. The amazingly efficient performance of the L-enhanced method, however, is still based on the availability of the model-based method and repetitive runs of the model-based methods (which is time-consuming). This defeats the purpose of developing the L-enhanced method if the method has to be trained for each network separately. In the next subsection, we explore the transferability of the L-enhanced method by training it first on specific types of intersections, and then apply the trained results to networks that contain these types of intersections.

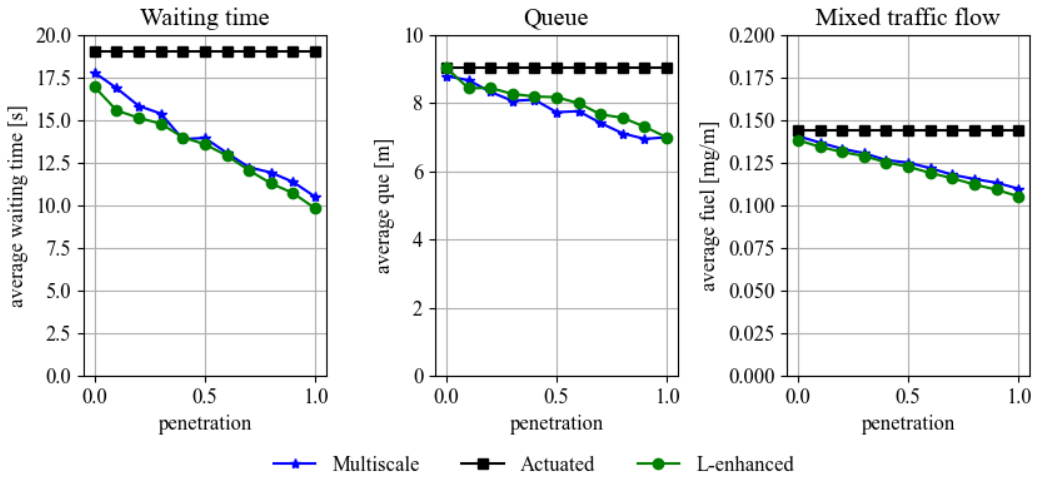

Figure 6.5: Performance of the L-enhanced method after learning on the 4 by 6 network

6.2.2 Transferability analysis

We first show that the L-enhanced method trained on a small network can be applied to larger networks. To do this, we directly apply the policies learned from the 4 by 4 network to a larger 7 by 7 network without further training. Considering that both the synthetic networks are consisted by identical intersections (i.e., only one 4-leg intersection type with three lanes for each incoming approach), we then show that a network that consists of different intersection types can be well controlled by the policies learned from separate small networks, each of which consists of only one intersection type. In other words, for this test, we learn six policies on six synthetic 4 by 4 networks separately (each network is constructed by each of the six intersection types in Figure 5.7) using the L-enhanced Algorithm 1, and directly apply the six learned policies to the real-world 4 by 6 Seattle downtown network.

The performance of applying the policies learned from the 4 by 4 network directly to the new 7 by 7 network (with the same intersection type) is shown in Figure 6.6. It shows that the L-enhanced method achieves almost the same performance as the Multiscale method (the fuel performance is even better), indicating that the policies learned from a small network

can be directly applied to a larger network. Essentially, the learning process is to mimic the actions of the Multiscale method on a single intersection basis due to the information sharing and distributed control. As long as all (or most) traffic states can be fully explored, the learned policies should work well for new traffics with same type of intersections.

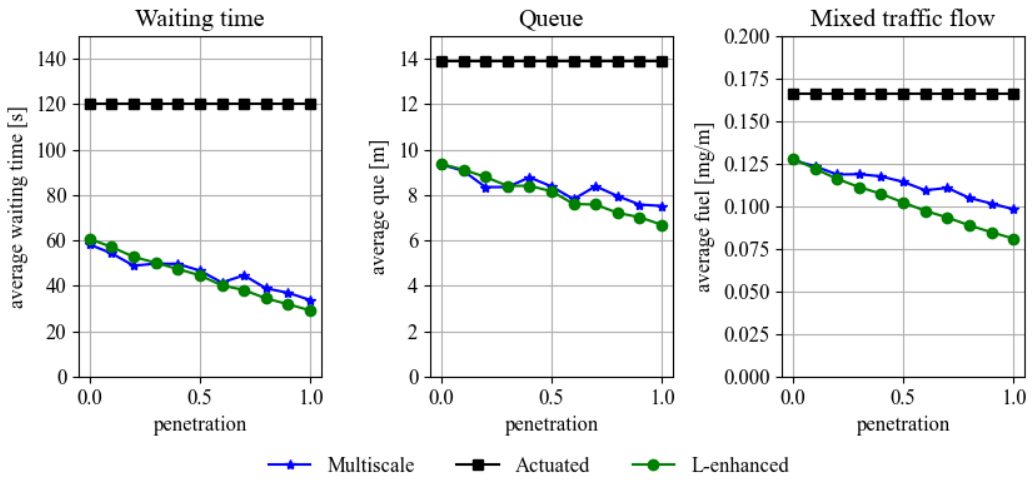

Figure 6.6: Performance of the transferred policies on the 7 by 7 network

The performance of applying the six individual policies trained in six 4 by 4 synthetic networks to the 4 by 6 Seattle downtown network is shown in Figure 6.7. Note that the general scale of the three performance indexes of this experiment is smaller to those of the 7 by 7 network, e.g., the average waiting time of the Actuated method is $120s$ for the 7 by 7 network and $20s$ for the Seattle downtown network. This is because we used a relatively high symmetric volume for the 7 by 7 network while the real volume for the Seattle downtown network is relatively medium. For the Seattle downtown network, the performance trends of the three control methods are similar to those of the 7 by 7 network. The L-enhanced method achieves almost the same performance as the Multiscale method. In other words, a complex network that consists of different types of intersections can be controlled well by the policies learned from individual small networks with identical type of intersections.

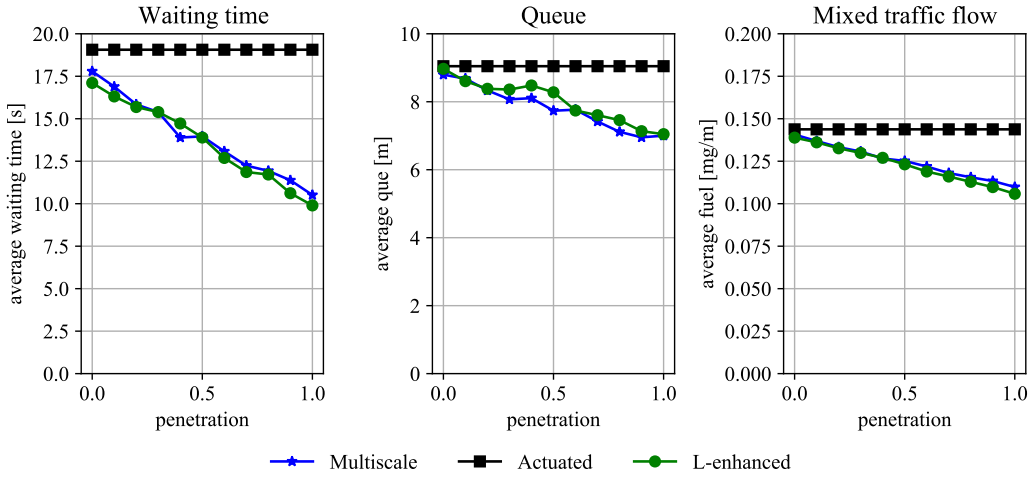

Figure 6.7: Performance of the transferred policies on the 4 by 6 Seattle downtown network

In summary, we trained individual policies for different intersection types, each under a 4 by 4 network with identical intersection type. Then, we applied the trained policies to 1) a larger network (i.e., a 7 by 7 network) with identical intersection type, and 2) a real-world network (i.e., the Seattle downtown network) with different intersection types, without further training. The experiment results showed that the learned policies performed almost the same as the Multiscale method (i.e., the "expert"), indicating that the L-enhanced method has certain level of transferability. We believe that this ability will make the training of the proposed L-enhanced method easier and more efficient, which can help apply the method to more networks broader applications. Further exploration of this transferability property is interesting and merits future research; see more discussions in Section 7.

6.3 Summary

In this section, we developed an IL-based algorithm to improve the computation efficiency of the model-based multi-scale SVCC algorithm. We built a ResNet-based policy network to represent the policies of the slower-scale signal control problem. We used the expert data generated by the model-based multi-scale SVCC algorithm to train the policy networks in a

DAGGER framework. For the faster-scale vehicle control problem, we designed a rule-based eco-driving algorithm for the mixed traffic flow and guarantee the safety using safety check technique. Numerical experiments showed that the proposed learning-enhanced SVCC algorithm could achieve almost the same performance as the benchmark model-based multi-scale SVCC algorithm, but with much higher computation efficiency. In addition, we explored the transferability of the learning-enhanced algorithm by training individual policy (i.e., individual intersection type) in small networks and apply them to large networks with different intersection types. Experiments indicated that the learning-enhanced methods had good transfer abilities.

Chapter 7

DISCUSSIONS OF PRACTICAL CONSIDERATIONS

Both the proposed model-based and learning-enhanced methods achieve good traffic and vehicle performances according to above numerical experiments. We have tried to make the experiment settings as close to the real-world conditions as possible. However, there are still some gaps in between. In this section, we discuss two of the major gaps in detail, i.e., how to consider different vehicle types and how to consider different road users, and summarize other considerations regarding real-world applications and future research directions. We firstly show how to consider different vehicle types such as internal combustion engine (ICE) vehicles, hybrid electric vehicles (HEV) and electric vehicles (EV) in Section 7.1. Considering that this might be the most urgent extension of the proposed algorithms, we conduct numerical experiments and show the initial results to illustrate the effectiveness of the proposed algorithms on such scenarios. Then, in Section 7.2, we discuss how to consider pedestrians and other road users in the proposed algorithm. Finally, we summarize other limitations of this study, the corresponding remedies, and potential directions that can further improve the proposed algorithm in Section 7.3.

7.1 Different vehicle types

So far, the vehicle performance is evaluated mainly by fuel consumption. As shown in **A3**, the goal of the vehicle control problem is to minimize fuel consumption and meanwhile fulfill the sparse trajectories generated by the signal control problem (i.e., maintain acceptable travel time). By fuel consumption, which is calculated in Equation (4.37), we assume that all vehicles are ICE-powered. However, there are other types of vehicles such as HEV and EV. In particular, CAVs are more likely to be electrical powered. The power-train systems

of different types of vehicles are different. It has been proved in this study that the proposed method could improve the performance of ICE vehicles. The effectiveness of the proposed method on other vehicle types also needs to be verified.

In this section, we apply the proposed model-based and learning-enhanced methods to other types of vehicles (i.e., HEVs and EVs). For this, we need to focus on energy consumption instead of fuel consumption. Considering that the detailed power-train systems of HEV and EV are complex, we use a power-based method to simplify and unify the energy consumption calculation of these vehicles. Specifically, we do not model the detailed power generation, split, and usage of HEV and EV; rather we focus on the raw energy consumption. In other words, we calculate the "energy" consumption instead of "electricity" consumption and "fuel" consumption of HEVs and EVs. In fact, using vehicle power to estimate the energy consumption of HEVs and EVs has been widely studied and used in the literature (e.g., Wu et al. [2015], Hu et al. [2016], Fiori et al. [2016]). In this study, we combine the two well-recognized power-based energy consumption models proposed in Wu et al. [2015], Fiori et al. [2016] (which were validated using real-world data) to estimate the raw energy consumption of HEVs and EVs. The total energy usage E for a vehicle finishing a trip with certain length is computed by integrating the power over the trip time

$$E = \int_{t_0}^{t_f} P(t)dt, \tag{7.1}$$

where $P(t)$ is the driving power, and is defined as

$$P(t) = \left(ma(t) + mgf_r \cos\theta(t) + mg \sin\theta(t) + \frac{1}{2}\rho ACv^2(t) \right) v(t). \tag{7.2}$$

where m is the mass of the vehicle, $a(t)$ is the acceleration at time t, g is the gravitational acceleration, f_r is the rolling resistance constant, $\theta(t)$ is the road slope at time t, ρ is the air density, A is the frontal area of the vehicle, C is the coefficient of air drag, $v(t)$ is the speed of the vehicle at time t. The four items in the right side parentheses of Equation (7.2) represent the acceleration force, rolling, grade, and aerodynamic resistances, respectively, which is a classical way to model vehicle dynamics in transportation (Mannering and Washburn [2020]).

One of the main differences between ICE vehicles and HEV/EV is that the energy can be regenerated under certain driving conditions such as during the braking process. Equation (7.2) actually can reflect such energy regeneration. When $P(t) > 0$, the vehicle is in traction mode, i.e., the power comes from either the electric motor or the ICE to the wheels to drive the vehicles ahead. In such a traction mode, the energy is purely consumed. When $P(t) < 0$, the vehicle is in braking mode, and the kinetic energy (which will be dissipated by the braking system in ICE vehicles) can be regenerated by HEV/EV. Through the integral operation in Equation (7.1) naturally deducts the regenerated energy from the consumed energy due to the negative values. Thus, the integral gives the final total energy consumption by taking into account the energy regeneration. Note that here we assume the energy regeneration efficiency is 100% for the sake of simplicity. It should be noted that the models and performances could be different if we adopt other energy consumption models. The objective function of the faster-scale vehicle control problem **A3** becomes

$$\min_{a^m_{i,j}(\hat{k}_f)} \sum_{i=1}^{n_j} \sum_{\hat{k}_f=k_f}^{h\frac{\Delta T_s}{\Delta T_f}+k_f-1} P^m_{i,j}(k_f)\Delta T_f + w f^m_{i,j}(k_f)\Delta T_f, \tag{7.3}$$

where i, j, m means the vehicle i on lane j at intersection m. When there are both HEV/EV and ICE vehicles in one lane, the energy consumption of each vehicle should be calculated using the correct formula based on its actual type; the objective function of the faster-scale vehicle control problem should then includes both Equation (7.3) and (4.19), but balanced by a weighting parameter that converts the fuel consumption to energy consumption, i.e., to make sure the units are consistent.

We conducted numerical experiments considering the existence of HEV and EV. We assume that the energy consumption model of all HEVs and EVs are the same, and use the same calibrated parameters in Wu et al. [2015], where $m = 1266kg, f_r = 0.006, \frac{1}{2}\rho AC = 1.30kg/m, g = 9.8m/s^2$. The CAV penetration increases from 20% to 100% using 20% as the increment, and the HEV/EV penetrations in CAV and HDV are both 50%. We test the synthetic 4 by 4 network using symmetric volume and the real-world Seattle downtown

network using real volume. Figure 7.1 shows the average waiting time, time loss, and queue length of all vehicles. It is shown that the performance of the Multiscale and L-enhanced methods follow the same trends as the above discussed case of pure ICE vehicles. In other words, the two proposed methods still work well regarding the traffic performance when HEVs/EVs are added into the traffic flow. Figure 7.2 shows the average fuel consumption of ICE vehicles and the energy consumption of HEVs/EVs under different control methods. Again, similar trends can be fount for both vehicle types: the energy consumption reduces as the CAV penetration increases. The L-enhanced method even achieves slightly better performance compared with the Multiscale method. This might be caused by the following reasons: 1) We tested the L-enhanced methods under each CAV penetration for 10 rounds using different random seeds and used the average value to plot the figures. However, due to the low computation efficiency, we only tested the Multiscale method for 1 round for each CAV penetration. There might be perturbations. 2) The the parameter that converts ICE vehicles' fuel consumption to HEVs/EVs' energy consumption were not the best. Considering that the goal of this subsection is to provide initial experiments and insights of the proposed algorithms' ability to handle different vehicle types, we believe that Figure 7.1 and 7.2 have demonstrated this purpose. In fact, the experiment of the 4 by 6 Seattle downtown network showed similar results (which is omitted here for the sake of brevity). We leave the parameter fine-tuning, model improving, and further testing considering different vehicle types as a future research task.

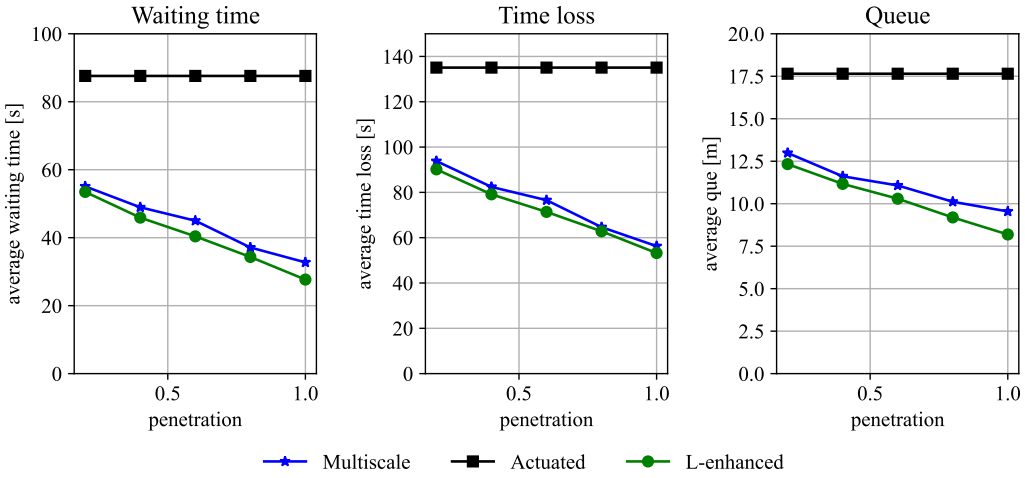

Figure 7.1: Performance of different control methods in the 4 by 4 network

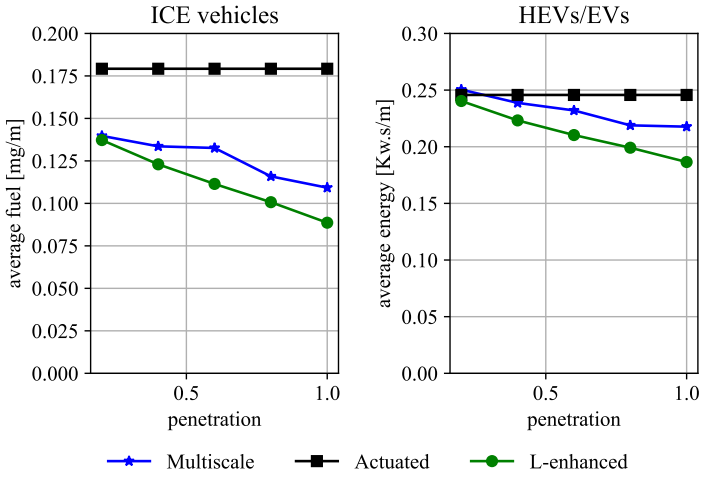

Figure 7.2: Energy consumption of different control methods in the 4 by 4 network

7.2 Pedestrians and other road users

In this study, we simplified the traffic conditions by focusing on vehicles and neglecting pedestrians, bikers, and other road users. Although this is a common assumption in traffic

signal and vehicle control, a good SVCC algorithm should be able to handle such practical considerations. In general, there are two approaches to take into account other road users besides vehicles. First and the most intuitive, we can add an exclusive pedestrian phase between a few slower-scale intervals. Second, we can integrate the pedestrian phase into the model and calculate the corresponding optimal signal phases.

For the first approach, we add an additional phase after a few slower-scale intervals (say 10, i.e., every 50 seconds if the slower-scale interval is 5 seconds) for the exclusive use of all pedestrians and no vehicular movements are allowed (Koonce and Rodegerdts [2008]). During the exclusive pedestrian phase, pedestrians can cross any leg and even the diagonal path of the intersection. Such exclusive pedestrian phase can reduce the potential conflicts between right-turning vehicles and pedestrians, which are common in traditional pedestrian intervals. However, this approach may reduce the intersection's capacity of serving the vehicles. When it comes to integrate the exclusive pedestrian phase into the proposed algorithms to provide mobility for pedestrians and other road users, there are two methods to determine when the exclusive pedestrian phase needs to be added and how long the phase should hold. The first method is to design rule-based fixed-time schedules. For example, we can create a map between the number of waited pedestrians and the waiting time to the next exclusive phase (with minimum and maximum waiting times) and the needed service time for the next exclusive phase. Such a map can be calculated using historical and experimental data. The second method is to develop an auxiliary optimization problem that runs in parallel with the proposed SVCC algorithm to adaptively determine when and how long the exclusive pedestrian phase should be, with the consideration of vehicle information. In this way, we can find a better balance between the real-time vehicular traffic condition and other road users. For example, the exclusive pedestrian phase should more likely be added when the vehicular volume is low.

For the second approach, we let the pedestrians and other road users pass the intersection together with vehicles, which is known as the leading and lagging pedestrian phases (Koonce and Rodegerdts [2008]). Pedestrians can only cross the intersection with the adjacent through

vehicular movements. Compared with the exclusive pedestrian phase, this approach have less impact of the vehicular service level. However, the right-turning vehicles may have conflicts with the pedestrians. In addition, this approach is usually used together with cycle-based traffic signals. In this study, we do not set cycles for the signals in order to bring more flexibility and optimality. Nevertheless, we can still integrate this approach into the proposed SVCC algorithm. To do this, for the slower-scale problem, we need to add minimum green time for the phases that pedestrians want to use, and design new objective functions to balance the utilities of vehicles and pedestrians. The complexity of the slower-scale problem will increase and potentially the computation efficiency will reduce. As a future research direction, testing and comparing the two approaches (i.e., exclusive pedestrian phase and integrated pedestrian phase) to find the one that best suits the proposed algorithm is in need.

7.3 Other limitations, potential remedies, and future research directions

First, in this study, we assume that the V2X communication range around a single intersection is $200m$ and information is perfectly shared, i.e., there are no latency, dropout, and packet loss, etc. However, it might be difficult to satisfy such an ideal condition in the real-world applications. Nowadays, there are two main communication standards for the V2X technologies: 1) IEEE 802.11p, e.g., the dedicated short-range communication (DSRC) protocol (Abboud et al. [2016]) and the intelligent transportation system (ITS)-G5 protocol (Atallah et al. [2015]); and 2) long term evolution (LTE), which is also known as LTE-V2X (Molina-Masegosa and Gozalvez [2017]). These techniques can work well with the $200m$ communication range. For example, the throughput could reach $600Mb/s$ and the latency could be as low as $10ms$. Such communication capacity has made it possible to transmit data between vehicles and other devices quickly and reliably. As such techniques evolve in the near future, the information transmission capacity would keep increasing. The assumption in this study, i.e., perfect information sharing, would be more and more viable.

Second, we assume that all the vehicles in the studied traffic network are rational, i.e.,

there are no accidents, emergencies, road closure, etc. However, in real-world traffic scenario, such conditions might happen due to irrational driving behavior or hardware failure. Those conditions can be considered as perturbations of the urban traffic network. The UTC system must have high robustness to the perturbations. For example, if there is a road closure or a traffic signal malfunction in the traffic network, the UTC system needs to be able to detect the perturbations and provide alternative solutions to guarantee the normal transportation functions. This will add another feature to the UTC system, i.e., the fault tolerance capability (Blanke et al. [2006]). In literature, there have already been studies regarding fault tolerance traffic control, e.g., speed limit control (Du and Razavi [2020]), perimeter control (Haddad and Mirkin [2017, 2020]), platoon control (Guo et al. [2020]), etc. However, fault tolerant control for urban traffic network has not been well-studied. How to detect and model the perturbations, how to design tolerance mechanism, and how to test different perturbation types are interesting research directions.

Third, for the faster-scale vehicle control, only longitudinal operations are considered in this study. Integrating both longitudinal and lateral (i.e., lane-changing) operations could potentially be beneficial. However, this will bring much more complexity to the model. How to select and even design a proper lateral vehicle dynamics model that can be well integrated into the proposed framework is of great interest. In addition, we used the same microscopic behavior setting (e.g., vehicle's car-following and lane-changing models) for all vehicles when we simulate the traffic in SUMO. However, in real-world condition, different drivers had different driving behaviors and in particular, CAVs' driving behavior could be different from HDVs' behavior. For example, CAVs can usually accept shorter car-following distance and reaction time due to the high-performance processors and actuators. To make the simulations closer to the real-world conditions, we need to set different driving behaviors for different vehicles. In fact, the proposed algorithms could be easily modified to accommodate different vehicle parameters. The bottleneck is that we do not have enough real-world data to unearth those parameters. We hope that in the near future, as the CV and V2X technologies develop, we could collect abundant data to help us build better microscopic traffic flow models.

Fourth, in this study, we simplify the scenario by assuming that the numbers of HDVs under the mixed traffic flow condition are known in order to estimate the full traffic states and build the slower-scale signal control problem. This assumption can be potentially relaxed. Full traffic states can be inferred by partially observed states (i.e., CAVs' states) through temporal sequential modeling and estimation methods (Qi and Chen [2021], Guo and Ma [2021]). We did not utilize such methods because 1) these methods are usually complicated which will introduce more complexity to our model; 2) full traffic state estimation is not the focus of this study; and 3) the linear interpolation used in this study performs fairly well. However, the performance of the proposed methods could potentially be improved if a more accurate estimation method is used.

Fifth, the learning enhanced methods could be further improved. In this study, we used the imitation learning as the fundamental idea to design the learning policies. The algorithm tries to mimic the expert's behavior and therefore, the expert's performance (i.e., the model-based multi-scale SVCC method) is the upper bound for the learned policies. We simplified the traffic scenario in order to build the analytical model-based algorithm, there are spaces and gaps between the idea optimal strategies and the model-based solutions. Although the learning based algorithm shows good control performances and strong generalization abilities, it can be potentially improved by more sophisticated learning methods. For example, we can use the policies learned by the current method as the initial policies, and design proper reinforcement learning algorithms to further explore different traffic states and actions to improve the policies. In this way, the learning based method can not only improve the computation efficiency as shown in this study, but also improve the optimality.

Finally, real-world testing is needed to evaluate the performance of the proposed algorithms. Although we have built the Seattle downtown network as one test case, the numerical experiments were all conducted in simulation. It will be very interesting to test the proposed algorithm in real-world settings, e.g., using CAV test-beds.

Chapter 8

CONCLUSIONS

In this study, we developed a general multi-scale control framework for UTC problems, and discussed a few key challenges and some potential research directions. We applied the framework to the two-scale SVCC problem. In order to analyze and efficiently solve the SVCC formulation, we proposed an MPC scheme to decompose and approximate the two-scale problem into two sub-problems, including a faster-scale vehicle control problem and a slower-scale signal control problem. We also proposed the concept of state consistency between the two scales, and based on which, to analyze the stability of the corresponding MPC scheme. We then presented numerical experiments to show the results of the SVCC algorithm, including the performance of the MPC scheme and comparisons with the traditional actuated signal control method.

Based on the multi-scale UTC framework and a the SVCC method for single intersection with full CAV penetration, we developed a multi-scale SVCC algorithm under the scenarios of multiple intersections and mixed traffic flow of CAVs and HDVs. We used the information sharing to enable the communication between neighboring intersections, and designed a distributed slower-scale problem for each intersection to generate the optimal phases for the traffic signal and reference trajectories for surrounding vehicles. We solved the multi-scale SVCC problem in an MPC style, and proved the asymptotic stability of the MPC under full CAV penetration. For the mixed traffic flow, we designed a vehicle state estimation method to estimate non-CAVs' states and built the corresponding slower-scale problems. For the faster-scale vehicle control problem, we developed a safety check mechanism to adjust the commands generated by the faster-scale problem. The two techniques, i.e., methods to deal with multiple intersections and mixed traffic flow, were combined together as an integrated

framework, which can be used for real-world traffic networks.

Considering that the model-based methods suffer from low computation efficiency, especially when the traffic networks are large, we developed an IL-based algorithm to improve the computation efficiency. We built the policies using the deep neural networks and trained the policies using the expert data generated by the model-based multi-scale SVCC algorithm in a DAGGER framework. In addition, we explored the generalization abilities of the learning based methods by training in small networks with identical intersections and applying directly to larger networks with different types of intersections. The numerical experiments on a single intersection, a 5-intersection corridor, a synthetic 4 by 4 network, and a real-world 4 by 6 Seattle downtown network showed that the proposed model-based multi-scale SVCC algorithm could largely improve both the traffic performance (such as average travel time and queue length) and individual vehicle's performance (such as fuel consumption). The learning enhanced algorithm could achieve almost the same performance as the model-based methods while the computation efficiency was dramatically improved, and the policies learned from small networks can be applied to large and different new networks.

We also discussed the gaps between the proposed algorithms and real-world applications. For example, how to integrate different vehicle types, how to consider pedestrians and other road users, etc. In addition, we pointed out future research directions by examining the limitations of this study. For example, integrating both longitudinal and lateral vehicle control, designing more advanced learning algorithm to improve optimality, etc. We hope this study could provide others with a potential framework to tackle the multi-scale UTC problem, and draw their interests and attentions to this important and promising area.

BIBLIOGRAPHY

Ali Aalipour, Hamed Kebriaei, and Mohsen Ramezani. Analytical optimal solution of perimeter traffic flow control based on mfd dynamics: A pontryagin's maximum principle approach. *IEEE Transactions on Intelligent Transportation Systems*, 20(9):3224–3234, 2018.

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

Khadige Abboud, Hassan Aboubakr Omar, and Weihua Zhuang. Interworking of dsrc and cellular network technologies for v2x communications: A survey. *IEEE transactions on vehicular technology*, 65(12):9457–9470, 2016.

Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3):278–285, 2003.

SMA Bin Al Islam and Ali Hajbabaie. Distributed coordinated signal timing optimization in connected transportation networks. *Transportation Research Part C: Emerging Technologies*, 80:272–285, 2017.

SMA Bin Al Islam, Ali Hajbabaie, and HM Abdul Aziz. A real-time network-level traffic signal control methodology with partial connected vehicle information. *Transportation Research Part C: Emerging Technologies*, 121:102830, 2020.

Osman D Altan, Guoyuan Wu, Matthew J Barth, Kanok Boriboonsomsin, and John A Stark. Glidepath: Eco-friendly automated approach and departure at signalized intersections. *IEEE Transactions on Intelligent Vehicles*, 2(4):266–277, 2017.

Ribal F Atallah, Maurice J Khabbaz, and Chadi M Assi. Vehicular networking: A survey on spectrum access technologies and persisting challenges. *Vehicular Communications*, 2 (3):125–149, 2015.

J Andrew Bagnell. An invitation to imitation. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Robotics Inst, 2015.

Hareesh Bahuleyan and Lelitha Devi Vanajakshi. Arterial path-level travel-time estimation using machine-learning techniques. *Journal of Computing in Civil Engineering*, 31(3): 04016070, 2017.

PG Balaji, X German, and Dipti Srinivasan. Urban traffic signal control using reinforcement learning agents. *IET Intelligent Transport Systems*, 4(3):177–188, 2010.

Lakshmi Dhevi Baskar, Bart De Schutter, and Hans Hellendoorn. Hierarchical traffic control and management with intelligent vehicles. In *2007 IEEE Intelligent Vehicles Symposium*, pages 834–839. IEEE, 2007.

Byungho Beak, K Larry Head, and Yiheng Feng. Adaptive coordination based on connected vehicle technology. *Transportation Research Record*, 2619(1):1–12, 2017.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

Mogens Blanke, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki, and Jochen Schröder. *Diagnosis and fault-tolerant control*, volume 2. Springer, 2006.

Robert Bodenheimer, Alexej Brauer, David Eckhoff, and Reinhard German. Enabling glosa for adaptive traffic lights. In *2014 ieee vehicular networking conference (vnc)*, pages 167–174. IEEE, 2014.

Kanok Boriboonsomsin, Matthew J Barth, Weihua Zhu, and Alexander Vu. Eco-routing navigation system based on multisource historical and real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1694–1704, 2012.

M Broucke and P Varaiya. The automated highway system: A transportation technology for the 21st century. *Control Engineering Practice*, 5(11):1583–1590, 1997.

Giuseppe Castorina, Maria Teresa Caccamo, Salvatore Magazù, and Liliana Restuccia. Multiscale mathematical and physical model for the study of nucleation processes in meteorology. *Atti della Accademia Peloritana dei Pericolanti-Classe di Scienze Fisiche, Matematiche e Naturali*, 96(S3):6, 2018.

Airlie Chapman and Mehran Mesbahi. Multiple time-scales in network-of-networks. In *2016 American Control Conference (ACC)*, pages 5563–5568. IEEE, 2016.

Nadeem A Chaudhary, Vijay G Kovvali, and SM Mahabubul Alam. Guidelines for selecting signal timing software. Technical report, Texas Transportation Institute, Texas A & M University System, 2002.

Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3414–3421, 2020.

Shukai Chen and Daniel Jian Sun. An improved adaptive signal control method for isolated signalized intersection based on dynamic programming. *IEEE Intelligent Transportation Systems Magazine*, 8(4):4–14, 2016.

Xianzhong Chen, Mohsen Heidarinejad, Jinfeng Liu, David Muñoz de la Peña, and Panagiotis D Christofides. Model predictive control of nonlinear singularly perturbed systems: Application to a large-scale process network. *Journal of Process Control*, 21(9):1296–1305, 2011.

Yanqiu Cheng, Xianbiao Hu, Qing Tang, Hongsheng Qi, and Hong Yang. Monte carlo tree search-based mixed traffic flow control algorithm for arterial intersections. *Transportation research record*, 2674(8):167–178, 2020.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095, 2019.

Joseph O Dada and Pedro Mendes. Multi-scale modelling and simulation in systems biology. *Integrative Biology*, 3(2):86–96, 2011.

Carlos F Daganzo and Nikolas Geroliminis. An analytical approximation for the macroscopic fundamental diagram of urban traffic. *Transportation Research Part B: Methodological*, 42 (9):771–781, 2008.

Christopher M Day and Darcy M Bullock. Detector-free signal offset optimization with limited connected vehicle market penetration: Proof-of-concept study. *Transportation Research Record*, 2558(1):54–65, 2016.

Suvranu De, Wonmuk Hwang, and Ellen Kuhl. *Multiscale modeling in biomechanics and mechanobiology*. Springer, 2015.

Lucas Barcelos De Oliveira and Eduardo Camponogara. Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1):120–139, 2010.

Yongqi Dong, Shuofeng Wang, Li Li, and Zuo Zhang. An empirical study on travel patterns of internet based ride-sharing. *Transportation research part C: emerging technologies*, 86: 1–22, 2018.

Asen L. Dontchev and R. Tyrrell Rochafellar. *Implicit Functions and Solution Mappings*. Springer, 2014.

Runjia Du, Sikai Chen, Yujie Li, Jiqian Dong, Paul Young Joun Ha, and Samuel Labi. A cooperative control framework for cav lane change in a mixed traffic environment. *arXiv preprint arXiv:2010.05439*, 2020.

Shuming Du and Saiedeh Razavi. Fault-tolerant control of variable speed limits for freeway work zone using likelihood estimation. *Advanced Engineering Informatics*, 45:101133, 2020.

Matthew Ellis, Mohsen Heidarinejad, and Panagiotis D Christofides. Economic model predictive control of nonlinear singularly perturbed systems. *Journal of Process Control*, 23 (5):743–754, 2013.

Yiheng Feng, K Larry Head, Shayan Khoshmagham, and Mehdi Zamanipour. A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55:460–473, 2015.

Neil Fenichel. Geometric singular perturbation theory for ordinary differential equations. *Journal of differential equations*, 31(1):53–98, 1979.

Chiara Fiori, Kyoungho Ahn, and Hesham A Rakha. Power-based electric vehicle energy consumption model: Model development and validation. *Applied Energy*, 168:257–268, 2016.

Ford. Energy systems d3 2017 ford f150. `https://www.anl.gov/es/energy-systems-d3-2017-ford-f150`, 2021. Accessed: 2021-12-03.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

Hongwei Ge, Yumei Song, Chunguo Wu, Jiankang Ren, and Guozhen Tan. Cooperative deep q-learning with q-value transfer for multi-intersection signal control. *IEEE Access*, 7:40797–40809, 2019.

Amir Ghiasi, Xiaopeng Li, and Jiaqi Ma. A mixed traffic speed harmonization model with connected autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 104:210–233, 2019.

Siyuan Gong and Lili Du. Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation research part B: methodological*, 116:25–61, 2018.

Noah J Goodall, Brian L Smith, and Byungkyu Park. Traffic signal control with connected vehicles. *Transportation Research Record*, 2381(1):65–72, 2013.

Wojciech W Grabowski and Piotr K Smolarkiewicz. A multiscale anelastic model for meteorological research. *Monthly Weather Review*, 130(4):939–956, 2002.

Victor Gradinescu, Cristian Gorgorin, Raluca Diaconescu, Valentin Cristea, and Liviu Iftode. Adaptive traffic lights using car-to-car communication. In *2007 IEEE 65th vehicular technology conference-VTC2007-Spring*, pages 21–25. IEEE, 2007.

S Ilgin Guler, Monica Menendez, and Linus Meier. Using connected vehicle technology to improve the efficiency of intersections. *Transportation Research Part C: Emerging Technologies*, 46:121–131, 2014.

Ge Guo, Ping Li, and Li-Ying Hao. Adaptive fault-tolerant control of platoons with guaranteed traffic flow stability. *IEEE Transactions on Vehicular Technology*, 69(7):6916–6927, 2020.

Qiangqiang Guo and Xuegang Jeff Ban. Macroscopic fundamental diagram based perimeter control considering dynamic user equilibrium. *Transportation Research Part B: Methodological*, 136:87–109, 2020.

Qiangqiang Guo, Li Li, and Xuegang Jeff Ban. Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, 101:313–334, 2019a.

Qiangqiang Guo, Xuegang Jeff Ban, and HM Abdul Aziz. Mixed traffic flow of human driven vehicles and automated vehicles on dynamic transportation networks. *Transportation research part C: emerging technologies*, 128:103159, 2021.

Yi Guo and Jiaqi Ma. Drl-tp3: A learning and control framework for signalized intersections with mixed connected automated traffic. *Transportation Research Part C: Emerging Technologies*, 132:103416, 2021.

Yi Guo, Jiaqi Ma, Chenfeng Xiong, Xiaopeng Li, Fang Zhou, and Wei Hao. Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. *Transportation research part C: emerging technologies*, 98:54–72, 2019b.

Alexandru Gurghian, Tejaswi Koduri, Smita V Bailur, Kyle J Carey, and Vidya N Murali. Deeplanes: End-to-end lane position estimation using deep neural networksa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–45, 2016.

Jack Haddad and Boris Mirkin. Adaptive perimeter traffic control of urban road networks based on mfd model with time delays. *International Journal of Robust and Nonlinear Control*, 26(6):1267–1285, 2016.

Jack Haddad and Boris Mirkin. Distributed fault tolerant perimeter control for urban road networks. *IFAC-PapersOnLine*, 50(1):4234–4239, 2017.

Jack Haddad and Boris Mirkin. Resilient perimeter control of macroscopic fundamental diagram networks under cyberattacks. *Transportation research part B: methodological*, 132:44–59, 2020.

Andrew Hamilton, Ben Waterson, Tom Cherrett, Andrew Robinson, and Ian Snell. The evolution of urban traffic control: changing policy and technology. *Transportation planning and technology*, 36(1):24–43, 2013.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Qing He, K Larry Head, and Jun Ding. Pamscod: Platoon-based arterial multi-modal signal control with online data. *Transportation Research Part C: Emerging Technologies*, 20(1): 164–184, 2012.

K Larry Head, Pitu B Mirchandani, and Dennis Sheppard. *Hierarchical framework for real-time traffic control*. Number 1360. 1992.

Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Jia Hu, Yunli Shao, Zongxuan Sun, Meng Wang, Joe Bared, and Peter Huang. Integrated optimal eco-driving on rolling terrain for hybrid electric vehicle with vehicle-infrastructure communication. *Transportation Research Part C: Emerging Technologies*, 68:228–244, 2016.

Ta-Yin Hu, Chee-Chung Tong, Tsai-Yun Liao, and Li-Wen Chen. Dynamic route choice behaviour and simulation-based dynamic traffic assignment model for mixed traffic flows. *KSCE Journal of Civil Engineering*, 22(2):813–822, 2018.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Adib Kanafani and Robert E Parsons. Program on advanced technology for the highway: Vehicle/highway research and development. In *Conference Record of papers presented at the First Vehicle Navigation and Information Systems Conference (VNIS'89)*, pages 270–272. IEEE, 1989.

David Kari, Guoyuan Wu, and Matthew J Barth. Development of an agent-based online adaptive signal control strategy using connected vehicle technology. In *17th international ieee conference on intelligent transportation systems (itsc)*, pages 1802–1807. IEEE, 2014.

Konstantinos Katsaros, Ralf Kernchen, Mehrdad Dianati, and David Rieck. Performance study of a green light optimized speed advisory (glosa) application using an integrated cooperative its simulation platform. In *2011 7th International Wireless Communications and Mobile Computing Conference*, pages 918–923. IEEE, 2011.

Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.

Ioannis G Kevrekidis, C William Gear, James M Hyman, Panagiotis G Kevrekidid, Olof Runborg, Constantinos Theodoropoulos, et al. Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

Mehdi Keyvan-Ekbatani, Xueyu Gao, Vikash V Gayah, and Victor L Knoop. Traffic-responsive signals combined with perimeter control: investigating the benefits. *Transportmetrica B: Transport Dynamics*, 7(1):1402–1425, 2019.

Petar Kokotović, Hassan K Khalil, and John O'reilly. *Singular perturbation methods in control: analysis and design*. SIAM, 1999.

Peter Koonce and Lee Rodegerdts. Traffic signal timing manual. Technical report, United States. Federal Highway Administration, 2008.

Krisada Kritayakirana and J Christian Gerdes. Autonomous vehicle control at the limits of handling. *International Journal of Vehicle Autonomous Systems*, 10(4):271–296, 2012.

Alex Kuefler, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer. Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE, 2017.

J Nathan Kutz, Xing Fu, and Steven L Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.

Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Chang Hyeong Lee and Hans G Othmer. A multi-time-scale analysis of chemical reaction networks: I. deterministic systems. *Journal of mathematical biology*, 60(3):387–450, 2010.

Joyoung Lee, Byungkyu Park, and Ilsoo Yun. Cumulative travel-time responsive real-time intersection control algorithm in the connected vehicle environment. *Journal of Transportation Engineering*, 139(10):1020–1029, 2013.

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: an approach to modeling networks. *Journal of Machine Learning Research*, 11(2), 2010.

L Li and FY Wang. A review of past 100-yera and perspective of next 50-year development of ground traffic control. *Automatica Sinica*, 44(4):577–583, 2018.

Li Li, Ding Wen, Nan-Ning Zheng, and Lin-Cheng Shen. Cognitive cars: A new frontier for adas research. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):395–407, 2011.

Li Li, Yisheng Lv, and Fei-Yue Wang. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 3(3):247–254, 2016a.

Shengbo Eben Li, Huei Peng, Keqiang Li, and Jianqiang Wang. Minimum fuel control strategy in automated car-following scenarios. *IEEE Transactions on Vehicular Technology*, 61 (3):998–1007, 2012.

Shengbo Eben Li, Qiangqiang Guo, Shaobing Xu, Jingliang Duan, Shen Li, Chengjun Li, and Kuifeng Su. Performance enhanced predictive control for adaptive cruise control system considering road elevation information. *IEEE Transactions on Intelligent Vehicles*, 2(3): 150–160, 2017.

Wan Li and Xuegang Ban. Connected vehicles based traffic signal timing optimization. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4354–4366, 2018.

Wan Li and Xuegang Ban. Connected vehicle-based traffic signal coordination. *Engineering*, 6(12):1463–1472, 2020.

Wan Li and Xuegang Jeff Ban. Traffic signal timing optimization in connected vehicles environment. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1330–1335. IEEE, 2017.

Xiaoshuang Li, Zhongzheng Guo, Xingyuan Dai, Yilun Lin, Junchen Jin, Fenghua Zhu, and Fei-Yue Wang. Deep imitation learning for traffic signal control and operations based on graph convolutional neural networks. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

Zhiyi Li, Mohammad Shahidehpour, Shay Bahramirad, and Amin Khodaei. Optimizing traffic signal settings in smart cities. *IEEE Transactions on Smart Grid*, 8(5):2382–2393, 2016b.

Zhuofei Li, Lily Elefteriadou, and Sanjay Ranka. Signal control optimization for automated vehicles at isolated signalized intersections. *Transportation Research Part C: Emerging Technologies*, 49:1–18, 2014.

Xiao Liang, S Ilgin Guler, and Vikash V Gayah. Signal timing optimization with connected vehicle technology: Platooning to improve computational efficiency. *Transportation Research Record*, 2672(18):81–92, 2018.

Qingfeng Lin, Shengbo Eben Li, Shaobing Xu, Xuejin Du, Diange Yang, and Keqiang Li. Eco-driving operation of connected vehicle with v2i communication among multiple signalized intersections. *IEEE Intelligent Transportation Systems Magazine*, 13(1):107–119, 2020.

Jennie Lioris, Ramtin Pedarsani, Fatma Yildiz Tascikaraoglu, and Pravin Varaiya. Platoons of connected vehicles can double throughput in urban roads. *Transportation Research Part C: Emerging Technologies*, 77:292–305, 2017.

Chaoru Lu, Jing Dong, Liang Hu, and Chenhui Liu. An ecological adaptive cruise control for mixed traffic and its stabilization effect. *IEEE Access*, 7:81246–81256, 2019.

Hani S Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and spatial economics*, 1(3):267–292, 2001.

Fred L Mannering and Scott S Washburn. *Principles of highway engineering and traffic analysis*. John Wiley & Sons, 2020.

Dave McKenney and Tony White. Distributed and adaptive traffic signal control within a realistic traffic simulation. *Engineering Applications of Artificial Intelligence*, 26(1): 574–583, 2013.

Mehrzad Mehrabipour and Ali Hajbabaie. A cell-based distributed-coordinated approach for network-level signal timing optimization. *Computer-Aided Civil and Infrastructure Engineering*, 32(7):599–616, 2017.

Scott Menard. *Applied logistic regression analysis*. Number 106. Sage, 2002.

Pitu Mirchandani and Larry Head. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6): 415–432, 2001.

Pitu Mirchandani and Fei-Yue Wang. Rhodes to intelligent transportation systems. *IEEE Intelligent Systems*, 20(1):10–15, 2005.

Pitu Mirchandani, Xuesong Zhou, and Jiangao Liu. Developing a multi-resolution traffic simulation platform for integrated active traffic operations evaluation in metropolitan areas. Technical report, Technical Report, 2018.

Zhaobin Mo, Rongye Shi, and Xuan Di. A physics-informed deep learning paradigm for car-following models. *Transportation research part C: emerging technologies*, 130:103240, 2021.

Rafael Molina-Masegosa and Javier Gozalvez. Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications. *IEEE Vehicular Technology Magazine*, 12(4):30–39, 2017.

Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

Rudy R Negenborn and Jose Maria Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4):87–97, 2014.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

Wei Ni and Michael Cassidy. City-wide traffic control: modeling impacts of cordon queues. *Transportation Research Procedia*, 38:284–298, 2019.

Yu Marco Nie. How can the taxi industry survive the tide of ridesourcing? evidence from shenzhen, china. *Transportation Research Part C: Emerging Technologies*, 79:242–256, 2017.

Ramin Niroumand, Mehrdad Tajalli, Leila Hajibabai, and Ali Hajbabaie. Joint optimization of vehicle-group trajectory and signal timing: Introducing the white phase for mixed-autonomy traffic stream. *Transportation research part C: emerging technologies*, 116: 102659, 2020.

William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

Agostino Nuzzolo and Antonio Comi. Advanced public transport and intelligent transport systems: new modelling challenges. *Transportmetrica A: Transport Science*, 12(8):674–699, 2016.

Jinxin Ouyang, Mengyang Li, Zhen Zhang, and Ting Tang. Multi-timescale active and reactive power-coordinated control of large-scale wind integrated power system for severe wind speed fluctuation. *IEEE Access*, 7:51201–51210, 2019.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Imitation learning for agile autonomous driving. *The International Journal of Robotics Research*, 39(2-3):286–302, 2020.

Kartik Pandit, Dipak Ghosal, H Michael Zhang, and Chen-Nee Chuah. Adaptive traffic signal control with vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 62 (4):1459–1471, 2013.

Srinivas Peeta and Hani S Mahmassani. Multiple user classes real-time traffic assignment for online operations: a rolling horizon solution framework. *Transportation Research Part C: Emerging Technologies*, 3(2):83–98, 1995.

Srinivas Peeta and Athanasios K Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and spatial economics*, 1(3):233–265, 2001.

R Phillips and P Kokotovic. A singular perturbation approach to modeling and control of markov chains. *IEEE Transactions on Automatic Control*, 26(5):1087–1094, 1981.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

Hongsheng Qi and Peng Chen. Real-time traffic flow topology sensing in partial vehicular ad hoc network: a deep learning solution. *Transportmetrica A: Transport Science*, pages 1–28, 2021.

Feng Qiao, Jin Ma, and Yaran Li. Bi-level multi time-scale voltage/var optimization and control in a hybrid distribution network. In *8th Renewable Power Generation Conference (RPG 2019)*, pages 1–7. IET, 2019.

Hesham Rakha and Raj Kishore Kamalanathsharma. Eco-driving at signalized intersections using v2i communication. In *2011 14th international IEEE conference on intelligent transportation systems (ITSC)*, pages 341–346. IEEE, 2011.

Roger P Roess, Elena S Prassas, and William R McShane. Traffic engineering, 2004.

Ugo Rosolia, Stijn De Bruyne, and Andrew G Alleyne. Autonomous vehicle control: A nonconvex approach for obstacle avoidance. *IEEE Transactions on Control Systems Technology*, 25(2):469–484, 2016.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19): 70–76, 2017.

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

Peter J Schmid, Larry Li, Matthew P Juniper, and O Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1):249–259, 2011.

Philipp Schuricht, Oliver Michler, and Bernard Bäker. Efficiency-increasing driver assistance at signalized intersections using predictive traffic state estimation. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 347–352. IEEE, 2011.

Suvrajeet Sen and K Larry Head. Controlled optimization of phases at an intersection. *Transportation science*, 31(1):5–17, 1997.

Donald R Smith. *Singular-perturbation theory: an introduction with applications*. Cambridge University Press, 1985.

Yan-Yan Song and LU Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

Weili Sun, Jianfeng Zheng, and Henry X Liu. A capacity maximization scheme for intersection management with automated vehicles. *Transportation research procedia*, 23:121–136, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, Andrew G Barto, and Ronald J Williams. Reinforcement learning is direct adaptive optimal control. *IEEE control systems magazine*, 12(2):19–22, 1992.

Mehrdad Tajalli and Ali Hajbabaie. Traffic signal timing and trajectory optimization in a mixed autonomy traffic stream. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Tie-Qiao Tang, Zhi-Yan Yi, Jian Zhang, Tao Wang, and Jun-Qiang Leng. A speed guidance strategy for multiple signalized intersections based on car-following model. *Physica A: Statistical Mechanics and its Applications*, 496:399–409, 2018.

Constantinos Theodoropoulos, Yue-Hong Qian, and Ioannis G Kevrekidis. "coarse" stability and bifurcation analysis using time-steppers: A reaction-diffusion example. *Proceedings of the National Academy of Sciences*, 97(18):9840–9843, 2000.

Stelios Timotheou, Christos G Panayiotou, and Marios M Polycarpou. Distributed traffic signal control using the cell transmission model via the alternating direction method of multipliers. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):919–933, 2014.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

Sasayuki Tsugawa, Shin Kato, Takeshi Matsui, Hiroshi Naganawa, and H Fujii. An architecture for cooperative driving of automated vehicles. In *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 00TH8493)*, pages 422–427. IEEE, 2000.

Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.

Gerard Aguilar Ubiergo and Wen-Long Jin. Mobility and environment improvement of signalized networks through vehicle-to-infrastructure (v2i) communications. *Transportation Research Part C: Emerging Technologies*, 68:70–82, 2016.

USDOT. National transportation statistics. `https://www.bts.gov/content/us-vehicle-miles`. Accessed: 2022-07-31.

Joeri Van Mierlo, Gaston Maggetto, Erik Van de Burgwal, and Raymond Gense. Driving style and traffic measures-influence on vehicle emissions and fuel consumption. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 218(1):43–50, 2004.

Tinu Vellamattathil Baby, Pouria Karimi Shahri, Amir H Ghasemi, and Baisravan Hom-Chaudhuri. Suggestion-based fuel efficient control of connected and automated vehicles. In *Dynamic Systems and Control Conference*, volume 84270, page V001T22A001. American Society of Mechanical Engineers, 2020.

Jian Wang, Srinivas Peeta, and Xiaozheng He. Multiclass traffic assignment model for mixed traffic flow of human-driven vehicles and connected and autonomous vehicles. *Transportation Research Part B: Methodological*, 126:139–168, 2019.

Jiawei Wang, Yang Zheng, Chaoyi Chen, Qing Xu, and Keqiang Li. Leading cruise control in mixed traffic flow. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 226–232. IEEE, 2020a.

Jiawei Wang, Yang Zheng, Qing Xu, Jianqiang Wang, and Keqiang Li. Controllability analysis and optimal control of mixed traffic flow with human-driven and autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(12):7445–7459, 2020b.

Rui Wang and Rose Yu. Physics-guided deep learning for dynamical systems: A survey. *arXiv preprint arXiv:2107.01272*, 2021.

Zheng Wang, Kun Fu, and Jieping Ye. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 858–866, 2018.

Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1290–1298, 2019a.

Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922, 2019b.

Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117*, 2019c.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

Tichakorn Wongpiromsarn, Tawit Uthaicharoenpong, Yu Wang, Emilio Frazzoli, and Danwei Wang. Distributed traffic signal control for maximum network throughput. In *2012 15th international IEEE conference on intelligent transportation systems*, pages 588–595. IEEE, 2012.

Guoyuan Wu, Kanok Boriboonsomsin, Wei-Bin Zhang, Meng Li, and Matthew Barth. Energy and emission benefit comparison of stationary and in-vehicle advanced driving alert systems. *Transportation research record*, 2189(1):98–106, 2010.

Xinkai Wu, David Freese, Alfredo Cabrera, and William A Kitch. Electric vehicles' energy consumption measurement and estimation. *Transportation Research Part D: Transport and Environment*, 34:52–67, 2015.

Wei Xia, Huiyun Li, and Baopu Li. A control strategy of autonomous vehicles based on deep reinforcement learning. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 198–201. IEEE, 2016.

Xiao-Feng Xie, Gregory J Barlow, Stephen F Smith, and Zachary B Rubinstein. Platoon-based self-scheduling for real-time traffic signal control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 879–884. IEEE, 2011.

Xiao-Feng Xie, Stephen F Smith, Liang Lu, and Gregory J Barlow. Schedule-driven intersection control. *Transportation Research Part C: Emerging Technologies*, 24:168–189, 2012.

Yuanhao Xiong, Guanjie Zheng, Kai Xu, and Zhenhui Li. Learning traffic signal control from demonstrations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2289–2292, 2019.

Biao Xu, Xuegang Jeff Ban, Yougang Bian, Jianqiang Wang, and Keqiang Li. V2i based cooperation between traffic signal and approaching automated vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1658–1664. IEEE, 2017a.

Biao Xu, Xuegang Jeff Ban, Yougang Bian, Wan Li, Jianqiang Wang, Shengbo Eben Li, and Keqiang Li. Cooperative method of traffic signal optimization and speed control of connected vehicles at isolated intersections. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1390–1403, 2018.

Ming Xu, Jianping Wu, Ling Huang, Rui Zhou, Tian Wang, and Dongmei Hu. Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning. *Journal of Intelligent Transportation Systems*, 24(1):1–10, 2020.

Yan Xu, Zhao Yang Dong, Rui Zhang, and David J Hill. Multi-timescale coordinated voltage/var control of high renewable-penetrated distribution systems. *IEEE Transactions on Power Systems*, 32(6):4398–4408, 2017b.

Kaidi Yang, S Ilgin Guler, and Monica Menendez. Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 72:109–129, 2016.

Zhen Yang, Yiheng Feng, and Henry X Liu. A cooperative driving framework for urban arterials in mixed traffic conditions. *Transportation research part C: emerging technologies*, 124:102918, 2021.

Handong Yao and Xiaopeng Li. Decentralized control of connected automated vehicle trajectories in mixed traffic at an isolated signalized intersection. *Transportation research part C: emerging technologies*, 121:102846, 2020.

Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys (CSUR)*, 50(3):1–38, 2017.

Bao-Lin Ye, Weimin Wu, Keyu Ruan, Lingxi Li, Tehuan Chen, Huimin Gao, and Yaobin Chen. A survey of model predictive control methods for traffic signal control. *IEEE/CAA Journal of Automatica Sinica*, 6(3):623–640, 2019.

Mehmet Yildirimoglu, Isik Ilber Sirmatel, and Nikolas Geroliminis. Hierarchical control of heterogeneous large-scale urban road networks via path assignment and regional route guidance. *Transportation Research Part B: Methodological*, 118:106–123, 2018.

Yafeng Yin, Meng Li, and Alexander Skabardonis. Offline offset refiner for coordinated actuated signal control systems. *Journal of transportation engineering*, 133(7):423–432, 2007.

Changxi You, Jianbo Lu, Dimitar Filev, and Panagiotis Tsiotras. Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, 114:1–18, 2019.

Maram Bani Younes and Azzedine Boukerche. Intelligent traffic light controlling algorithms using vehicular networks. *IEEE transactions on vehicular technology*, 65(8):5887–5899, 2015.

Chunhui Yu, Yiheng Feng, Henry X Liu, Wanjing Ma, and Xiaoguang Yang. Integrated optimization of traffic signals and vehicle trajectories at isolated urban intersections. *Transportation Research Part B: Methodological*, 112:89–112, 2018.

Yun Yuan, Qinzheng Wang, and Xianfeng Terry Yang. Modeling stochastic microscopic traffic behaviors: a physics regularized gaussian process approach. *arXiv preprint arXiv:2007.10109*, 2020.

Yun Yuan, Qinzheng Wang, and Xianfeng Terry Yang. Traffic flow modeling with gradual physics regularized learning. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Ilsoo Yun and Byungkyu Park. Stochastic optimization for coordinated actuated traffic signal systems. *Journal of Transportation Engineering*, 138(7):819–829, 2012.

Raheel Zafar, Jayashri Ravishankar, John E Fletcher, and Hemanshu R Pota. Multi-timescale voltage stability-constrained volt/var optimization with battery storage system in distribution grids. *IEEE Transactions on Sustainable Energy*, 11(2):868–878, 2019.

Guohui Zhang and Yinhai Wang. Optimizing minimum and maximum green time settings for traffic actuated control at isolated intersections. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):164–173, 2010.

Wenqing Zheng, Qiangqiang Guo, Hao Yang, Peihao Wang, and Zhangyang Wang. Delayed propagation transformer: A universal computation engine towards practical control in cyber-physical systems. *Advances in Neural Information Processing Systems*, 34, 2021.

Xing Zheng, Will Recker, and Lianyu Chu. Optimization of control parameters for adaptive traffic-actuated signal control. *Journal of Intelligent Transportation Systems*, 14(2):95–108, 2010.

Xuesong Zhou and Hani S Mahmassani. A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework. *Transportation Research Part B: Methodological*, 41(8):823–840, 2007.

Zhao Zhou, Bart De Schutter, Shu Lin, and Yugeng Xi. Two-level hierarchical model-based predictive control for large-scale urban traffic networks. *IEEE Transactions on Control Systems Technology*, 25(2):496–508, 2016a.

Zhao Zhou, Shu Lin, Yugeng Xi, Dewei Li, and Jun Zhang. A hierarchical urban network control with integration of demand balance and traffic signal coordination. *IFAC-PapersOnLine*, 49(3):31–36, 2016b.