Title

Towards Scalable Model Predictive Control for Autonomous Driving in Dynamic and Complex Environments

ISBN

9798189271434

Author

Kim, Hansung

Publication Date

2026-05-01

Peer reviewed|Thesis/dissertation

Towards Scalable Model Predictive Control for Autonomous Driving in Dynamic and Complex Environments

by

Hansung Kim

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Francesco Borrelli, Chair
Professor Negar Mehr
Professor Scott J. Moura

Spring 2026

Towards Scalable Model Predictive Control for Autonomous Driving in Dynamic and Complex Environments

Abstract

Towards Scalable Model Predictive Control for Autonomous Driving in Dynamic and Complex Environments

by

Hansung Kim

Doctor of Philosophy in Mechanical Engineering

University of California, Berkeley

Professor Francesco Borrelli, Chair

Model Predictive Control (MPC) has become a fundamental tool for motion planning and control in autonomous driving, valued for its ability to enforce constraints on states and inputs while optimizing a performance objective over a finite prediction horizon. Even as the autonomous vehicle industry increasingly delegates perception, prediction, and high-level planning to learned models from data, MPC remains a critical component of the control and planning stack at major autonomous driving companies, providing interpretable constraint enforcement, safety verification, and fallback planning that complement the primary neural planner. However, as driving scenarios grow in complexity—interacting agents with coupled decisions, surrounding vehicles exhibiting uncertain multi-modal behavior, and long planning horizons that span multiple traffic lights in urban driving—the optimization problem MPC must solve at each control step becomes computationally intractable for real-time deployment. This dissertation develops computationally efficient MPC algorithms that make optimization-based planning practical and scalable across three distinct areas of complexity in autonomous driving.

The first part addresses scalable multi-agent interactive planning using MPC. We propose a distributed MPC coordination framework for cooperative multi-vehicle lane changing, in which a designated *facilitator* vehicle proactively modifies the traffic environment to enable lane changes for the remaining vehicles in a platoon, yielding a formulation that scales with the number of agents without a corresponding growth in per-agent computational cost. We then address competitive and cooperative two-agent interactions through a game-theoretic lens: a value function learned offline from generalized Nash equilibrium solutions is embedded as the terminal cost-to-go in a short-horizon MPC policy, preserving strategic behavior while reducing online computation to a fixed-size optimal control problem that each agent solves independently.

The second part addresses scalable stochastic MPC with multi-modal trajectory predictions of surrounding vehicles, where collision-avoidance constraints grow combinatorially with the number of agents and prediction modes, preventing real-time deployment. We introduce a duality-based

screening framework that uses an attention-based recurrent neural network to predict active constraints via Lagrangian duality, enabling significant reductions in optimization problem size with open-source numerical solvers. We then extend this framework with certifiable safety guarantees, deriving conditions from strong convexity and Lagrangian duality that ensure all removed constraints remain satisfied and all eliminated decision variables are zero at optimality. The resulting algorithm is validated in complex urban driving scenarios using an open-source simulator based on real traffic data.

The third part develops a real-time energy-aware lane planning framework for connected electric vehicles that jointly optimizes longitudinal speed and lateral lane-change decisions using vehicle-to-infrastructure communication. A hybrid approach combines short-horizon optimal control with a graph-based approximation of long-term energy costs, supported by a data-driven energy model calibrated on a physical battery electric vehicle to reason about long-horizon energy optimal decisions in real-time. Vehicle-in-the-loop experiments demonstrate over 39% reduction in motion energy consumption relative to a human driver baseline and up to 24% reduction in total energy consumption, validating both the real-time deployability and energy efficiency of the proposed framework.

To my family and friends

Contents

List of Figures

List of Tables

Acknowledgments

I am deeply grateful to my advisor, Professor Francesco Borrelli, for his guidance and support throughout my Ph.D. He always welcomed my ideas and gave me the space to explore them, while providing the feedback and push needed to sharpen them into rigorous and meaningful contributions. I have grown immensely as a researcher under his mentorship, particularly in communicating complex ideas clearly and investigating multifaceted problems. I thank him for giving me the opportunity to set foot into the autonomous vehicle field. I have gained so much experience and skills that I will continue to apply them in the industry. I will carry the values and skills I learned from him into my future scientific and engineering endeavors.

I am also grateful to Professor Scott Moura for his co-advising of the ARPA-E project and for serving on both my qualifying exam and dissertation committees. His kind words of encouragement after the qualifying exam gave me assurance and confidence in my research. I always appreciate and enjoy discussing research and tech with him. I would like to thank Professor Negar Mehr for also serving on both my qualifying exam and dissertation committees. It has been a great privilege to receive such insightful feedback that strengthens my research and writing.

I would like to express my gratitude to my lab mates, who fostered a comfortable and collaborative environment in which we could grow together. In particular, I thank the ARPA-E team members—Eric Choi, Eunhyek Joa, and Hotae Lee—with whom I spent many hours working on the ARPA-E project throughout the majority of my Ph.D. They taught me how to use the platform when I knew nothing about autonomous vehicles and mentored me in both research and career decisions. A special thanks to Fionna Kopp for always being ready to lend a helping hand in the lab and pushing me to go on runs together. Your passion for running pushed me to train harder for the half and full marathons.

Moreover, I would like to thank my friends who helped me take my mind off graduate school. All the soccer games and tournaments, snowboarding trips, Friendsgivings, Discord sessions, running the half-marathon, working out together, going out, and random calls during the day fueled me and are memories that I will cherish for years to come. Thank you all for distracting and recharging me to finish my Ph.D.

Lastly, I dedicate this dissertation to my family, who have always supported me throughout my academic journey. This was unimaginable, but your unwavering support has made this accomplishment possible. My deepest gratitude goes to my life partner, Suk Ji Han, for her constant love, patience, encouragement, and resilience over the last five years, walking this journey with me through thick and thin.

Chapter 1

Introduction

1.1 Background

MPC for Autonomous Driving: Role and Computational Challenge

Autonomous driving has progressed from research prototypes to commercial deployment in recent years. Waymo launched a fully driverless ride-hailing service in Phoenix in 2020 and has since expanded to San Francisco and Los Angeles [95]. Zoox, acquired by Amazon, has developed a purpose-built autonomous vehicle without a steering wheel and is testing on public roads in Las Vegas and San Francisco [105]. Tesla has also deployed a supervised robotaxi service leveraging its vision-based Full Self-Driving system in Houston and San Francisco Bay Area [88]. Nuro has deployed autonomous delivery vehicles for last-mile logistics [77] and is planning on partnering with Uber to deploy their Nuro Driver to provide ride-hailing services in 2026, and Wayve, based in London, has pursued an end-to-end learned driving approach trained from real-world data [97] with plans to deploy ride-hailing services soon. While these companies differ in their sensing, perception, and planning stacks, all share a fundamental requirement: a motion planning and control layer that generates safe, dynamically feasible trajectories in real time under hard constraints.

Model Predictive Control (MPC) has become a fundamental tool for satisfying this requirement. At each control step, MPC solves a constrained optimization problem using a model of vehicle dynamics and a finite-horizon performance objective, applies the first element of the optimal input sequence, and re-solves at the next time step. This receding-horizon structure provides a natural mechanism for reacting to changing environments while enforcing constraints arising from vehicle dynamics, road geometry, traffic rules, and surrounding agents [12]. MPC has been applied across a range of autonomous driving tasks including path tracking [28], lane keeping, obstacle avoidance [102], and autonomous racing [61, 104].

Even as data-driven methods dominate perception and prediction, model-based principles continue to play a critical role in ensuring safety. Waymo's control architecture hybridizes MPC with neural policies for smooth, safe execution and maintains redundant safety systems that perform high-frequency fault checks [98, 96]. Approaches such as SafetyNet [92] combine machine-learned motion planners with lightweight fallback layers that validate trajectories against safety and traffic rule constraints. These architectures share a common structure where data-driven planners generate nominal trajectories while MPC provides constraint enforcement, safety verification, and fallback planning. MPC expertise continues to appear in open engineering positions at Waymo, Zoox, and other industry leaders as of 2026, further confirming its practical relevance.

However, the central challenge for MPC in autonomous driving is computational tractability. As the driving scenario grows in complexity—more surrounding vehicles, longer prediction horizons, nonlinear dynamics, and uncertain multi-modal predictions—the optimization problem that must be solved at each control step becomes larger and harder to solve within real-time requirements (typically on the order of tens of milliseconds). This dissertation is organized around three distinct areas, each presenting its own source of computational complexity, and develops algorithms that exploit the structure of each to restore real-time feasibility.

Three Areas of Scalability Challenge

Multi-Agent Coupling

In multi-agent driving scenarios, the ego vehicle must account for the behavior of surrounding agents whose actions are coupled through shared constraints such as collision avoidance. In connected autonomous vehicle (CAV) platoons, an explicit coordination mechanism can be designed to manage this coupling. Lane changing within a platoon, for example, demands that surrounding vehicles actively adjust their trajectories to create feasible gaps—a task whose complexity grows with the number of agents and whose real-time coordination through distributed MPC becomes increasingly expensive. Chapter 2 addresses this setting by introducing a facilitator-based framework in which a designated facilitator vehicle proactively modifies the surrounding traffic environment to enable lane changes for the platoon, coordinated through distributed MPC with finite state machines that scales with the number of agents.

When no such coordination structure exists, agents must independently reason about their interactions. Game theory provides a natural framework for modeling these settings by treating each agent as a player with its own objective function and constraints. The generalized Nash equilibrium (GNE)—a set of strategies from which no agent has an incentive to unilaterally deviate without violating the constraints—is a common solution concept for such problems [27]. Several computational methods have been proposed for solving constrained dynamic games in the context of multi-agent motion planning. ALGAMES [56] uses an augmented Lagrangian approach for general-sum games. Iterative linear-quadratic methods [32] approximate the game solution through successive linearizations. The Potential-iLQR approach [99] formulates multi-agent interactions as potential games and uses distributed optimization to approximate the GNE, though it is designed for cooperative settings. The DG-SQP solver [104] addresses open-loop dynamic games with general state and input constraints using sequential quadratic programming. All the aforementioned methods involve significant online cost that scales poorly with the number of agents, horizon length, and constraint complexity, and efficient computation of generalized Nash equilibria remains an active area of research. Chapter 3 addresses this bottleneck by learning a value function from offline GNE solutions and embedding it as the terminal cost-to-go in a short-horizon

MPC policy. The result is a distributed policy—each agent independently solves a fixed-size MPC problem—that emulates GNE behavior without online equilibrium computation.

Stochastic Multi-Modal Predictions

In urban driving, surrounding vehicles exhibit uncertain behavior—a vehicle approaching an intersection may turn left, proceed straight, or turn right, each with different likelihoods. Because human driver behavior is difficult to predict, motion prediction models are designed to capture this uncertainty, often representing predictions as multi-modal distributions. Many state-of-the-art prediction models consumed by downstream planners, such as MPC, represent their outputs as Gaussian mixture models to encode this uncertainty, including Wayformer [75] and MTR++ [84]. Stochastic MPC (SMPC) provides a principled framework for incorporating such probabilistic forecasts and enforcing collision-avoidance constraints with a prescribed probability level [68].

Multi-modal trajectory predictions are typically represented as marginal distributions per agent, serving as exogenous inputs to the optimal control problem. This treats surrounding vehicles' future trajectories as independent of the ego vehicle's decisions. Although agent behaviors are in principle governed by a joint distribution over all participants including the ego vehicle, marginal predictions are amenable to integration into MPC-based planning because they carry no circular dependency on the ego vehicle's own planned trajectory, making them favorable for model-based planning.

In the SMPC formulation proposed by [73], the marginal multi-modal forecasts of surrounding vehicles leads to a combinatorial growth of scenarios. For V surrounding vehicles each with M predicted modes, the number of scenario configurations is M^V, and the collision-avoidance constraints scale as $O(N \cdot M^V)$ over a prediction horizon N. Even with efficient solvers, this combinatorial growth renders the SMPC problem intractable at the control rates required for deployment. Nair et al. [73] proposed an SMPC formulation using affine disturbance feedback (ADF) policies to reduce conservatism, but the underlying constraint scaling problem remains.

Chapters 4 and 5 address this bottleneck via duality-based screening. The key insight is that at any given operating point, most collision-avoidance constraints are inactive at the optimum; enforcing them wastes computation without affecting the solution. Chapter 4 introduces RAID-Net, an attention-based recurrent neural network trained via imitation learning to predict which constraints are active using Lagrangian duality as a labeling criterion, enabling significant speedups without formal guarantees on removed constraints. Chapter 5 extends this framework to provide certifiable safety guarantees: by adding ℓ_1 regularization and deriving certification conditions from strong convexity and Lagrangian duality, the algorithm guarantees that all removed constraints remain satisfied and all removed decision variables are zero at optimality—producing a joint constraint and variable reduction with provable safety certificates.

Long-Horizon Connected Operation

Vehicle-to-Everything (V2X) communication enables vehicles to receive information—such as Signal Phase and Timing (SPaT) data from traffic lights—that extends the planning horizon far beyond what onboard sensors alone can support. V2X encompasses several communication categories: Vehicle-to-Vehicle (V2V) for sharing states and intentions to enable cooperative maneuvers such as platooning [51] and coordinated lane changes [47]; Vehicle-to-Infrastructure (V2I) for accessing traffic signal data to improve energy efficiency [7, 8]; and Vehicle-to-Cloud (V2C) for traffic flow data and over-the-air updates [33]. The U.S. Department of Transportation has outlined a plan for nationwide deployment of secure, interoperable V2X communication within the coming decade [89], and research in connected autonomous vehicles has demonstrated energy savings of 11–16% in signalized corridors through eco-driving strategies that leverage SPaT data [58, 36]. However, most eco-driving studies focus exclusively on longitudinal speed optimization in a single lane and ignore the impact of lateral decisions on energy efficiency—a gap that becomes significant on multi-lane urban roads with heterogeneous traffic.

The computational challenge here is one of horizon length: jointly optimizing speed and lane-change decisions over the extended horizons enabled by V2X yields a combinatorially large planning problem. Chapter 6 addresses this by developing a hybrid framework that pairs a short-horizon MPC with a graph-based approximation of long-term energy costs over multiple traffic signals. A data-driven energy model calibrated on a physical battery electric vehicle provides the energy estimates, and vehicle-in-the-loop experiments on a real vehicle demonstrate over 39% reduction in motion energy consumption compared to baseline driving.

1.2 Outline

The dissertation is organized in three parts. Part I, comprising Chapters 2 and 3, proposes distributed planning and control algorithms for interactive multi-vehicle environments where multiple vehicles share the drivable area with the objective of reaching their destinations while remaining safe. Chapter 2 introduces the vehicle models used in Chapters 2 and 3 and presents an algorithm for distributed multi-vehicle cooperative lane changing. Chapter 3 provides a novel framework for distributed model predictive control in interactive multi-agent scenarios based on game-theoretic reasoning, improving computational efficiency compared to existing game-theoretic approaches.

Part II, comprising Chapters 4 and 5, proposes algorithms for scalable model predictive control in complex urban scenarios with multi-modal predictions, leveraging duality and machine learning. Chapter 4 motivates the scalability challenge for stochastic model predictive control approaches that handle multi-modal predictions and proposes an algorithm for reducing the number of constraints using duality and imitation learning. Chapter 5 presents an algorithm that extends the approach proposed in Chapter 4 to enable both constraint and decision-variable reduction with safety (i.e., original constraint satisfaction) guarantees by leveraging strong convexity and duality.

Finally, Part III consists of Chapter 6, which focuses on real-time long-horizon operations of connected autonomous vehicles that leverage machine learning to enable real-time energy-aware motion planning. This chapter presents a data-driven, energy-efficient planning framework with experimental validation for real-time deployment.

Chapter 2: Distributed and Cooperative Multi-vehicle Lane Change

Based on [47], this chapter introduces the vehicle models used throughout the dissertation and presents a distributed coordination method for cooperative multi-vehicle lane change maneuvers. A designated facilitator vehicle proactively modifies the environment to enable lane changes for the platoon, coordinated through distributed MPC path-planners with finite state machines. The distributed formulation scales with the number of agents and establishes a computationally scalable multi-vehicle cooperative coordination framework.

Chapter 3: Learning Two-agent Motion Planning Strategies for MPC

This chapter, based on [50], proposes a novel approach to multi-agent motion planning by augmenting MPC with game-theoretic reasoning through embedding a value function learned from generalized Nash equilibrium solutions as the terminal cost-to-go in a short-horizon MPC policy. This replaces computationally expensive online equilibrium computation while preserving strategic behavior. The resulting policy is distributed: each agent independently solves its own fixed-size MPC problem online, with the learned terminal cost implicitly encoding the game-theoretic interactions with other agents.

Chapter 4: Learning-Guided Scalable MPC using Duality-based Criterion: Constraint Reduction

Based on [48], this chapter focuses on addressing the computational scalability challenge arising from multi-modal predictions in stochastic MPC. We use Lagrangian duality to identify active constraints and propose RAID-Net, an attention-based recurrent neural network that predicts which collision-avoidance constraints to enforce. This chapter introduces the duality-based screening framework, which enables fast and scalable MPC. We demonstrate the effectiveness of our algorithm via numerical simulation of unsignalized intersection navigation using the SMPC formulation [73] to handle uncertain, multi-modal predictions as an example. We also identify its key limitation: the lack of formal guarantees for removed constraints.

Chapter 5: Learning-Guided Scalable MPC using Convex Duality-based Criterion: Safe Variable and Constraint Reduction

This chapter addresses this limitation by extending the duality-based screening framework to provide certifiable safety guarantees. We add ℓ_1 regularization to promote sparsity in the control policy parameters and derive certification conditions from strong convexity and Lagrangian duality that guarantee all removed constraints remain satisfied, and all removed variables are zero at optimality. A transformer-based classifier trained via imitation learning guides the dual certificate computation, and the complete algorithm that chains classification, dual estimation, certification, and reduced problem formulation, enabling fast and scalable MPC for autonomous driving in complex scenarios such as urban driving. The effectiveness of the approach is demonstrated via numerical simulation on nuPlan [35] to safely accelerate the computation of the SMPC framework [73] for motion planning.

Chapter 6: Energy-Aware Lane Planning for Connected Electric Vehicles

This chapter, based on [49], proposes an energy-aware lane selection framework for connected electric vehicles that jointly optimizes longitudinal speed and lateral lane-change decisions using vehicle-to-infrastructure communication. A hybrid approach combines short-horizon optimal control with a graph-based approximation of long-term energy costs, supported by a data-driven energy model calibrated on a physical battery electric vehicle. This chapter demonstrates that the optimization-based planning tools underlying this dissertation (MPC) can be deployed for real-time and energy-efficient autonomous driving beyond safety and lane following, and provides experimental validation through vehicle-in-the-loop experiments.

Chapter 7: Conclusion

Chapter 7 concludes the dissertation by summarizing key results, highlighting our contributions, and discussing future research directions.

1.3 New Contributions

The overarching goal of this dissertation is to develop computationally efficient MPC algorithms that make optimization-based planning practical and scalable for autonomous driving in complex environments. This objective is comprised of contributions across three areas: multi-agent interactive planning, scalable stochastic MPC with multi-modal predictions, and long-horizon connected operation. The novel contributions of each chapter are summarized below.

Chapter 2: Distributed and Cooperative Multi-vehicle Lane Change

- **The facilitator concept.** We introduce the concept of a *facilitator*, a designated vehicle that proactively changes lanes first and then regulates the gap to create free space for the remaining platoon vehicles. By actively modifying the traffic environment rather than passively waiting for a feasible gap, the facilitator reduces the interaction complexity between the platoon and surrounding target vehicles and enhances the feasibility of multi-vehicle lane changes in dense traffic.

- **Three-mode finite state machine for distributed coordination.** We design a finite state machine (FSM) with three modes (Lane Keeping, Lane Change, and Gap Regulation) whose transition logic depends on the states of other connected autonomous vehicles in the platoon, not solely on the controlled vehicle. This enables coordinated sequential maneuvers across the platoon without centralized computation.

- **Mode-dependent MPC path planner.** We formulate a single constrained finite-time optimal control problem in which constraints, terminal conditions, and cost terms change depending on the FSM mode: a terminal constraint to the target lane in Lane Change mode, free-space constraints in Lane Change mode, adaptive cruise control with slack variables in Lane Keeping and Gap Regulation modes, and a gap tracking cost only in Gap Regulation mode. The resulting distributed formulation scales with the number of agents without a corresponding growth in per-agent computational cost.

Chapter 3: Learning Two-agent Motion Planning Strategies for MPC

- **Implicit Game-Theoretic MPC (IGT-MPC) algorithm.** We propose a decentralized two-agent motion planning scheme that embeds a neural network value function, learned offline from generalized Nash equilibrium (GNE) solutions, as the terminal cost-to-go in a short-horizon MPC policy. By integrating a learned value function that encodes the strategic outcomes arising from game-theoretic motion planning, IGT-MPC enables implicit interaction-aware motion planning without the need for explicit interaction models or computationally intensive online equilibrium computation.

- **Offline game-theoretic dataset generation pipeline.** We develop a data generation procedure that randomly samples joint initial conditions, solves constrained dynamic games for the GNE, and extracts reward outcomes as training targets. The reward design is problem-specific: progress advantage over the opponent for competitive racing, and collective progress along each agent's route for cooperative intersection navigation.

- **Problem-specific reward and feature design.** We design distinct reward definitions and input feature constructions for competitive settings (progress advantage, relative state features) and cooperative settings (collective progress, scenario encoding), enabling a single algorithmic framework to handle both competitive and cooperative multi-agent interactions.

- **Implicit symmetry breaking through perspective-dependent features.** We demonstrate that identical decentralized policies using the same learned value function produce asymmetric cooperative behavior, such as establishing a passing order at an intersection, through perspective-dependent feature construction. This avoids gridlock without explicit communication of intent between agents.

- **Robustness to prediction model quality.** We demonstrate that replacing oracle predictions with a constant-acceleration forecast model preserves the same qualitative strategic behavior, since the game-theoretic strategy is encoded in the learned value function at the terminal state rather than derived from the accuracy of intermediate trajectory forecasts.

Chapter 4: Learning-Guided Scalable MPC using Duality-based Criterion: Constraint Reduction

- **Duality-based screening criterion using sensitivity analysis.** We develop a sensitivity-based rule that uses the shadow price interpretation of Lagrangian dual variables to determine when a target vehicle or scenario can be safely removed from the optimization problem. The screening threshold is bounded by the maximum possible constraint violation and a user-specified cost deviation, providing an interpretable and tunable mechanism for constraint reduction.
- **RAID-Net architecture.** We propose the Recurrent Attention for Interaction Duals Network (RAID-Net), an attention-based recurrent neural network that predicts active and inactive dual classes given the scene observation. The architecture is designed with three key properties: permutation invariance to vehicle ordering and number via a graph encoding with summed feature vectors, horizon-independent memory footprint due to the shared-parameter recurrent structure, and ego-vehicle-centric encoding using time-to-collision.

Chapter 5: Learning-Guided Scalable MPC using Convex Duality-based Criterion: Safe Variable and Constraint Reduction

- **Joint constraint and variable reduction with certifiable safety.** We propose the SHIELD (Safe Hierarchical Inference for Lightweight Duality-screened MPC) algorithm, which removes both screenable constraints and decision variables (affine disturbance feedback gains) from an ℓ_1-regularized convex program while guaranteeing that all removed constraints remain satisfied and all removed variables are zero at the reduced solution.
- **Safety certification conditions from strong convexity and Lagrangian duality.** We derive certification conditions that extend the screening conditions from optimal dual variables to any feasible dual via a dual optimality-gap bound. This enables certifiable screening with approximate duals computed from a neural network, without requiring exact optimal dual solutions.
- **Two-stage SHIELD pipeline.** We design a complete algorithmic pipeline that chains neural network classification, reduced dual solve and projection, gap-based certification, and reduced primal solve. When certification fails, the algorithm defaults to solving the full problem, ensuring that the worst-case behavior is equivalent to standard MPC.

Chapter 6: Energy-Aware Lane Planning for Connected Electric Vehicles

- **Formulation of lane selection as a joint lateral-longitudinal behavior planning problem.** We formulate the lane selection problem in urban environments as a joint behavior planning task over lateral lane-change and longitudinal pass/nonpass decisions, using Signal Phase and Timing (SPaT) data and surrounding traffic information. This extends conventional eco-driving, which considers only longitudinal speed optimization in a single lane, to a multi-lane context.

- **Hybrid lane selection algorithm.** We propose a hybrid approach that combines short-horizon optimal control, solved up to the next traffic light for each candidate lane and pass/nonpass decision, with a graph-based approximation of long-term energy costs across multiple downstream traffic lights. The graph solution serves as a terminal cost-to-go, enabling the lane selector to reason about energy consequences beyond the immediate traffic light without solving a computationally intractable long-horizon problem.

- **Graph construction for multi-traffic-light planning.** We construct a directed graph with discretized velocity nodes at each traffic light and edges encoding kinetic energy changes and red-light stopping penalties. The additive cost structure enables efficient computation of the globally optimal path via Dijkstra's algorithm.

- **Insight on regenerative braking and electric vehicles.** We highlight that traditional eco-driving benefits, primarily minimizing stop-and-go driving, are diminished for electric vehicles with regenerative braking, motivating a refined understanding of energy-optimal driving behavior specific to electric drivetrains.

- **Vehicle-in-the-loop experimental validation.** We validate the proposed framework through vehicle-in-the-loop experiments using a physical battery electric vehicle embedded in virtual traffic, demonstrating a 39% reduction in motion energy consumption compared to a human driver, a 24% reduction in total energy consumption compared to a human driver, and a 7% savings over a lane-keeping eco-driving baseline.

Chapter 2

Distributed and Cooperative Multi-vehicle Lane Change

This chapter is adapted from a previous publication: Hansung Kim and Francesco Borrelli. "Facilitating Cooperative and Distributed Multi-Vehicle Lane Change Maneuvers". In: *IFAC-PapersOnLine* 56.2 (2023). 22nd IFAC World Congress

As discussed in Chapter 1, multi-agent coupling in cooperative driving scenarios requires coordination among vehicles that share the drivable space, and the computational cost of centralized planning grows prohibitively with the number of agents. This chapter presents a distributed coordination method for cooperative multi-vehicle lane change maneuvers that addresses this scaling challenge. The main contribution is the concept of a *facilitator*—a designated vehicle that proactively modifies the environment to enable lane changes for the remaining vehicles in the platoon. The coordination is implemented through distributed MPC path-planners governed by finite state machines, which avoids the need for centralized computation and yields a formulation that scales with the number of agents without a corresponding growth in per-agent computational cost. We demonstrate the effectiveness of the proposed approach through numerical simulations in dense traffic scenarios.

2.1 Introduction

By leveraging vehicle connectivity, connected autonomous vehicles (CAVs) can form platoons with short headway time and coordinate multi-vehicle maneuvers such as lane changes, resulting in increased traffic throughput, reduced motion delays, and enhanced energy efficiency [33]. In recent years, there has been growing interest in vehicle platooning and its cooperative systems as a means to improve safety, energy efficiency, and robustness of CAV controllers. For example, [51] demonstrated significant energy savings and jerk reduction using a model predictive control architecture that incorporated preceding platoon vehicle's state and acceleration forecast information received via vehicle-to-vehicle (V2V) communication while ensuring collision avoidance within a model predictive control (MPC) framework

With advancements in vehicle connectivity technology, more complex coordinated and cooperative maneuvers such as platoon merging, splitting, reconfiguration, and lane change can be implemented. In existing literature, cooperative platoon maneuver coordination is typically formulated in two ways: 1) a centralized control assuming either no uncooperative target vehicles (TV) are present in the environment, or their state forecasts are exactly known [64, 30] or 2) a distributed control assuming no target vehicles are present in the environment [55, 60]. However, target vehicles, typically human-driven vehicles, are present in the environment and their motion forecasts are uncertain in a real world environment. Moreover, platoon lane change maneuvers in [64] and [30] are executed simultaneously and are *passive and opportunistic* as the platoon adapts their policies to the changing environment. When the environment is configured such that a platoon lane change is not possible, the platoon remains in its original lane until the environment evolves into a configuration where safe execution of a lane change is possible. Simultaneous platoon lane changes have fewer opportunities to change lanes in certain traffic conditions than a single vehicle, and a larger free space is required for safe execution of the maneuver with multiple vehicles. Thus, lane change maneuvers are more complex when considering a multi-vehicle platoon interacting with target vehicles.

In this chapter, we propose a distributed coordination method to implement a novel *proactive and cooperative* multi-CAV platoon lane change strategy. In the proposed strategy, a CAV performs a role of a *facilitator*. The *facilitator* actively interacts and modifies the environment to aid the multi-CAV platoon in executing a lane change while reducing the interaction between the multi-CAV platoon and target vehicles. A multi-CAV platoon refers to a single-lane platoon with a small number of interconnected vehicles. Our approach to distributed coordination uses three *modes* (*Lane Keeping*, *Lane Change*, *Gap Regulation*) to describe the CAV maneuver modes. Finite state machines which control the mode transitions of each CAVs are also used to coordinate CAV maneuvers. The mode transition logic depends on not only the state of the controlled CAV but also on those of other CAVs in the multi-CAV platoon by design. In each mode, distinct constraints and cost functions in distributed MPC path-planners are designed to execute the desired maneuvers for multi-CAV coordination.

We demonstrate that the proposed lane change strategy enhances feasibility of multi-CAV platoon lane change compared to the *passive and opportunistic* multi-CAV platoon lane change strategy through numerical simulation. In numerical simulation, a real-world NGSIM dataset [71] is used to generate realistic initial conditions for the set of CAVs and target vehicles under consideration. The contributions are summarized as follows:

1. A novel proactive and cooperative multi-CAV platoon lane change strategy which relies on the role of a *facilitator*.

2. A distributed coordination method consisting of a MPC based path-planner with three modes—in which different terminal constraints and safety constraints as well as cost functions are used—and a finite state machine that controls the mode transitions of the CAVs.

3. A simulation study demonstrating enhanced feasibility of multi-CAV platoon lane change using the proposed strategy compared to opportunistic simultaneous multi-CAV lane change

strategy.

The remainder of the chapter is organized as follows. Section 2.2 introduces the concept of a *facilitator* and its role in *proactive and cooperative* multi-CAV platoon lane change strategy. Section 2.3 discusses the vehicle dynamic model and the target vehicle prediction model used in our MPC path-planner formulation. Section 2.4 discusses the distributed coordination method including the higher-level FSM and distributed MPC path-planner used to implement the proposed lane change strategy. The numerical simulation setup and results of multi-vehicle lane change in dense traffic is presented in Section 2.5. Finally, Section 2.6 concludes the chapter and discusses future research directions.

2.2 The Concept of a Facilitator

In a scenario with a multi-CAV platoon of n-CAVs traveling in the same lane, the platoon must interact with target vehicles when executing a maneuver to change into the Target Lane (TL) from the Original Lane (OL). For instance, the multi-CAV platoon (orange vehicles) of 3 CAVs in the environment shown in Fig. 2.1(a) must consider TV 2 and TV 3's positions and velocities to determine the feasibility of a lane change. The *passive and opportunistic* lane change strategy can result in the multi-CAV platoon only changing lanes when the target vehicles' positions and velocities are such that lane change is simultaneously feasible for all CAVs in the multi-CAV platoon (i.e. if lane change is feasible for all but one vehicle, such as CAV 3, the multi-CAV platoon does not change lanes). Under certain traffic conditions, this approach may prevent the platoon from changing lanes for a long time. Our proposed lane change strategy introduces a *facilitator* that proactively manipulates the environment in favor of the multi-CAV platoon. The facilitator changes lanes when feasible and regulates the free space between itself and the vehicle in front of it, allowing the other CAVs in the platoon to change lanes into the free space without considering target vehicles. This reduces the complexity of interaction between the platoon and target vehicles and potentially enhances the feasibility of multi-vehicle lane change under certain traffic conditions. In order to implement the *proactive and cooperative* lane change strategy, a sequence of coordinated maneuvers must be executed by each CAVs in the platoon as shown in Fig. 2.1(a)-(d).

2.3 Vehicle Models

We first describe the kinematic bicycle model of nonlinear vehicle dynamics in the Frenet frame. Then, we describe the surrounding prediction model used in our path-planner formulation. In this work, we assume the OL center line and curvature information are known.

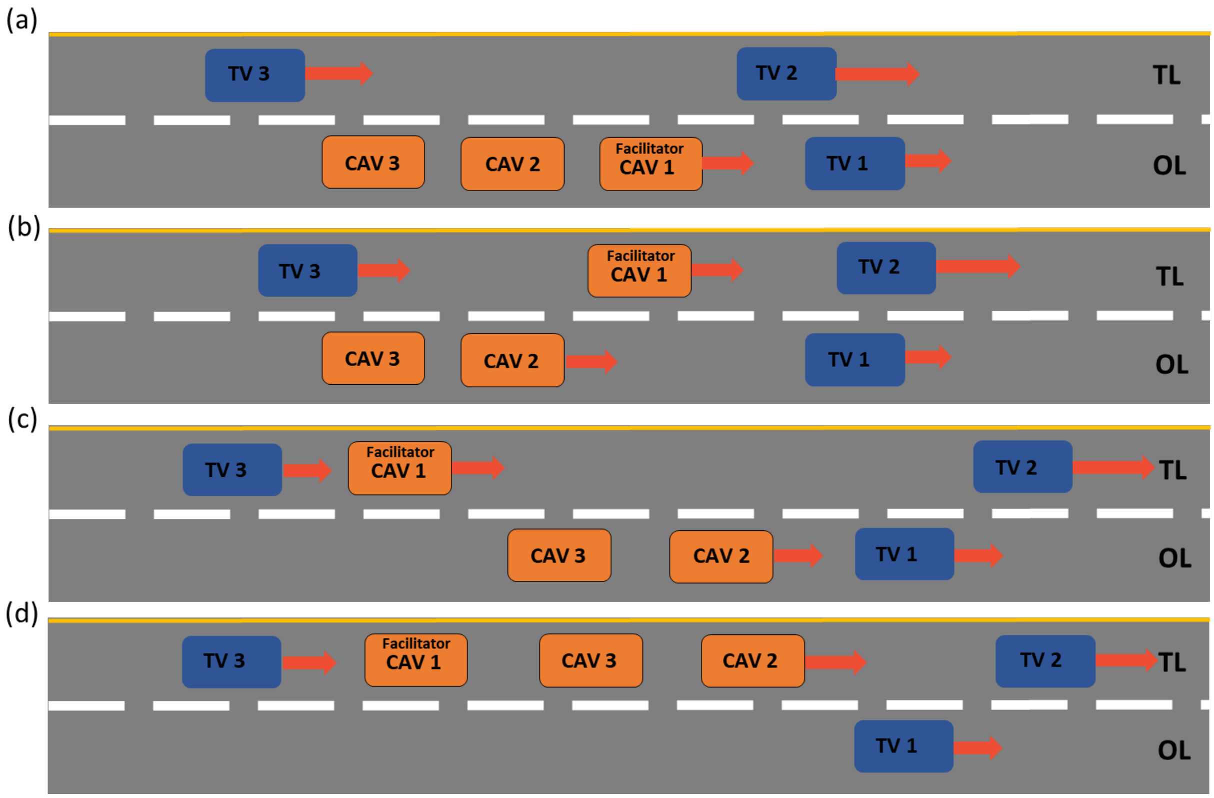


Figure 2.1: Snapshots of the proactive and cooperative lane change strategy: (a) the facilitator changes lane into the Target Lane (TL), (b) CAV 1 and CAV 2 accelerate while the facilitator decelerates to create free space for CAV 1 and CAV 2, (c) lane change for CAV 1 and CAV 2 is rendered feasible, (d) CAV 1 and CAV 2 complete a lane change

Vehicle Dynamics

The i-th CAV state variables are defined as $\mathbf{z}^i(t) := [s^i(t), e^i_y(t), e^i_\psi(t), v^i(t)]^\top$ for $i \in \{1, 2, ..., n\}$, where $s^i(t)$ and $e^i_y(t)$ represent the longitudinal and lateral displacement of the vehicle with respect to the desired lane's center line, respectively, as seen in Fig. 2.2. n is the number of CAVs in the platoon. $e^i_\psi(t)$ is the heading angle difference between that of the vehicle and the center line and v^i is the longitudinal velocity of the vehicle at the center of gravity (C.G.). The control input vector is $\mathbf{u}^i(t) = [a^i(t), \delta^i_f(t)]^\top$, where $a^i(t)$ is the longitudinal acceleration of the vehicle at the C.G. and

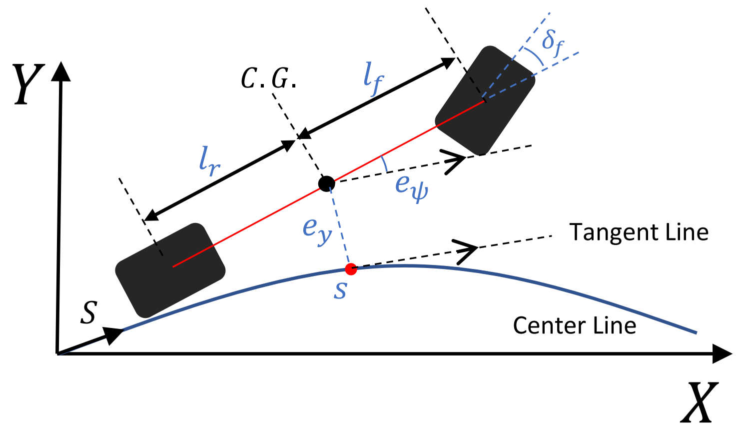


Figure 2.2: The kinematic bicycle model in Frenet Frame

$\delta_f^i(t)$ is the steering angle. The vehicle dynamics are

$$\begin{aligned} \dot{s}^i(t) &= v^i(t)\,\tfrac{cos\, e_\psi^i(t)}{1-e_y^i(t)\,\kappa(t)}, \\ \dot{e_y}^i(t) &= v^i(t)\, sin\, e_\psi^i(t), \\ \dot{e_\psi}^i(t) &= v^i(t)\,(\tfrac{tan\,\delta_f^i(t)}{l_f^i+l_r^i} - \tfrac{\kappa(t)\, cos\, e_\psi^i(t)}{1-\kappa(t)\, e_y^i(t)}), \\ \dot{v}^i(t) &= a^i(t), \end{aligned} \tag{2.1}$$

l_f^i and l_r^i are distance from the front axle to the C.G. and distance from the rear axle to the C.G., respectively. $\kappa(t)$ is the curvature of the reference centerline.

The continuous-time dynamics model is discretized using forward Euler method with sampling time Δt as follows

$$\mathbf{z}^i(k+1) = f(\mathbf{z}^i(k), \mathbf{u}^i(k)) \tag{2.2}$$

where $\mathbf{z}^i(k)$ is the state vector $\mathbf{z}^i$ at time $k\Delta t + t$ for $k \in \mathbb{Z}_0^+$.

Target Vehicle Prediction Model

During the prediction horizon N, we use a longitudinal uncertain prediction model to predict target vehicle's motion. The model assumes uncertainty on acceleration and that the target vehicle does not change lanes within the prediction horizon. Given the OL centerline and road curvature information: $(\kappa_t, \ldots, \kappa_{t+N})$, the resulting discretized dynamics of the target vehicle is

$$\begin{aligned} s^{TV}_{k+1|t} &= s^{TV}_{k|t} + \Delta t\, v^{TV}_{k|t}\, \tfrac{cos\, e^{TV}_{\psi,k|t}}{1-e^{TV}_{y,k|t}\kappa_k}, \\ e^{TV}_{y,k+1|t} &= e^{TV}_y(t), \\ e^{TV}_{\psi,k+1|t} &= 0, \\ v^{TV}_{k+1|t} &= v^{TV}_{k|t} + \Delta t\; w_k, \end{aligned} \tag{2.3}$$

for $k = t, ..., t+N$; where $e^{TV}_y(t)$ is e_y of the target vehicle at time t. $w_k \sim \mathcal{N}(0, \sigma^2)$ is a random variable representing acceleration at time k. The variance σ^2 is the design parameter of the model and depends on the acceleration limitation of the target vehicle. The proposed approach can accommodate different prediction models for improved prediction accuracy and robustness.

2.4 Distributed Coordination

A distributed coordination of CAVs to implement a proactive and cooperative lane change strategy is designed as follows: MPC Path-planners with three modes in which different terminal constraints and safety constraints as well as cost functions are used to plan trajectories of each CAVs. Each CAVs has a corresponding finite state machine that manages the mode transitions of the planners and coordinates multi-CAV platoon maneuvers. In this section, the FSM and MPC path-planner are formulated.

Finite State Machine

The CAVs in the platoon are indexed by i starting from the front-most CAV. Without loss of generality, we assign the facilitator role to CAV $i = 1$. The FSM makes use of parameters and binary inputs associated with every CAVs in the platoon which are denoted by the superscript $i \in \{1, 2, ..., n\}$ for the i-th CAV. The parameters and binary inputs are listed in Table 2.1.

The desired minimum longitudinal inter-vehicular signed distances are defined as the following:

$$d^1_{des} = n \cdot d_{safe} \tag{2.4}$$

$$d^i_{des} = -i \cdot d_{safe}, \text{ for } i \neq 1 \tag{2.5}$$

where d_{safe} is the user-defined safe vehicular distance and impacts the conservativeness of the controlled vehicle's driving behavior. In the context of multi-CAV lane change, d^1_{des} represents the desired free space that the facilitator must create. d^i_{des} corresponds to the desired distance (relative to the facilitator) for the i-th CAV to execute a lane change into the free space created by the facilitator. These inter-vehicular signed distances fully describe the relative positions of the multi-CAV platoon and the facilitator.

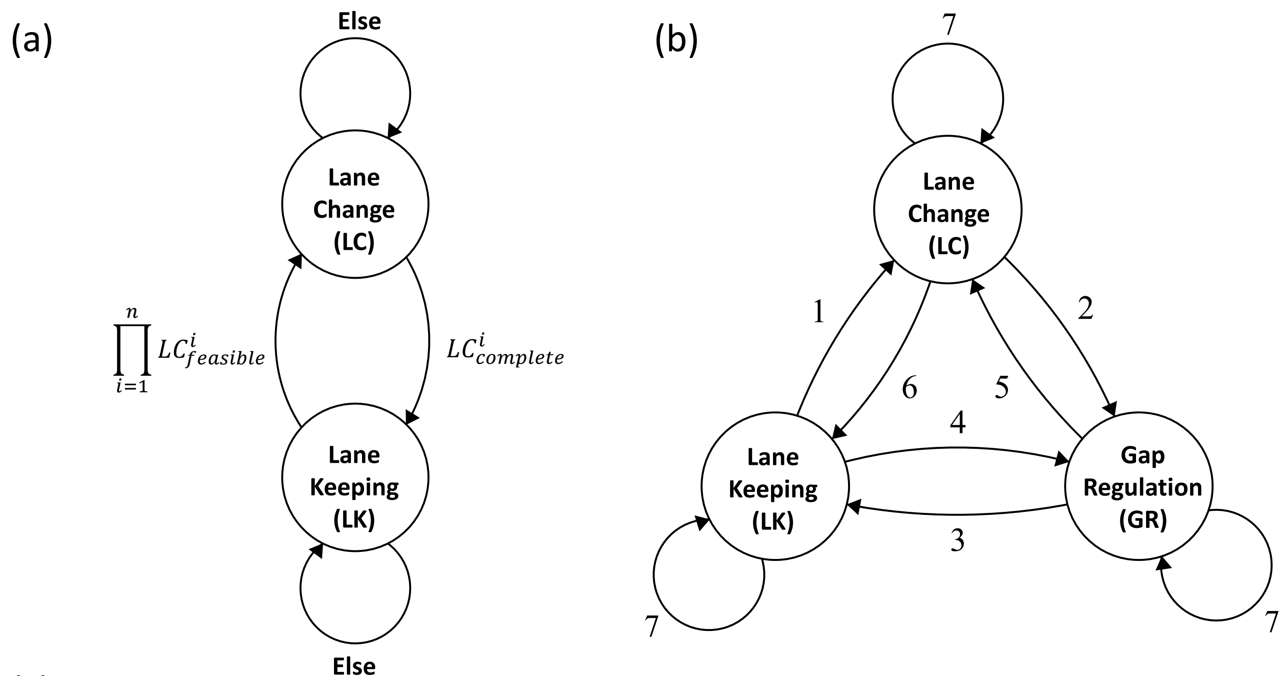
Typically, autonomous highway driving can be categorized into two modes: *Lane Keeping* (*LK*) and *Lane Change* (*LC*). In *LK* mode, the CAV maintains its lane and follows a reference

Table 2.1: FSM parameters and inputs

Parameter	**Definition**
TL	Target lane for lane change
OL	Original lane of the platoon
Δs_{TL}	Distance from the facilitator to the preceding vehicle in the TL
Δs_{OL}	Distance from the facilitator to the preceding vehicle in the OL
d^1	$\min(\Delta s_{TL}, \Delta s_{OL})$
d^i	Signed-distance from i-th CAV to the facilitator, for $i : [2, n]$
d^1_{des}	Desired minimum distance from the facilitator to the preceding vehicles in TL or OL
d^i_{des}	Desired minimum signed distance from i-th CAV to the facilitator for $i : [2, n]$
FSM binary inputs	**Definition**
$LC^i_{feasible}$	Lane change feasibility of the i-th CAV in the platoon
$LC^i_{complete}$	Lane change completion of the i-th CAV in the platoon
$LC^P_{complete}$	$\prod_{i=2}^{n} LC^i_{complete}$
$r^i_{satisfied}$	$d^i \geq d^1_{des}$ satisfied

velocity while keeping a safe distance from the preceding vehicle. In *LC* mode, the CAV attempts to safely change lanes into the TL. An FSM for the transition between the two modes in a multi-CAV platoon is shown in Fig. 2.3(a). This FSM is sufficient for a *passive and opportunistic* lane change strategy, in which the platoon changes lanes only if it is simultaneously feasible for all vehicles ($\prod_{i=1}^{n} LC^i_{feasible}$ is *True*). After completing the lane change, the planner switches back to *LK* mode.

Our proposed FSM has an additional mode: *Gap Regulation* (*GR*). In *GR* mode, the CAV regulates the inter-vehicular signed distance of interest (gap) to the desired longitudinal distance set according to (2.4)-(2.5). This reconfigures the multi-CAV platoon into a desired formation in a distributed manner. The mode transition depends on not just the controlled CAV's states but also on the states of other CAVs as a mechanism for coordination. For instance, the multi-CAV platoon (excluding the facilitator) switches to *GR* mode only after the facilitator has completed a lane change into the TL. In this mode, the *facilitator* modifies the environment and creates free space in the TL for the rest of the multi-CAV platoon to change lanes while they get in desired formation to execute a simultaneous lane change as shown in Fig. 2.1(b)-(c).



(c)

Transition	Facilitator	Platoon (non-Facilitators)
1	$\sim LC^1_{complete} \wedge LC^1_{feasible}$	$\sim LC^P_{complete} \wedge LC^P_{feasible} \wedge (LC^1_{feasible} \mid LC^1_{complete})$
2	$LC^1_{complete} \wedge \sim r^1_{satisfied} \wedge \sim LC^P_{complete}$	N/A
3	$(LC^1_{complete} \wedge r^1_{satisfied}) \mid LC^P_{complete}$	N/A
4	$LC^1_{complete} \wedge \sim r^1_{satisfied} \wedge \sim LC^P_{complete}$	$LC^1_{complete} \wedge \sim r^i_{satisfied} \wedge \sim LC^i_{complete}$
5	N/A	$LC^P_{feasible} \wedge r^i_{satisfied} \wedge \sim LC^i_{complete}$
6	$(LC^1_{complete} \wedge r^1_{satisfied}) \mid \sim LC^1_{feasible}$	$LC^i_{complete} \mid \sim LC^i_{feasible}$
7	Else	Else

Figure 2.3: Finite state machines of path planner modes in a highway driving scenario: (a) a FSM representing a passive and opportunistic lane change strategy (b) FSM with an additional mode—*Gap Regulation*—and its switching conditions listed in (c) for the proposed proactive and cooperative lane change strategy.

The mode transition logic for the facilitator and the non-facilitator CAV are listed in Fig. 2.3(c) in separate columns. We denote the FSM mode of the i-th CAV at time t as $z^i_{FSM}(t) \in \{LK, LC, GR\}$ and the vector containing FSM binary inputs listed in Table 2.1 for all CAVs as $\mathbf{x}_{FSM}$. The i-th CAV's FSM mode transition at time t is denoted as

$$z^i_{FSM}(t + \Delta t) = \tilde{f}^i(z^i_{FSM}(t), \mathbf{x}_{FSM}), \tag{2.6}$$

Note that the proposed FSM in Fig. 2.3 and Table 2.1 also allows simultaneous platoon-wise lane change when feasible by design. Further, in cases of infeasibilities while in *LC* mode, the mode transitions back to *LK* mode and returns to OL by design for safety.

MPC Path Planner

Path planning is formulated as a constrained finite time optimization problem (CFTOP) to plan dynamically feasible, smooth, and collision-free trajectories for each vehicle. The MPC path-planner solves the CFTOP repeatedly in receding horizon at a constant frequency. The path-planner is designed using the MPC framework with an augmented state vector $\tilde{\mathbf{z}}^i_{k|t} := [{\mathbf{z}^i_{k|t}}^\top, d^i_{k|t}]^\top$ where $\mathbf{z}^i_{k|t}$ is defined as $\mathbf{z}$ of the i-th vehicle at time k predicted at time t. $d^i_{k|t}$ is d^i at time k predicted at time t with the following dynamics

$$d^1_{k+1|t} = s^{TV}_{k+1|t} - s^1_{k+1|t}, \tag{2.7}$$

$$d^i_{k+1|t} = s^1_{k+1|t} - s^i_{k+1|t}, \text{ for } i \neq 1. \tag{2.8}$$

Therefore, the augmented state dynamics $\tilde{\mathbf{z}}^i_{k+1|t} := g(\tilde{\mathbf{z}}^i_{k|t}, \mathbf{u}^i_{k|t}, w_k)$ is represented by (2.2) and $d^i_{k+1|t}$ which depends on the target vehicles' prediction with uncertainty in (2.3) when $i = 1$. For $i \neq 1$, $d^i_{k+1|t}$ is deterministic because it is the inter-vehicular distance between two CAVs. In a distributed multi-CAV system, the exact motion forecasts of other CAVs are communicated via V2V. Using the augmented state dynamics, the path-planner is defined as follows

$$\min_{\substack{\tilde{\mathbf{z}}^i(\cdot|t),\\ \mathbf{u}^i(\cdot|t),\\ \epsilon^i(\cdot|t)}} \sum_{k=t}^{t+N} \mathrm{E}[\|Q(\tilde{\mathbf{z}}^i_{k|t} - \tilde{\mathbf{z}}^{i,des}_{k|t})\|_2^2] + c_s \cdot \epsilon^i_{k|t} + \sum_{k=t}^{t+N-1} \|\mathcal{R}_u \mathbf{u}^i_{k|t}\|_2^2 + \|\mathcal{R}_{\Delta \mathbf{u}} \Delta \mathbf{u}^i_{k|t}\|_2^2 \tag{2.9a}$$

$$\begin{aligned}
\text{s.t.} \quad & \tilde{\mathbf{z}}^i_{k+1|t} = g(\tilde{\mathbf{z}}^i_{k|t}, \mathbf{u}^i_{k|t}, w_k), && \text{(2.9b)}\\
& \Delta \mathbf{u}^{min}_{k|t} \leq \Delta \mathbf{u}^i_{k|t} \leq \Delta \mathbf{u}^{max}_{k|t}, && \text{(2.9c)}\\
& \mathbf{u}^i_{k|t} \in \mathcal{U}, \quad \forall k = t, ..., t+N-1 && \text{(2.9d)}\\
& \tilde{\mathbf{z}}^i_{t|t} = \tilde{\mathbf{z}}^i(t) && \text{(2.9e)}\\
& \mathbf{z}^i_{k|t} \in \mathcal{Z}, \quad \forall k = t, ..., t+N && \text{(2.9f)}\\
& \textbf{if } z^i_{FSM}(t) \in \{LC\} : && \\
& \mathbf{z}^i_{t+N|t} \in \mathcal{Z}_f && \text{(2.9g)}\\
& \mathrm{E}[s^{i,min}_{k|t}] + d_{safe} \leq s^i_{k|t} \leq \mathrm{E}[s^{i,max}_{k|t}] - d_{safe} && \text{(2.9h)}\\
& \textbf{if } z^i_{FSM}(t) \in \{LK, GR\} : && \\
& d_{safe} - \epsilon^i_{k|t} \leq \mathrm{E}[s^i_{front,k|t}] - s^i_{k|t} && \text{(2.9i)}\\
& \epsilon^i_{k|t} \geq 0, \quad \forall k = t, ..., t+N && \text{(2.9j)}
\end{aligned}$$

where $\mathbf{u}^i(\cdot|t) = \{\mathbf{u}^i_{t|t}, ..., \mathbf{u}^i_{t+N-1|t}\}$ is the control input sequence over the horizon for the i-th vehicle, N. Furthermore, $\Delta\mathbf{u}^i_{k|t} := \mathbf{u}^i_{k|t} - \mathbf{u}^i_{k-1|t}$ and $\tilde{\mathbf{z}}^{i,des}_{k|t}$ is the desired state of the i-th vehicle which is determined by the lane center line and the speed limit of the road. $Q := diag(0, c_y, c_\psi, c_v, c_d)$ where each scalar elements represent the tracking costs for e^i_y, e^i_ψ, v^i, and d^i, respectively. $\mathcal{R}_u$ $\mathcal{R}_{\Delta u}$ are the input cost matrix and input rate cost matrix, respectively. Note that these cost matrices are positive definite matrices and tuning parameters. (2.9e) sets the initial condition at time *t*. $\mathcal{Z}$ is a feasible state set representing the road boundaries and vehicle dynamics limitations, $\mathcal{U}$ is a feasible input set, and $\Delta\mathbf{u}^{min}_{\cdot|\cdot}$ and $\Delta\mathbf{u}^{max}_{\cdot|\cdot}$ are input rate limits that arise from vehicle dynamic limitations. $z^i_{FSM}(t)$ is the FSM mode of the i-th CAV at time t at which the CFTOP (2.9) is solved. The FSM mode is fixed during the horizon and its transition is described by (2.6).

As previously stated, the planner has three *modes* with different constraints and tracking costs. Firstly, in *LC* mode, a state terminal constraint (2.9g) is applied. The terminal state set, $\mathcal{Z}_f$, is determined by the TL's lateral displacement from the OL and curvature at $s^i_{N|t}$ and requires the controlled vehicle to reach the target lane at the end of the horizon. (2.9h) is applied to constrain the vehicle to the free space in the target lane and avoid collisions. The $s^{i,min}_{k|t}$ and $s^{i,max}_{k|t}$ are the longitudinal displacement of the rear and front (with respect to that of the controlled vehicle) vehicles in the TL at time k predicted at time t, respectively, if any. The expectation-type constraint is applied because $s^{i,min}_{k|t}$ and $s^{i,max}_{k|t}$ may be stochastic target vehicle predictions. This constraint ensures that the controlled vehicle remains in free space in the TL while changing lanes within the horizon. Secondly, in *LK* and *GR* mode, (2.9i) and (2.9j) are applied for safe adaptive cruise control if there exists a preceding vehicle (its longitudinal position denoted as $s^i_{front,k|t}$) in the same lane as the controlled vehicle. Otherwise, (2.9i) and (2.9j) are not applied. To prevent infeasibility occurring from target vehicles' sudden deceleration, the safety constraint (2.9i) is relaxed by the time-varying slack variable $\epsilon^i_{k|t}$ with the cost c_s. In *LC* mode, the desired e_y is the lateral-displacement of the TL whereas it is zero (OL) in other modes. Lastly, the tracking cost for $d^i_{k|t}$ is only included in *GR* mode. Accordingly, $c_d = q_d$ if in *GR* mode and $c_d = 0$ otherwise, where $q_d \in \mathbb{R}^+$ is a tracking cost for gap regulation and a design parameter.

2.5 Numerical Example

We validate the proposed distributed coordination method and demonstrate the enhanced feasibility of platoon lane change using the *proactive and cooperative* multi-CAV platoon lane change strategy in comparison to that of using *passive and opportunistic* strategy (the baseline) through numerical simulations. In the baseline strategy, the multi-CAV will change lanes only when it is feasible for all three CAVs simultaneously. The nonlinear optimization problem (2.9) is modeled using CasADi [2] and solved using IPOPT [16]. In this section, the target vehicle controller and the numerical simulation scenario setup are discussed.

The assumptions in our numerical simulations include the following.

1. The platoon has 3 CAVs with identical lengths

2. The *Facilitator* role is preassigned to CAV 1
3. An undirected communication network is established between the CAVs

Target Vehicle Controller

To close the loop in our simulation, the target vehicles are controlled using the Enhanced Intelligent Driver Model (EIDM) to emulate an adaptive cruise control behavior. The EIDM relaxes sudden deceleration in situations that are not safety-critical while inheriting the crash-free characteristics of the Intelligent Driver Model's (IDM) [44]. This model combines the IDM and constant-acceleration heuristics, which assumes that the leading vehicle will not change its acceleration for the next few seconds, to relax the sudden deceleration in non-safety-critical situations [44]. Furthermore, a scaling constant called *coolness factor*, c, determines the sensitivity with respect to changes in the gap with the vehicle in front. [44] recommend $c \in [0.95, 1.00]$ to simulate a realistic driving behavior. The EIDM parameters used in the simulation are reported in Table 2.2. s_0 represents the minimum desired spacing between the controlled vehicle and the preceding vehicle. v_0 is the desired velocity, T is the desired time headway, a is the maximum vehicle acceleration, and b is the comfortable braking deceleration where $b > 0$.

NGSIM Dataset Simulation

Our proposed planner was tested in a dense traffic scenario based on reconstructed NGSIM dataset [71]. The NGSIM dataset contains the recorded trajectories (sampling frequency of 10 Hz) of all vehicles passing a segment of freeway I-80 for 15 minutes [71]. The initial scene is generated by randomly selecting the initial frame and two adjacent lanes (i.e. lane 1 and 2) out of 5 total lanes. Then, three vehicles with the shortest inter-vehicle distances in the initial scene are selected as CAVs. The inter-vehicle distances must be smaller than the user-defined threshold which is set to 9 meters in our simulation. If the threshold condition is not satisfied, the scene is discarded and generated again randomly. After selecting the CAVs, the desired velocity of all CAVs and target vehicles are set to the maximum velocity of all vehicles in the initial scene: v_{scene}^{max}. The target vehicles are simulated using the target vehicle controller as described previously. CAVs and surroundings are simulated in a closed loop with a kinematic bicycle model by applying the planner and target vehicle controller at 20 Hz using the parameters in Table 2.2 for 25 seconds. The prediction horizon and cost function parameters are manually tuned to balance real-time implementation and performance. Note that the target vehicles in the scene have different initial speeds and time-varying accelerations, which renders it a more challenging environment for a multi-CAV platoon to change lanes.

In 200 randomly generated dense traffic scenarios, the baseline and our proposed lane change strategies were applied. The performance metrics are the number of successful lane change completion of all three CAVs and the average lane change competition time in successful lane change simulation runs. Table 2.3 reports the obtained results. Over two-fold increase in number of lane change completions using our proposed strategy is mainly because it only requires a single vehi-

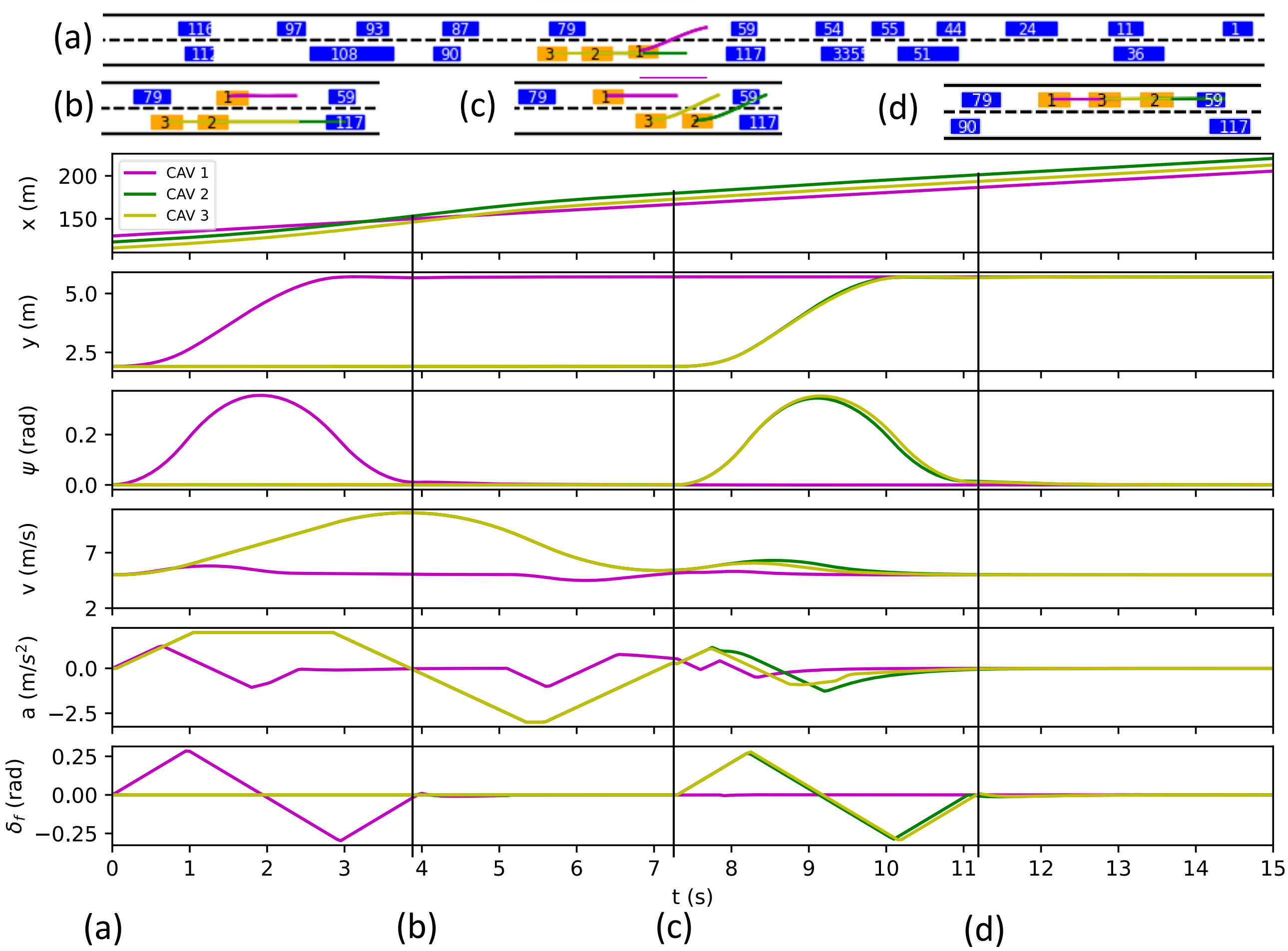


Figure 2.4: **Top:** A multi-CAV platoon changes lanes using the proactive and cooperative lane change strategy in a dense traffic scenario. Note that orange vehicle 1, vehicle 2, and vehicle 3 correspond to CAV 1, CAV 2, and CAV 3, respectively. Scene (a) shows the initial scene of lane 2 and 3 at frame 561 from the NGSIM dataset. Scene (b)-(c) show the facilitator's (CAV 1) changing lanes and regulating the distance while CAV 2 and 3 get in the desired formation. Scene (d) shows the completion of platoon-wise lane change. **Bottom:** The multi-CAV platoon vehicle's states and inputs are shown. The labeled sections on the time axis correspond to the scenes described above.

cle (facilitator) to change lanes to initiate the coordinated maneuvers to enable multi-CAV lane change whereas the baseline controller requires three-vehicles to change lanes simultaneously. After changing lanes, the facilitator regulates the free space by proactively modifying the environment to render the lane change feasible for the other CAVs. Moreover, multi-CAV platoon completed lane change using the proposed strategy in every traffic scenario that the baseline strategy successfully completed lane change. On average, the baseline strategy takes 12.20 seconds to complete a

Table 2.2: Numerical simulation parameters

Vehicle Parameters		**target vehicle Model Parameters**	
Parameter	**Value**	**Parameter**	**Value**
l_f	2.235 m	c	1.0
l_r	2.235 m	s_0	2 m
w	2 m	v_0	v_{scene}^{max}
$v_{\min}$	1.0 m/s	T	1.0 s
$v_{\max}$	32 m/s	a	0.73 m/s^2
$a_{\min}$	-3 m/s^2	b	1.67 m/s^2
$a_{\max}$	2 m/s^2	σ^2	0.2 m/s^2
$\delta_{\min}$	-0.4	**Planner Parameters**	
$\delta_{\max}$	0.4	**Parameter**	**Value**
$\Delta a_{\min}$	-2 m/s^3	N	40
$\Delta a_{\max}$	2 m/s^3	Δt	0.1 s
$\Delta\delta_{\min}$	-0.3 rad/s	d_{safe}	$2 + (l_f + l_r)$ m
$\Delta\delta_{\max}$	0.3 rad/s	n	3
Scene Parameter		c_s	20
Parameter	**Value**	c_y, c_ψ, q_d	3
Lane Width	3.8 m	c_v	2

Table 2.3: Performance comparison of baseline and proposed lane change strategies

	Num. of $LC_{complete}^{P}$	Avg. $LC_{complete}^{P}$ time
Baseline	27	12.20 s
Proposed	57	12.64 s

multi-CAV platoon lane change while the proposed strategy takes 12.64 seconds. Fig. 2.4 shows snapshots of platoon-wise lane change using the proposed strategy in a dense traffic scenario generated from NGSIM dataset as well as time series of states and inputs of the CAVs during the simulation. In this scenario, platoon-wise lane change using the baseline strategy is not feasible during the duration of the simulation.

2.6 Chapter Summary

In this chapter, we introduced a proactive and cooperative multi-CAV platoon lane change strategy that utilizes a facilitator. A distributed coordination method with MPC path-planners with three modes and a higher-level FSM to manage mode transitions were used to implement the proposed lane change strategy. By proactively modifying the environment, the facilitator enhances the feasibility of the lane change objective. Simulations with 200 dense traffic scenarios based on the NGSIM dataset demonstrate over a two-fold increase in lane change completion compared to a *passive and opportunistic* approach. The results substantiate the proposed approach in certain traffic conditions (i.e. dense traffic and small speed difference between OL and TL). Furthermore, the proposed distributed coordination method is scalable to more CAVs without adding constraints to the CFTOP (2.9). As the number of CAVs increases, the implementation of the front/rear vehicle selection algorithm (which is essentially a sorting algorithm) in constraints (2.9h) and (2.9i) has a complexity of $\mathcal{O}(n \log n)$. Future work will consider the vehicle interaction modeling and a facilitator selection algorithm to improve performance.

Chapter 3

Learning Two-agent Motion Planning Strategies for MPC

This chapter is adapted from a previous publication: Hansung Kim et al. "Learning Two-agent Motion Planning Strategies from Generalized Nash Equilibrium for Model Predictive Control". In: *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*. Ed. by Necmiye Ozay et al. Vol. 283. Proceedings of Machine Learning Research. PMLR, Apr. 2025, pp. 112–123. URL: `https://proceedings.mlr.press/v283/kim25a.html`

Chapter 2 addresses multi-agent coupling through explicit coordination via a finite state machine mechanism. This chapter considers settings where no such coordination structure exists and agents must independently reason about their interactions. As outlined in Chapter 1, solving for generalized Nash equilibria online is computationally expensive and scales poorly with horizon length and constraint complexity. We propose Implicit Game-Theoretic Model Predictive Control (IGT-MPC), a decentralized algorithm for two-agent motion planning. We formulate multi-agent interactions as constrained dynamic games and solve for the generalized Nash equilibrium (GNE) offline to generate a dataset of game-theoretic interaction outcomes. A neural network is trained to predict the reward outcome of these interactions and embedded as the terminal cost-to-go in a short-horizon MPC policy, enabling each agent to implicitly account for strategic interactions without solving the game online. By shifting the computational cost of GNE solving offline, the resulting online policy reduces to a short-horizon MPC problem of fixed size, yielding predictable and bounded computation time at each control step. The resulting policy is distributed: each agent independently solves its own fixed-size MPC problem online, with the learned terminal cost implicitly encoding the game-theoretic interactions with other agents. The primary aim of this work is not to solve autonomous racing or driving, but to demonstrate that game-theoretic strategic outcomes can be effectively encoded in neural networks and integrated into numerical optimization schemes for model-based optimal control—paving the way for research at the intersection of game-theoretic reasoning, machine learning, and model-based optimal control.

We demonstrate IGT-MPC for two agents in competitive head-to-head racing and cooperative unsignalized intersection navigation, showing that the learned value function produces emergent behaviors consistent with Nash equilibrium solutions at a fraction of the computational cost of direct game solvers.

3.1 Introduction

Multi-agent motion planning is a fundamental challenge for autonomous systems navigating complex and dynamic environments, especially when an ego agent must adapt in real-time to the actions of other agents. Multi-agent settings are generally categorized as competitive, cooperative, or non-cooperative. In competitive settings, agents pursue conflicting goals, often seeking outcomes that favor themselves at the expense of their opponents. An example, as shown in Fig. 3.1a, is head-to-head car racing, where competing agents attempt to finish a race ahead of their opponent. In cooperative settings, agents collaborate to achieve shared objectives, such as navigating an un-signalized intersection with connected autonomous vehicles (CAVs), which must coordinate to move safely and efficiently without causing delays or collisions, as shown in Fig. 3.1b. Lastly, in non-cooperative settings, agents act independently to maximize their own interests without collaboration. While our approach applies to all three multi-agent environment settings, we will demonstrate our approach on the two examples illustrated in Fig. 3.1. The main source of challenge in multi-agent motion planning is the complex interaction between agents in a shared environment. Robotaxi services, now operational in select U.S. cities, showcase advanced autonomous navigation but occasionally face the “freezing robot” problem—vehicles freezing to avoid collisions—highlighting challenges in real-world interactions between robots [11, 67]. As analyzed by Jankovic [39], the propensity to gridlock is fundamentally tied to the stability properties of the multi-agent system’s equilibria: decentralized algorithms that produce stable equilibria in the interagent dynamics trap agents in deadlock states from which small perturbations cannot escape, while algorithms with unstable equilibria naturally break symmetry and establish a passing order through continuous dynamics without requiring explicit discrete decisions. More broadly, decentralized non-convex algorithms with coupled collision avoidance constraints are prone to reciprocal oscillations caused by the combination of discontinuous control actions and perception delays, while purely decentralized strategies without interaction modeling often lead to gridlock. In contrast, interaction-aware decentralized approaches that account for other agents’ actions through best-response computations or co-optimization effectively avoid these issues.

A common method for interaction-aware motion planning leverages game-theoretic principles to explicitly model multi-agent interactions, accounting for how each agent’s actions influence others. As in [104, 94, 43, 99], the multi-agent motion planning problem is posed as an equilibrium finding problem of a constrained dynamic game, for which the generalized Nash equilibrium (GNE)—a set of strategies where no agent has an incentive to unilaterally deviate from its GNE strategy—is a common solution concept. In [104], a novel numerical solver for computing GNE for open-loop dynamic games with state and input constraints is proposed and the solver is demonstrated in autonomous racing examples. However, this method suffers from computational efficiency which limits its use for real-time planning. Similarly, [94] proposed a game-theoretic formulation for multi-robot racing that approximates the GNE using an iterative best-response method but also faces scalability challenges. The Potential-iLQR approach [43, 10] formulates

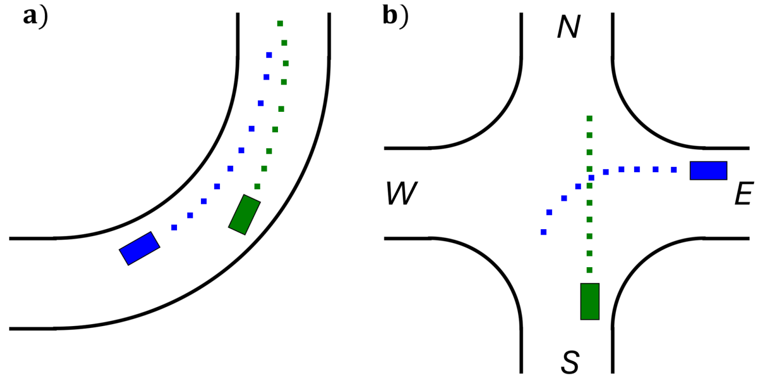


Figure 3.1: Two-agent interaction scenarios: a) competitive head-to-head racing, and b) cooperative un-signalized two-way intersection navigation, with colored squares showing vehicles' planned trajectories.

multi-agent interactions as a potential dynamic game [70] and uses distributed optimization to approximate the GNE in a scalable manner in [99]. However, the potential game formulation cannot typically capture purely zero-sum components in agent objective functions, limiting its applicability in competitive scenarios such as autonomous racing. More critically, the game-theoretic motion planning approaches presented in these works require precise knowledge of other agents' dynamics, costs, and decision-making processes [103], often requiring assumptions about other agents' behavior that may not hold true in real-world scenarios. In practice, such information is rarely available—agents may have unknown or partially observed objectives, heterogeneous dynamics, or strategies that deviate from assumed rationality—making these methods fragile when deployed outside of controlled settings.

Furthermore, a key limitation shared by these game-theoretic motion planning methods is computational cost. Solving for the GNE online at each control step requires iteratively solving coupled optimization problems whose complexity grows with the number of agents, the prediction horizon, and the number of constraints. ALGAMES [56], one of the fastest constrained dynamic game solvers, achieves real-time performance at over 60 Hz in an MPC loop but is demonstrated only for scenarios with up to three vehicles using simplified dynamics. The iLQGames algorithm [99] provides feedback Nash equilibrium strategies through iterative linear-quadratic approximations but cannot natively handle state and input constraints, requiring penalty-based workarounds for collision avoidance that can degrade convergence and solution quality. DG-SQP [104] handles general nonlinear constraints through sequential quadratic programming but its per-iteration cost scales with the joint state and input dimensions of all agents. Even among these more scalable iterative methods, computation times on the order of seconds are common for problems with long horizons.

This limitation motivates our approach to design a feedback policy that uses game-theoretic reasoning as a guiding strategy while adapting to the observed behaviors of other agents by using a learned value function to integrate game-theoretic reasoning into a feedback MPC policy via the terminal cost function.

In this chapter, we present an algorithm for learning two-agent motion planning strategies from a dataset of game-theoretic interactions and embedding it in a real-time model predictive control (MPC) scheme to guide the solution of short-horizon optimal control problems. Our approach comprises three main steps:

1. **Game-theoretic Interaction Dataset Generation:** Generate the dataset offline by randomly sampling joint initial conditions and solving constrained dynamic games over agents and storing the GNE solutions.

2. **Reward Outcome Learning:** Train a neural network value function to predict the reward outcomes associated with these interactions. The reward outcome captures the objective of the multi-agent task in a low-dimension. For example, in racing scenarios, the reward reflects progress advantage along a racetrack relative to the opponent; in cooperative navigation, it represents collective progress toward the agents' target positions.

3. **Implicit Game-Theoretic MPC (IGT-MPC):** Use the learned value function as the terminal cost-to-go function in an MPC policy for online control.

By integrating a learned value function that encodes the strategic outcomes arising from game-theoretic motion planning, IGT-MPC enables implicit interaction-aware motion planning without the need for explicit interaction models or computationally intensive game-theoretic equilibrium finding [104, 94].

Remark 1. *The current IGT-MPC implementation focuses on two-agent motion planning problems, leaving open the challenges of scaling to larger numbers of agents and generalizing to unseen environments such as varying race track layouts or intersection geometries. While this initial effort does not address these challenges, we acknowledge their importance for enabling practical real-world applications. The primary aim of this work is not to provide an immediate solution for autonomous racing or driving but to demonstrate that game-theoretic strategic outcomes can be effectively encoded in neural networks which can then be integrated into numerical optimization schemes for model-based optimal control, paving the way for research in integration of game-theoretic reasoning, machine learning, and model-based optimal control.*

3.2 Problem Formulation

Consider a two-agent autonomous vehicle system where the finite-horizon, discrete-time dynamics are described by

$$z_{t+1} = f(z_t, u_t), \tag{3.1}$$

where $z_t^i \in \mathcal{Z}^i, u_t^i \in \mathcal{U}^i$ are the state and input of agent i at time t and

$$z_t := [z_t^1, z_t^2]^\top \in \mathcal{Z}^1 \times \mathcal{Z}^2 = \mathcal{Z} \subseteq \mathbb{R}^{2n_z}, \tag{3.2}$$

$$u_t := [u_t^1, u_t^2]^\top \in \mathcal{U}^1 \times \mathcal{U}^2 = \mathcal{U} \subseteq \mathbb{R}^{2n_u}, \tag{3.3}$$

are the concatenated states and inputs of all agents.

Agent i is subject to state and input constraints defined by $\mathcal{Z}^i = \{z \in \mathbb{R}^{n_z} \mid \underline{z}^i \leq z \leq \overline{z^i}\}$ and $\mathcal{U}^i = \{u \in \mathbb{R}^{n_u} \mid \underline{u}^i \leq u \leq \overline{u^i}\}$, where $\underline{x}$ and $\overline{x}$ are lower and upper bounds on the respective variables which can be derived from the vehicle's actuation limitations, traffic law, or road boundaries. Also, agent i is subject to actuation rate limits denoted by $\Delta\mathcal{U}^i = \{\Delta u \in \mathbb{R}^{n_u} \mid \underline{\Delta u}^i \leq \Delta u \leq \overline{\Delta u^i}\}$, where $\Delta u_k := [u_k^1 - u_{k-1}^1, u_k^2 - u_{k-1}^2]^\top \in \Delta\mathcal{U}^1 \times \Delta\mathcal{U}^2 = \Delta\mathcal{U} \subseteq \mathbb{R}^{2n_u}$.

Each agent i minimizes its own cost function, which is comprised of stage cost $\ell_k^i(z_k, u_k^i)$ and terminal cost $\ell_{N_G}^i(z_{N_G})$ over a finite horizon length of N_G:

$$J^i(\mathbf{z}, \mathbf{u^i}) = \sum_{k=0}^{N_G-1} \ell_k^i(z_k, u_k^i) + \ell_{N_G}^i(z_{N_G}) \tag{3.4a}$$

$$= J^i(\mathbf{u}^i, \mathbf{u}^{\neg i}, z_0), \tag{3.4b}$$

where the $\mathbf{u^i} = [u_0^i \dots, u_{N_G-1}^i]$ denote the input sequences of agent i, and $\mathbf{z} = [z_0, \dots, z_{N_G}]$ is the sequence of joint states over the prediction horizon N_G. In this work, we use the notation $z_k^{\neg i}$ and $u_k^{\neg i}$ to denote the collection of states and inputs for all but the i-th agent. We derive (3.4b) by recursively substituting the dynamics (3.1) into the cost function (3.4a), which inherently depends on the open-loop input sequences of all agents. The cost is a function of the joint state and input sequence of the i-th agent, capturing dependence on the behavior of other agents. Since each agent's strategy is a time-indexed input sequence conditioned on the initial state z_0, rather than a state-feedback policy, the resulting equilibrium concept is an open-loop generalized Nash equilibrium. The agents are also subject to additional n_c constraints, $C(\mathbf{u}^1, \mathbf{u}^2, z_0) \leq 0$, which describe the coupling between agents, with its dependence on joint dynamics remaining implicit.

3.3 Learning Game-theoretic Motion Planning Strategy from Generalized Nash Equilibrium

In this section, we describe our approach to learning game-theoretic motion planning strategies from GNE and how to embed it in an MPC policy for motion planning in competitive and cooperative two-autonomous vehicle systems: 1) two-vehicle head-to-head racing on an L-shaped track. 2) two-vehicle navigating through an un-signalized four-way intersection.

Game-theoretic Interaction Data Generation

First, we define the constrained dynamic game as the tuple:

$$\Gamma = (N_G, \mathcal{Z}, \mathcal{U}, \Delta\mathcal{U}, f, \{J^i\}_{i=1}^2, C). \tag{3.5}$$

For such a game, a GNE [27] is attained when all agents have no incentive to unilaterally change their strategies along feasible directions of the solution space, which means $\mathbf{u}^\star = [\mathbf{u}^{1,\star}, \mathbf{u}^{2,\star}]$ minimizes (3.4) for both agent 1 and 2. There exists a vast literature on formulating constrained dynamic games for different robotic multi-agent motion planning problems and solving for GNE, such as ALGAMES, iterative Linear Quadratic Regulator (iLQR), Potential-iLQR, and DG-SQP [56, 31, 99, 104]. Without loss of generality, we use the DG-SQP solver [104] to compute a GNE for constrained dynamic games in this work.

We randomly sample joint initial conditions z_0 while other components of the game Γ are fixed and solve for the GNE from that initial condition. Then, the GNE is used to compute the reward outcome of game-theoretic interactions. The reward $R^i \in \mathbb{R}$ quantifies the success in achieving the primary objective of the multi-agent interactive motion planning problem from the perspective of Agent i. While J^i in (3.4) describes Agent i's cost function and includes stage costs ℓ^i_k, R^i is derived from the terminal costs $\ell^i_{N_G}$, $i = 1, 2$, and thus directly reflects the game outcome.

Two-vehicle head-to-head racing

Here the objective is to stay ahead of the opponent. We use the reward

$$R^i = s^i_{N_G} - s^{\neg i}_{N_G}, \quad i = 1, 2, \tag{3.6}$$

that represents the advantage of vehicle i over vehicle $\neg i$, where $s^i_{N_G}$ and $s^{\neg i}_{N_G}$, are the longitudinal progress of Agent i and Agent $\neg i$ on the race track at time step N_G, respectively. When $i = 1$, we are considering Agent 1 as the ego agent, which means we are predicting the progress advantage that Agent 1 will have over Agent 2. We also construct the reward from the perspective of Agent 2 as the ego agent when $i = 2$. Thus, from a single GNE solution, we extract two data points from the perspective of each agent. Each reward target R is associated with an input feature vector which is the joint initial condition augmented with additional information about the interaction. We note that the input vectors are also problem-dependent, and in the two-vehicle head-to-head racing example, we define

$$\tilde{x}^i_0 = [z^{\neg i}_0, z^i_0 - z^{\neg i}_0]^\top \in \mathbb{R}^{n_x} \quad i = 1, 2. \tag{3.7}$$

Here, $z^i_0 = [v^i_0, e^i_{\psi,0}, s^i_0, e^i_{y,0}]$, where v^i_0 denotes the initial longitudinal velocity, $e^i_{\psi,0}$ is the heading angle error, s^i_0 represents the progress along the race line, and $e^i_{y,0}$ is the lateral error of agent i.

Collaborative navigation through an un-signalized intersection

The main objective of the agents is to collaborate with other agents to ensure all agents pass the intersection safely and promptly. In this example, we propose the reward

$$R^i = \sum\nolimits_{i=1}^{2} s^i_{N_G} \quad i = 1, 2, \tag{3.8}$$

which represents the collective progress along corresponding routes at the end of the game horizon N_G. The associated input feature to R^i is

$$\tilde{x}^i_0 = [\tilde{z}^{\neg i}_0, \tilde{z}^i_0 - \tilde{z}^{\neg i}_0]^\top \in \mathbb{R}^{n_x} \quad i = 1, 2, \tag{3.9}$$

where $\tilde{z}_0^i = [s_0^i, v_0^i, sc_i]$. Assuming all agents do not deviate from the centerline of the road, we only include the longitudinal states and scenario encoding sc_i, which is a categorical variable, of the two-vehicle intersection to distinguish what type of interaction is taking place in the perspective of agent i. Without it, data points from different interactions (i.e. left turn & left turn vs. left turn & straight) become indistinguishable. For more details about scenario encoding and its implementation, please refer to Sec. 3.7. Similarly, we construct the features and reward targets from the perspective of both agents. Note that the reward target for each agent is the same, which reflects the cooperative setting where all agents share the reward. The computed reward $R = \{R^i\}_{i=1}^2$ and feature vector $\tilde{x}_0 = \{\tilde{x}_0^i\}_{i=1}^2$ are added to the dataset as follows

$$\begin{aligned} X &= X \cup \{\tilde{x}_0\} \\ Y &= Y \cup \{R\}, \end{aligned} \tag{3.10}$$

Remark 2. *Capturing interaction between agents in the generated dataset is key for a successful implementation of the proposed method. In practice, the distribution of training data often differs from the distribution seen during IGT-MPC deployment, a phenomenon known as distribution shift. As covering the entire feature space by exhaustively sampling data points is intractable, a more data-efficient approach leverages strategic initial condition sampling based on any prior knowledge of the deployment data distribution. For example, in a head-to-head racing scenario, when two agents are far enough apart, then there isn't any significant coupling between their decision-making. Thus, sampling from the set of initial conditions where an interaction between two agents is possible will be sufficient in capturing the interactive behavior.*

Remark 3. *When sampling initial conditions, the game solver may occasionally fail to compute the GNE due to infeasibility from collision avoidance constraints under the joint dynamics. Storing these infeasible points in the dataset and associating them with low reward outcomes is crucial, as they represent undesirable states leading to collisions that the model must learn to avoid.*

Learning the Reward Outcome

After generating the dataset $\mathcal{D} = (X, Y)$, we obtain the game-theoretic value function $V_{GT} : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ that estimates the reward outcome R^i of the game-theoretic interactions between agents at time step N_G from the input features $\tilde{x}_0^i$. We use the following simple neural network architecture with $tanh$ activation function:

$$y_l = tanh(W_l y_{l-1} + b_l), \quad l = 1, ..., L, \tag{3.11}$$

where W_l, b_l are the parameters of the l-th layer of the network with hidden state dimension h. Additionally, $y_0 = \tilde{x}^i$ and $y_L = \hat{R}^i$, and $tanh$ is a non-linear, differentiable activation function ideal for expressing negative-valued outputs.

The regression model is trained on $\mathcal{D}$ using the mean-squared error loss function defined as $\mathcal{L} = \frac{1}{n_b} \sum_{k=1}^{n_b} \sum_{i=1}^{2} (R_k^i - \hat{R^i}_k)^2$ where n_b is the training batch size. We employ the Adam optimizer [52] with early stopping to prevent overfitting. Also, the features and target values in $\mathcal{D}$ are normalized during training. From this point forward, the trained neural network value function on a game-theoretic interaction dataset is referred to as V_{GT}. Notably, this relatively simple neural network architecture effectively captures game-theoretic strategies in the GNE for two-agent motion planning in both competitive and cooperative settings, as demonstrated in Sec. 3.4. We note however, that our current approach focuses on two-agent systems and is limited in generalizability to unseen environments (e.g., different racetrack or intersection geometries). To enhance scalability w.r.t. the number of agents and generalizability, the interactions among all agents and environmental information must be incorporated into the feature vector, which may require more complex neural network architectures—a direction we plan to explore in future work.

Implicit Game-theoretic MPC Policy

After training V_{GT} that predicts the game-theoretic reward outcome N_G steps into the future given the input features, we use this predictor as a terminal cost-to-go function for a real-time MPC policy that implicitly emulates the game-theoretic strategies encoded in GNE.

Collision Avoidance

While methods like optimization-based collision avoidance constraints [102], which model agents as polytopes, can be used, we opt to represent obstacles as circles and impose distance constraints, maintaining simplicity without compromising effectiveness. Inter-agent collision avoidance constraints are given as $\mathbb{CA} = \{(p_1, p_2) \mid \|p_1 - p_2\|_2 \geq d_{min}\}$, where p_1 and p_2 are global Cartesian position vectors, and d_{min} is the user-defined parameter for minimum allowable distance between any two agents. We denote $P(z^i, z^j)$ as the projection of the state vector pair to Cartesian position vector space. Two agents i and j are considered to be collision-free if and only if $P(z^i, z^j) \in \mathbb{CA}$.

MPC Formulation

The MPC policy is a *predict-then-plan* scheme that computes an optimal ego trajectory w.r.t. the open-loop predictions of the environment. We assume the forecasts are provided by a trajectory prediction scheme. In the numerical examples, we generate $\hat{\mathbf{z}}^{\neg i}$ using an oracle that provides the optimal planned trajectories of all agents. For competitive racing scenarios, we corrupt it with Gaussian noise [103].

Remark 4. *While the numerical examples in Sec. 3.4 use oracle predictions for clarity of exposition, the IGT-MPC policy is not dependent on the quality of the prediction oracle. In the cooperative intersection navigation example, replacing the oracle with a constant-acceleration prediction model yields the same qualitative strategic behaviors (e.g., yield ordering and gridlock avoidance). This is because the game-theoretic strategy is encoded in V_{GT} at the terminal state, not in the accuracy of the intermediate trajectory forecasts—V_{GT} guides the MPC toward terminal states that are strategically favorable regardless of the prediction model used for collision avoidance along the horizon.*

Each agent i computes a state-feedback control $u_t^i = \pi_{\text{MPC}}(z_t^i, \hat{\mathbf{z}}^{\neg i}, V)$ by solving the following finite-horizon optimal control problem (FHOCP) towards tracking a state reference $\{z_{r,k}\}_{k=t}^{t+N}$ given forecasts of other agents $\hat{\mathbf{z}}^{\neg i} = \{\hat{z}_{k|t}^{\neg i}\}_{k=t}^{t+N}$ over the N step MPC prediction horizon,

$$\min_{\boldsymbol{u}^i} \sum_{k=t}^{t+N-1} \|z_{k|t}^i - z_{r,k}\|_Q^2 + \|u_{k|t}^i\|_{R_1}^2 + \|\Delta u_{k|t}^i\|_{R_2}^2 + V(\tilde{x}_{t+N}^i) \tag{3.12a}$$

$$\text{s.t.} \quad z_{k+1|t}^i = f^i(z_{k|t}^i, u_{k|t}^i), \quad \forall k \in \mathbb{I}_t^{t+N-1} \tag{3.12b}$$

$$z_{k|t}^i \in \mathcal{Z}^i, \quad \forall k \in \mathbb{I}_t^{t+N} \tag{3.12c}$$

$$u_{k|t}^i \in \mathcal{U}^i, \quad \forall k \in \mathbb{I}_t^{t+N-1} \tag{3.12d}$$

$$\Delta u_{k|t}^i \in \Delta\mathcal{U}^i, \quad \forall k \in \mathbb{I}_t^{t+N-1} \tag{3.12e}$$

$$P(z_{k|t}^i, \hat{z}_{k|t}^j) \in \mathbb{CA}, \quad \forall j \in \{\neg i\} \tag{3.12f}$$

$$z_{t|t}^i = z^i(t), \tag{3.12g}$$

where $z^i(t)$ is the measured state of the i-th agent at time t, and $z_{k|t}^i$ indicates the state z^i at k-th step predicted at time t. f^i is the discrete-time dynamics of agent i, and $\mathbb{I}_{k_1}^{k_2}$ denote the index set $\{k_1, k_1+1, \ldots, k_2\}$. z_r is a reference state trajectory and $Q \succeq 0$ and $R_1, R_2 \succ 0$ are weighting matrices for the state reference tracking, input, and input rate costs respectively. The reference state trajectory is the pre-computed raceline or the centerline of a road.

The cost function (3.12a) includes a stage cost term $\ell_k^i(\cdot)$ and a terminal cost term $V(\cdot)$. It is important to note that the input to the terminal cost function $\tilde{x}_{t+N}^i$ includes the joint state $z_{t+N|t} = [z_{t+N|t}^i, \hat{z}_{t+N|t}^{\neg i}]$ where $z_{t+N|t}^i$ is a decision variable of the optimization problem. Therefore, if $V = -V_{GT}$, minimizing the learned value function over $z_{t+N|t}^i$ can be interpreted as agent i selecting its terminal state given the forecasts of other agents, $\hat{z}_{t+N|t}^{\neg i}$, to maximize its reward (i.e. maximize advantage over the opponent). The learned value function, V_{GT}, is used in closed-loop with the MPC policy as formalized in Algorithm 1. The FHOCP (3.12) is constructed using CasADi [2] where V_{GT} is constructed symbolically using the same architecture and trained weights from the learned model in PyTorch [78].

3.4 Numerical Examples

We demonstrate Algorithm 1 in two-vehicle competitive and cooperative multi-agent settings. The code for obtaining the presented results can be found in the GitHub repository [45]. We use $N = 10$ and a sampling time of 0.1 s for all MPC policies in our numerical examples.

Two Vehicle Head-to-Head Racing

In the two-vehicle head-to-head race example, we consider the Frenet-frame dynamic bicycle model given a precomputed race line trajectory for an L-shaped race track [104]. We formulate the autonomous racing problem as a constrained dynamic game and use DG-SQP solver to compute the GNE for such game. For an

Algorithm 1 Implicit Game-Theoretic MPC (IGT-MPC)

Require: V_{GT}

Set the task horizon T

$t \leftarrow 0$

while $t < T$ **do**

$\hat{\mathbf{z}}^{\neg i} \leftarrow$ Forecasts of other agents

$\tilde{x}_{t+N} \leftarrow \hat{\mathbf{z}}^{\neg i}$

$V \leftarrow V_{GT}$

$u_t^i \leftarrow \pi_{\text{MPC}}(z^i(t), \hat{\mathbf{z}}^{\neg i}, V)$ ▷ See Eq. (3.12)

Apply u_t^i to system

$t \leftarrow t + 1$

in-depth explanation of formulating the game and solving for GNE, refer to [103, Chapter 7]. We collected approximately 1800 samples with $N_G = 25$, and trained V_{GT} with $h = 48, L = 2$ using PyTorch [78].

The MPC policy (3.12) is additionally subject to performance-limiting factors such as friction limits from nonlinear tire models, as follows

$$a^f(z^i_{k|t}, u^i_{k|t}), a^r(z^i_{k|t}, u^i_{k|t}) \leq \mu g, \tag{3.13}$$

where $a_f(\cdot, \cdot)$ and $a_r(\cdot, \cdot)$ denote the front and rear tire lateral acceleration, respectively. μ is the coefficient of friction of the race track, and g is the gravitational acceleration.

To study the impact of the V_{GT}, we consider two different terminal cost-to-go functions,

$$V = \begin{cases} -V_{GT}(z^i, \hat{z}^{\neg i}), \\ V_{MP}(z^i, \hat{z}^{\neg i}) = -(s^i - \hat{s}^{\neg i}), \quad i = 1, 2, \end{cases} \tag{3.14}$$

where V_{MP} is a naive maximum progress function based on the opponent's prediction to encourage progress maximization with respect to the opponent but does not have any game-theoretic component.

Numerical Results

Fig. 3.2 shows snapshots at identical time instances of two simulated races starting from the same initial condition but different configurations to demonstrate a position defense capability of a slower vehicle when guided by V_{GT}. Race configurations are categorized by two settings: *fast* or *slow* and *GT* or *MP* value functions. The *GT* or *MP* setting denotes the terminal cost-to-go function used by the MPC policy controlling the vehicle. While both vehicles track pre-computed racelines of the same shape, we force the initially leading green vehicle to be slower than the initially trailing blue vehicle by reducing its velocity profile by 10%. For details on setting up the race, refer to [103, Chapter 7]. The level curves of V_{MP} (Top) and V_{GT} (Bottom) for the green vehicle are projected onto the track in Fig. 3.2. While the level curves of V_{MP} are constant over lateral positions, indicating no preference for where on the track the vehicle should end up at the end of the horizon, V_{GT} exhibits more intricate level curves, showing a strategic preference for lateral positions near the pre-computed race line, although the learned value functions had no prior knowledge of

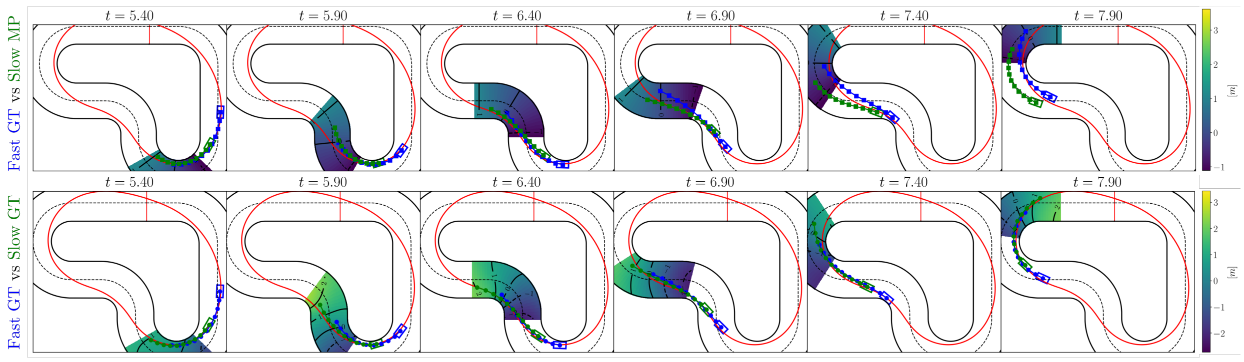


Figure 3.2: In a simulation experiment, the slower (green) vehicle is unable to defend its position against the faster (blue) vehicle with V_{MP} (Top). The slower vehicle with V_{GT} successfully defends its position against the faster car with V_{GT} (Bottom). The heatmap represents the level curves to visualize the value functions used in this experiment. The red curve is the raceline and the colored squares and circles are the planned trajectories for vehicles with corresponding colors.

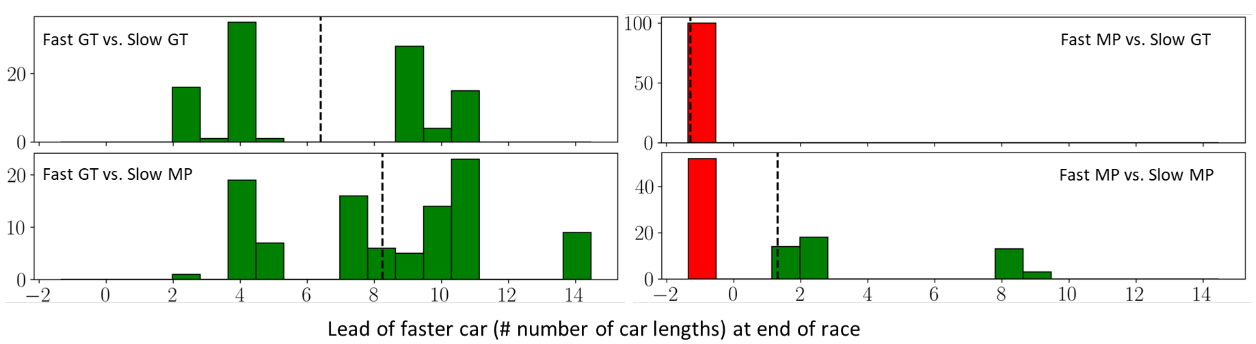


Figure 3.3: Histograms of the lead (in number of car lengths) of the faster vehicle over the slower vehicle at the end of each simulated race over 100 different initial conditions. Green bars indicate bins where the faster vehicle won the race, while red bars indicate losses. The black dashed lines represent average lead values over all simulation runs. Note that vertical scales vary between histograms.

it [103]. By $t = 7.90$ s, the green vehicle guided by V_{MP} is overtaken by the faster opponent, while under V_{GT}, it successfully defends its leading position. More simulation results can be found in the video [46].

Now, we report the Monte Carlo simulation results for various race configurations over 100 different initial conditions. The histograms display the lead of the faster car, measured in car lengths, at the end of each race across 100 different initial conditions for various race configurations in Fig. 3.3. The black dashed lines indicate the mean values. Comparing the histograms in the top row, the *Fast GT* vehicle consistently wins averaging approximately 6.5 car lengths of a lead against *Slow GT* vehicle, while the *Fast MP* vehicle trails the *Slow GT* by an average of 1.4 car lengths. Notably, *Fast MP* vehicle loses 100% of the race against a slower, but more strategic vehicle guided by V_{GT}. In the bottom row, a faster vehicle races against a *Slow MP* vehicle. When guided by V_{GT}, the faster vehicle is able to win 100% of the races while averaging 8.1 car lengths of a lead. However, with the V_{MP}, the win rate is decreased to about 50%. Overall, V_{GT} outperforms the vehicle using V_{MP} in both overtaking and defending, as highlighted in the *Fast GT vs Slow MP* and defending in *Fast MP vs Slow GT* cases.

Two Vehicle Intersection Navigation

In the two-vehicle un-signalized intersection example, we consider the Frenet-frame kinematic bicycle model [47] augmented with global Cartesian coordinates (x, y) for imposing the collision avoidance constraints. For simplicity, we consider homogeneous agents under identical constraint sets in this example. We collected approximately 2000 samples per scenario with $N_G = 200$ and sampling time of 0.1 s, and trained V_{GT} with $h = 128, L = 2$ using PyTorch. The implementation details for obtaining the GNE solutions and constructing the feature vectors are detailed in Sec. 3.7.

Again, to study the impact of the V_{GT}, we consider two different terminal cost-to-go functions,

$$V = \begin{cases} -V_{GT}(\tilde{z}^i, \hat{\tilde{z}}^{\neg i}), \\ V_{MP}(z^i, \hat{z}^{\neg i}) = -(s^i + \hat{s}^{\neg i}), \quad i = 1, 2, \end{cases} \tag{3.15}$$

where V_{MP} is a naive maximum progress function that maximizes the progress of agent i without considering the interaction with other agents. The optimization problem (3.12) is formulated in CasADi and solved using IPOPT. In the cooperative two-vehicle un-signalized intersection example, we assume the vehicles are connected and communicate their route information and planned optimal trajectories, $\hat{\mathbf{z}}^i = \{z^{i,\star}_{k|t}\}^{t+N}_{k=t}$ without delay, allowing $\tilde{x}^i_{t+N}$ to be computed exactly at time t.

Numerical Results

We highlight the differences in how V_{GT} and V_{MP} guide the short-horizon MPC solutions through an intersection from the same initial condition. In Fig. 3.4, we show Vehicles 1 (green) on route *SN* and Vehicle 2 (blue) on *ES* navigating through an un-signalized intersection starting from rest, where both vehicles are controlled using identical MPC policies (3.12) with the same V_{GT}. However, the input features to V_{GT} are constructed from each vehicle's perspective by concatenating their respective features, as specified in (3.9). Although both vehicles share the same policy and value function, their unique perspectives lead to different inputs and outputs. This enables each to follow the collaborative strategy while exhibiting distinct behaviors tailored to their individual contexts. For instance, at $t = 4$ s, as both vehicles approach the intersection, the level curves of V_{GT} for Vehicle 1 show a preference for the terminal states in $s^1_{t+N} \in [10,\ 13]$ m and $v^1_{t+N} \in [2,\ 3]$ m/s, whereas Vehicle 2's level curves indicate a preference for terminal states in $s^2_{t+N} \in [17,\ 20]$ m

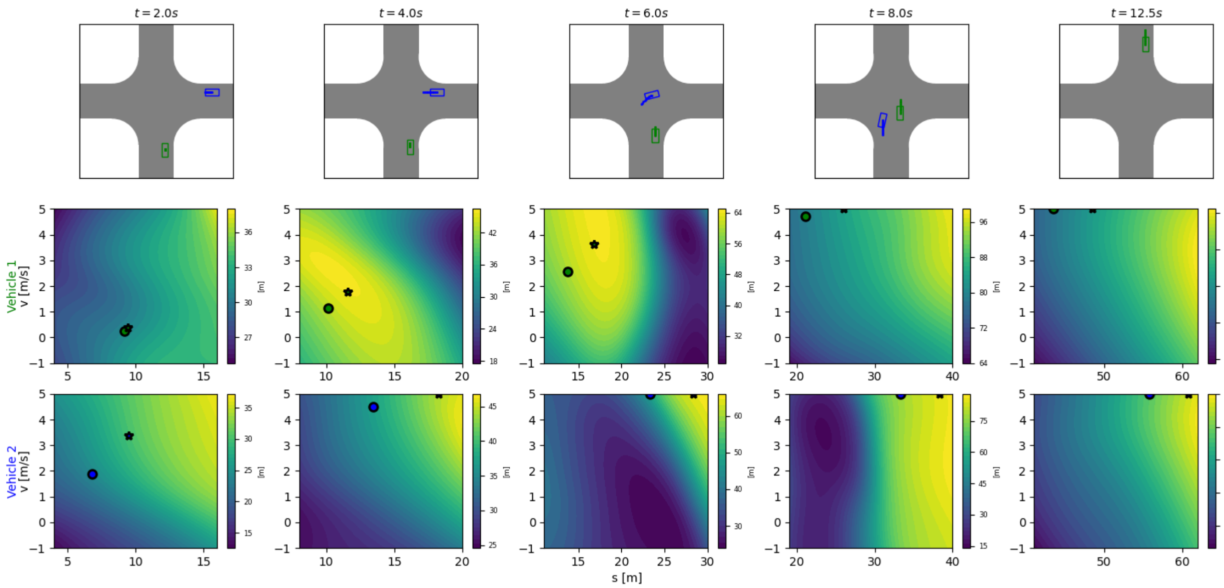


Figure 3.4: In a simulation experiment, Vehicle 1 (green) on route *SN* and Vehicle 2 (blue) on route *ES* navigate through an intersection using V_{GT} (Top). Colored squares represent each vehicle's planned trajectories. The middle and bottom rows display the contour of V_{GT} as perceived by Vehicle 1 and 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t + N$.

and $v^2_{t+N} \in [4, 5]$ m/s. This illustrates how V_{GT} is guiding the MPC planner for Vehicle 1 to accelerate slowly and yield, while V_{GT} is guiding the Vehicle 2 to accelerate quickly and pass the intersection first. At $t = 6$ s, when Vehicle 2 reaches the middle of the intersection at its maximum speed of 5 m/s, the level curves of V_{GT} for Vehicle 1 show a preference for higher speed and progress, guiding Vehicle 1 to start accelerating. After Vehicle 2 completes its left turn and is no longer in the path of Vehicle 1, V_{GT} guides Vehicle 1 to accelerate to maximum speed, allowing both vehicles to pass the intersection by $t = 12.5$ s.

In Fig. 3.5, when both MPC planners are guided by V_{MP}, they prioritize maximizing progress without accounting for their actions' impact on other agents, as shown by the linear level sets in the middle and bottom rows. This leads to planner infeasibility due to unavoidable collisions from inertia, triggering the safety controller to apply maximum braking at $t = 6$ s. As a result, the vehicles become stationary and gridlocked, unable to proceed due to coupled collision avoidance constraints. While extending the prediction horizon could mitigate this issue, it increases computational complexity. Even so, the naive *predict-then-plan* approach can still encounter infeasibility and gridlock in this intersection scenario, as demonstrated in the video [46]. Furthermore, simulation experiment results demonstrating the effectiveness of V_{GT} in emulating coordinated behaviors in different scenarios are reported in Sec. 3.7.

For each scenario $m \in \mathbb{I}_1^8$ as defined in Table 3.2 in Sec. 3.7, route combinations are sampled from each $\mathcal{S}_m$ to obtain the corresponding precomputed reference trajectory, z_r, for the route. Initial $s_0^1, s_0^1 \sim U(0, s_{int}/2)$ are sampled, and $v_0^1, v_0^2 = 0\ m/s$ is set to ensure vehicle interaction, where s_{int} denotes the longitudinal displacement at the intersection. Closed-loop simulations are then evaluated with both vehicles

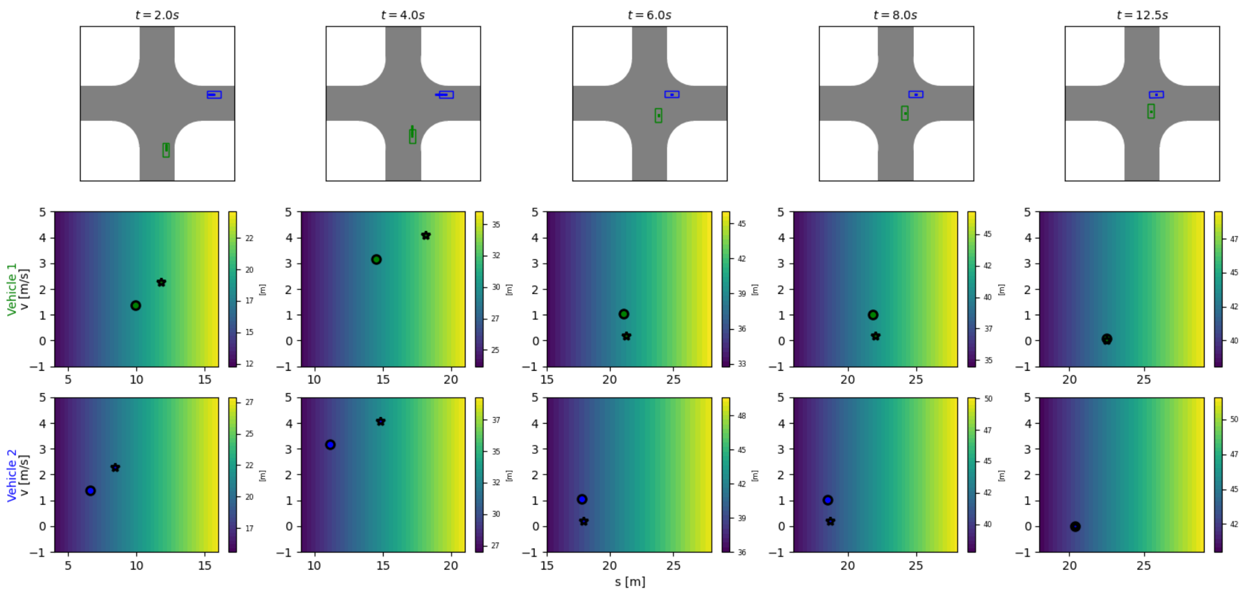


Figure 3.5: In a simulation experiment, Vehicle 1 (green) on route *SN* and Vehicle 2 (blue) on route *ES* navigate through an intersection using V_{MP} (Top), reaching a gridlock. Colored squares represent each vehicle's planned trajectories. The middle and bottom rows display the contour of V_{MP} as perceived by Vehicle 1 and 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t+N$. The initial conditions are identical to those in Fig. 3.4

using identical terminal cost-to-go functions in (3.12) over a task horizon of $T = 150$. The value functions, V_{MP} and V_{GT}, are evaluated over 100 initial conditions for each scenario. Performance metrics include % of feasibility, % of gridlock (indicating instances where both vehicles are in a stalemate and cannot proceed their routes) and the average solve time of (3.12). The feasibility percentage measures the proportion of simulation runs where the MPC planners remain feasible throughout the simulation run. A high feasibility percentage signifies that the terminal cost-to-go function effectively guided the vehicles to navigate collaboratively, avoiding the need for safety controller intervention. Conversely, a lack of clear passing order or collaborative navigation strategy can cause vehicles to enter the intersection simultaneously, resulting in gridlock. The numerical simulation results in Table 3.1 highlight the effectiveness of the proposed approach. Guided by V_{GT}, MPC planners achieve at least 97% feasibility across all scenarios, consistently outperforming V_{MP}. Notably, in scenarios 6 and 7, where the naive maximum-progress terminal cost V_{MP} fails completely, V_{GT} achieves 99% feasibility with no gridlocks. Vehicles guided by V_{GT} avoid gridlock by adhering to the passing order encoded within the value function, enabling smooth collaborative navigation. In contrast, V_{MP} often results in gridlocks due to its lack of interaction modeling. While V_{GT} introduces higher computation time, alternative NLP solvers could improve efficiency.

Fig. 3.6 illustrates that deadlock can be avoided by sufficiently extending the prediction horizon to encompass the conflict zone. In this scenario, horizons up to $N = 40$ fail to include the intersection zone, causing the planner to commit to conflicting trajectories and reach a deadlock. Beyond this threshold, the planner can anticipate the conflict and resolve it; the $N = 100$ solution shown here successfully navigates the

Table 3.1: Metrics for Two-Vehicle Intersection Navigation Monte Carlo Simulations

Scenario	**Feasibility** (%)		**Gridlock** (%)		**Avg. Solve Time** (s)	
	V_{MP}	V_{GT}	V_{MP}	V_{GT}	V_{MP}	V_{GT}
1	25	**99**	6	**0**	**0.045**	0.061
2	1	**97**	80	**0**	**0.055**	0.089
3	37	**100**	22	**0**	**0.073**	0.10
4	95	**100**	**0**	**0**	**0.087**	0.091
5	93	**99**	1	**0**	**0.075**	0.078
6	0	**99**	43	**0**	**0.082**	0.10
7	0	**99**	100	**0**	**0.077**	0.098
8	1	**98**	57	**0**	**0.083**	0.084

intersection without gridlock. However, this comes at a significant computational cost, as the optimization problem scales with the horizon length. Furthermore, while the $N = 100$ solution is feasible, it is not game-theoretically optimal: Vehicle 1 is assigned the right of way and passes first, which differs from the passing order of the potential game solution and is suboptimal for maximizing collective progress along each vehicle's respective route.

Fig. 3.7 demonstrates that IGT-MPC with $N = 10$ closely recovers the potential game solution despite its short prediction horizon. By encoding long-horizon interaction outcomes into the learned terminal cost, IGT-MPC captures cooperative dynamics that extend well beyond its 10-step horizon, avoiding the myopic behavior exhibited by standard MPC with the same horizon length. Both the position trajectories and the passing order—Vehicle 2 yielding to Vehicle 1—are consistent with the game-theoretically optimal solution. While transient deviations in the velocity profiles are present, IGT-MPC reproduces the cooperative yielding behavior present in the potential game, achieving a solution that is both computationally tractable and game-theoretically consistent. Notably, the potential game solution requires 14.4 seconds to solve, making it unsuitable for real-time deployment, whereas IGT-MPC recovers comparable behavior at a fraction of the computational cost. When both vehicles use identical MPC policies with the naive maximum-progress terminal cost V_{MP}, symmetric initial conditions produce symmetric behavior, as both vehicles accelerate toward the intersection simultaneously without accounting for their impact on one another. This symmetry leads to planner infeasibility and gridlock. In contrast, V_{GT} implicitly breaks this symmetry by encoding the cooperative passing order learned from GNE solutions. Although both vehicles share the same policy and value function, the input features are constructed from each vehicle's perspective, producing distinct terminal cost landscapes that guide one vehicle to yield and the other to proceed. This enables coordinated behavior to emerge from identical, decentralized policies without explicit communication of intent.

Fig. 3.8 shows the same SN–ES intersection scenario as Fig. 3.4, with identical initial conditions, but with the oracle prediction replaced by a constant-acceleration forecast model for the other agent. The resulting closed-loop behavior is nearly identical to the oracle case: Vehicle 2 (blue) is assigned the right of way by V_{GT} and passes through the intersection first, while Vehicle 1 (green) yields by maintaining low velocity until Vehicle 2 clears the conflict zone, then accelerates through. The V_{GT} contours in the middle and bottom rows exhibit the same topological structure as in Fig. 3.4: at $t = 4$ s, Vehicle 1's value landscape favors low-speed terminal states ($s \approx 8$ m, $v \approx 1.5$ m/s) while Vehicle 2's landscape favors high-speed states ($s \approx 10$ m, $v \approx 5$ m/s), encoding the yield-then-go strategy. By $t = 8$ s, both vehicles have resolved the interaction with the same passing order, and their value landscapes shift to favor maximum progress. Comparing Figs. 3.4 and 3.8, the yield ordering, interaction resolution timing, and value function gradient

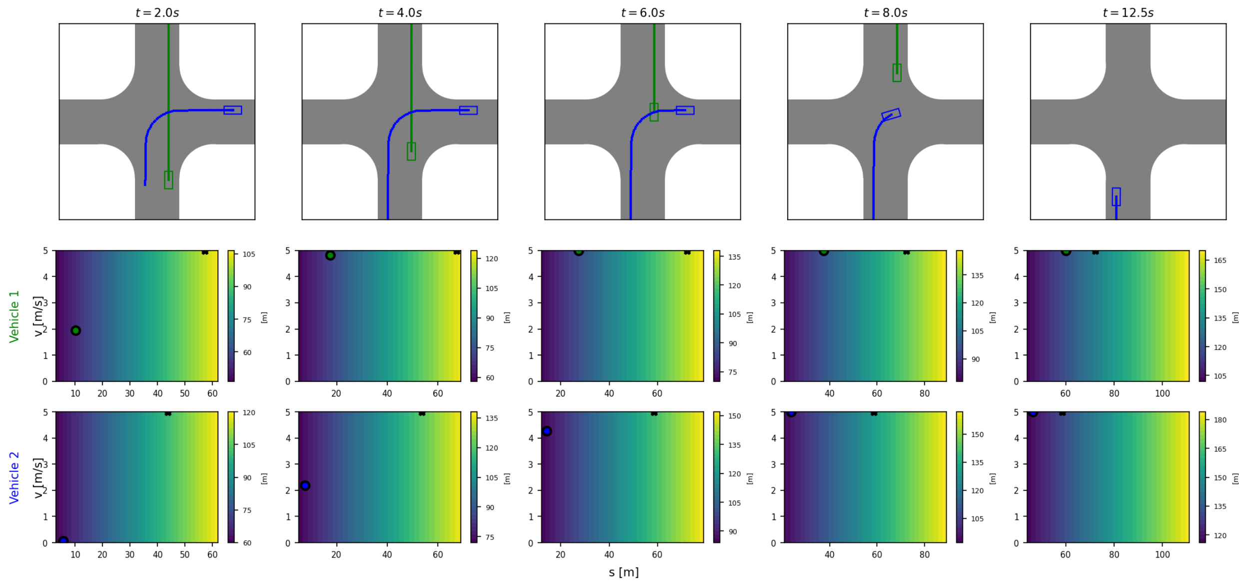


Figure 3.6: $N = 100$ MPC solution: Vehicle 1 (green) on route *SN* and Vehicle 2 (blue) on route *ES* navigate through the intersection using a long-horizon motion planner (top row). While the planner resolves the interaction without gridlock, Vehicle 1 is assigned the right of way and passes first, yielding a different passing order from the IGT-MPC solution. The middle and bottom rows display the contour of V_{MP} as perceived by Vehicle 1 and Vehicle 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t + N$.

structure are preserved despite the degraded prediction model. This corroborates the observation made in Remark 4: because the strategic behavior is encoded in V_{GT} at the terminal state rather than derived from the accuracy of the intermediate forecasts, IGT-MPC is robust to the choice of prediction model. The forecasts along the horizon need only be accurate enough to keep the collision avoidance constraints meaningful over the short N-step window; the game-theoretic passing order emerges from V_{GT} independently of the prediction source.

3.5 Beyond Two Agents

To evaluate the generalization of IGT-MPC beyond the two-agent setting for which it was trained, we apply it to a four-vehicle intersection scenario. Figure 3.9 shows snapshots of the resulting trajectories alongside a Potential Game GNE solution from identical initial conditions. IGT-MPC reproduces the same qualitative passing order as the Potential Game solution: the vehicles negotiate the intersection without collision and in the same sequential order across both methods. However, IGT-MPC exhibits a slight lag in progression through the intersection compared to the Potential Game GNE solution. This is expected and instructive: the Potential Game formulation explicitly coordinates all agents jointly at each step, whereas in IGT-MPC each

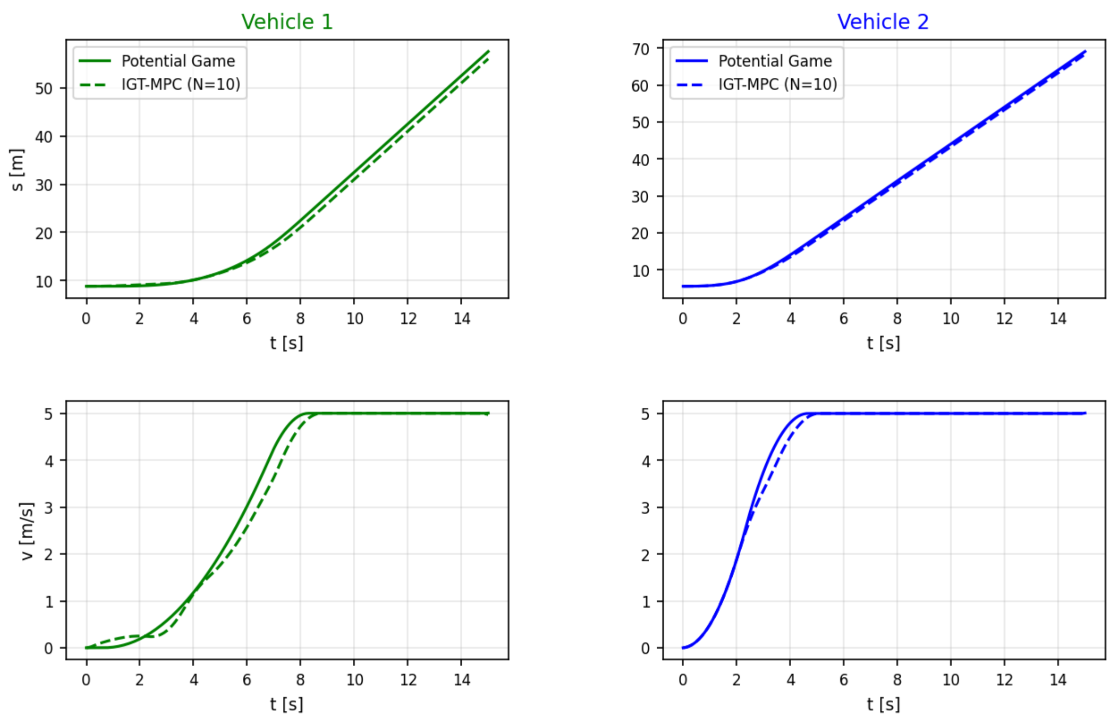


Figure 3.7: Comparison of position s and velocity v trajectories for Vehicle 1 (green) and Vehicle 2 (blue) between the potential game solution and IGT-MPC ($N = 10$) for the intersection scenario of Fig. 3.4. IGT-MPC closely tracks the potential game ground truth for both vehicles, with minor deviations in the transient velocity profiles.

agent acts independently, relying solely on the learned terminal cost to implicitly anticipate the behavior of others leading to suboptimal coordination. The emergent coordination is therefore a byproduct of each agent's individually rational behavior rather than explicit joint optimization. Notably, a naive MPC policy with the same horizon length of $N = 10$ fails to navigate the intersection successfully in this scenario, resulting in deadlock. The fact that IGT-MPC succeeds with the same horizon underscores that it is the game-theoretic terminal cost, rather than the planning horizon, that endows the policy with the foresight necessary to resolve multi-agent conflicts. We note that as the number of agents increases, the cost of generating GNE training data grows significantly, as equilibrium solve time scales poorly with the number of players. Nevertheless, this experiment demonstrates that the concept of embedding a learned GNE proxy as a terminal cost in MPC remains effective at four agents, suggesting that the offline data generation cost, while increasing, is an offline computational expense that does not affect the complexity of online execution.

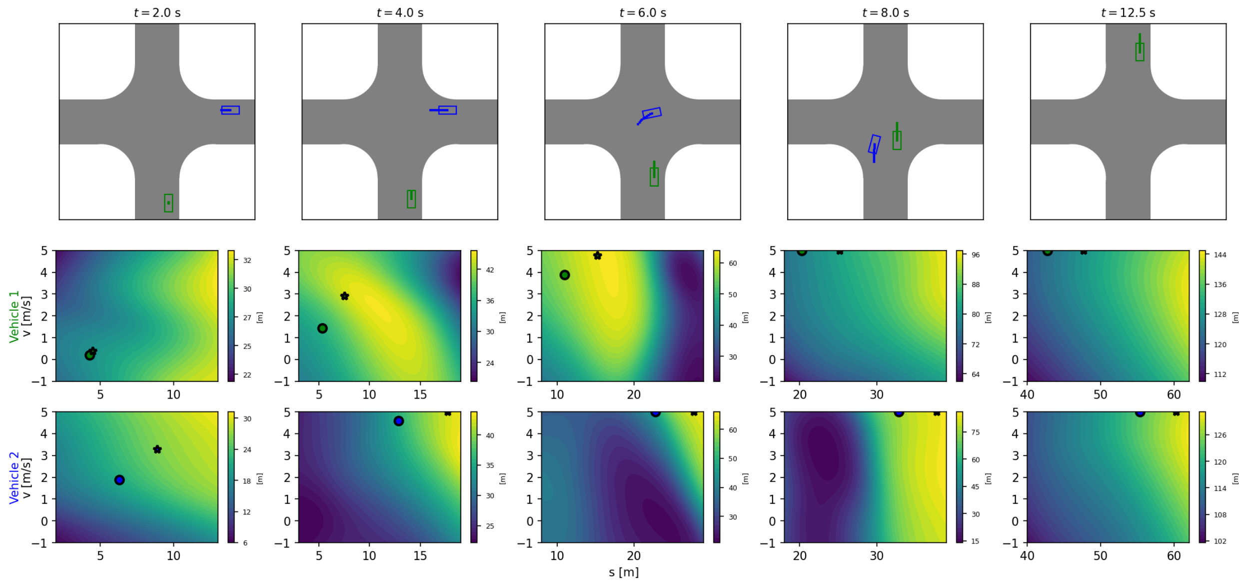


Figure 3.8: In a simulation experiment, Vehicle 1 (green) on route *SN* and Vehicle 2 (blue) on route *ES* navigate through an intersection using V_{GT} with a constant-velocity prediction model for the other vehicle's terminal state (Top), successfully resolving the intersection without gridlock. Colored squares represent each vehicle's planned trajectories. The middle and bottom rows display the contour of V_{GT} as perceived by Vehicle 1 and 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t + N$. The initial conditions are identical to those in Fig. 3.4

3.6 Chapter Summary

In this chapter, we presented IGT-MPC, an algorithm that enables decentralized, game-theoretic multi-agent motion planning in competitive and cooperative settings without requiring explicit game-theoretic modeling, which can be computationally expensive.

The approach involves generating a dataset of game-theoretic interactions by formulating constrained, dynamic games and solving for GNE from random initial conditions. The resulting reward outcomes, designed to reflect the multi-agent planning objective, are recorded alongside the initial conditions to train a value function. This learned value function, representing the terminal cost-to-go, is then applied in an MPC policy. We demonstrated IGT-MPC through two multi-agent examples: a head-to-head race and a two-vehicle navigation through an unsignalized intersection. The numerical results highlight the effectiveness of the learned value function in guiding MPC to replicate game-theoretic interactions, achieving competitive or coordinated behaviors. We further demonstrated IGT-MPC on a four-vehicle intersection navigation scenario, where it successfully reproduced the qualitative passing order of a Potential Game GNE solution from identical initial conditions, while naive MPC with the same horizon length of $N = 10$ failed to resolve the intersection. This result is promising evidence that the learned value function encodes game-theoretic strategic structure that generalizes beyond the two-agent setting for which it was trained. However, a more

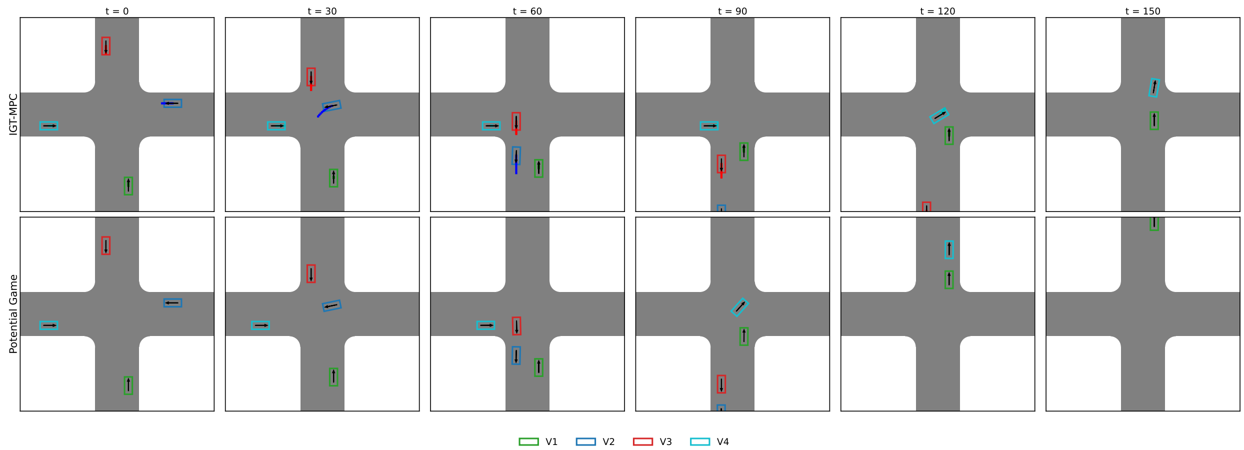


Figure 3.9: Snapshots of a four-vehicle intersection navigation scenario comparing IGT-MPC (top) and the Potential Game baseline (bottom) at various timesteps. V1 (green) is northbound, V2 (blue) is eastbound, V3 (red) is southbound, and V4 (cyan) is eastbound. Both methods start from identical initial conditions.

rigorous investigation into scalability is needed before broader conclusions can be drawn. We note that as the number of agents increases, generating GNE training data becomes increasingly expensive.

While the current IGT-MPC implementation focused on two-agent interactions, many multi-agent driving scenarios can be decomposed into a collection of pairwise interactions. In a traffic intersection with four vehicles, for instance, the ego vehicle does not simultaneously engage in a single four-player game; rather, it sequentially or concurrently resolves a series of two-agent interactions—yielding to one vehicle, negotiating priority with another, and ignoring a third that poses no conflict. However, pairwise decomposition may fail to capture higher-order strategic coupling that arises when the actions of three or more agents are mutually dependent. A direct extension that learns a single value function encoding the full multi-agent interaction structure, as described in the future work above, is a more principled approach and is sought for future research.

The chapters that follow shift from multi-agent coupling to a different area of MPC scalability challenges: the combinatorial growth of constraints arising from stochastic multi-modal predictions of surrounding vehicles.

3.7 Appendix

Implementation Details for Intersection Navigation Example

Cooperative Games

In a multi-agent cooperation problem, in which agents share resources, constraints, and objectives, a potential function can be derived to describe the coupling between agents and their collective objectives. Formulating cooperative games as potential games is advantageous because it captures the effects of all agents' strategy changes within a single global function, $\Phi(\mathbf{z}, \mathbf{u})$, whose minimizer corresponds to the GNE [70]. This simplifies both the analysis and computation of the GNE, as it can be formulated as a single optimization problem. Although the aforementioned solvers [104, 94, 56, 31] can address the game Γ in cooperative settings, the potential game formulation enables more efficient solution methods, including distributed optimization techniques such as Potential-iLQR [99]. In [99], the dynamics and the potential function are locally linearized and approximated around a nominal trajectory at each iteration, which induces a non-zero optimality gap. Since real-time computation is not critical during data generation, and assuming full coordination with a fully connected communication network, we formulate the potential game as a centralized optimization problem for the two-vehicle intersection example. Replacing $\{J^i\}_{i=1}^M$ with Φ, the potential function for coordinated navigation at an intersection used in our work is

$$\Phi(\mathbf{z}, \mathbf{u}) = \sum_{i=1}^{2} \sum_{k=0}^{N_G-1} \left(\ell_k^i(z_k^i, u_k^i)\right) - s_{t+N_G}^i, \tag{3.16}$$

It can be trivially shown that (3.16) is a convex potential function using its definition in [70]. Solving for the minimum of the potential function under the given constraints leads to a solution where no agent has an incentive to deviate from its strategy unilaterally, thus satisfying the conditions for a GNE. Therefore, the GNE is equivalent to the optimal solution of the following centralized optimization problem:

$$\begin{aligned}
\min_{\mathbf{u}} \ & \Phi(\mathbf{z}, \mathbf{u}) && \text{(3.17a)}\\
\text{s.t.} \quad & z_{k+1}^i = f^i(z_{k|t}^i, u_{k|t}^i), \quad \forall k \in \mathbb{I}_0^{N_G-1},\ \forall i \in \mathbb{I}_1^2 && \text{(3.17b)}\\
& z_k^i \in \mathcal{Z}^i, \quad \forall k \in \mathbb{I}_0^{N_G},\ \forall i \in \mathbb{I}_1^2 && \text{(3.17c)}\\
& u_k^i \in \mathcal{U}^i, \quad \forall k \in \mathbb{I}_0^{N_G-1}\ \forall i \in \mathbb{I}_1^2 && \text{(3.17d)}\\
& \Delta u_k^i \in \Delta\mathcal{U}^i, \quad \forall k \in \mathbb{I}_0^{N_G-1},\ \forall i \in \mathbb{I}_1^2 && \text{(3.17e)}\\
& P(z_k^1, z_k^2) \in \mathbb{CA}, && \text{(3.17f)}\\
& z_0^i = z^i(0) \quad \forall i \in \mathbb{I}_1^2 && \text{(3.17g)}
\end{aligned}$$

We formulate (3.17) in CasADi and solve the nonlinear program (NLP) with convex cost function using IPOPT [16]. After solving the potential game, the optimal solution, which corresponds to the GNE, is used to compute the reward outcome of the game. In this example, the main objective of the agents is to collaborate to ensure all agents pass the intersection safely. We use $R = \sum_{i=1}^{2} s_{N_G}^i$, which represents the collective progress along corresponding routes at the end of the game horizon N_G.

Scenario Encoding

We assume a predetermined reference trajectory is provided based on the desired route, and vehicles must adhere closely to it in compliance with traffic rules. A route is defined by a starting node and a destination node. In a four-way intersection, the nodes are (*N,W,S,E*) as depicted in Fig. 3.1b. For example, a left turn maneuver from *W* to N node is denoted as route *WN*. Note that no two vehicles can share identical routes. In the intersection example, we employ scenario encoding, where each agent i is assigned a scenario sc_i represented as a signed categorical variable. The magnitude of sc_i denotes a distinct interaction type between

Table 3.2: Scenarios for Two-vehicle Interactions at an Intersection

Scenario	Set of Route Combination Tuples $(+,-)$
$\mathcal{S}_1$	$\{(WE, NE), (NS, ES), (EW, SW), (SN, WN)\}$
$\mathcal{S}_2$	$\{(WN, SW), (NE, WN), (ES, NE), (SW, ES)\}$
$\mathcal{S}_3$	$\{(WE, NS), (NS, EW), (EW, SN), (SN, WE)\}$
$\mathcal{S}_4$	$\{(WN, EN), (NE, SE), (ES, WS), (SW, NW)\}$
$\mathcal{S}_5$	$\{(WE, SE), (NS, WS), (EW, NW), (SN, EN)\}$
$\mathcal{S}_6$	$\{(WE, SW), (NS, WN), (EW, NE), (SN, ES)\}$
$\mathcal{S}_7$	$\{(WN, ES), (NE, SW), (ES, WN), (SW, NE)\}$
$\mathcal{S}_8$	$\{(WN, EW), (NE, SN), (ES, WE), (SW, NS)\}$

the two vehicles, while the sign indicates the vehicle's role in that interaction. For instance, consider the green vehicle (vehicle 1) with a route *SN* and blue vehicle (vehicle 2) with a route *ES* depicted in Fig. 3.1b. In this case, the route combination tuple is $(SN, ES) \in \mathcal{S}_6$, where $\mathcal{S}_m$ are set of route combination tuples reported in Table 3.2 for $m \in \mathbb{I}_1^8$. The tuples in Table 3.2 are conventionally ordered where the vehicle of the first element is assigned a positive sign. Therefore, vehicle 1 is assigned $sc_1 = +6$ and vehicle 2 is assigned $sc_2 = -6$. The magnitude informs us that a vehicle going straight and a vehicle turning left into the same destination node as the vehicle going straight are interacting. Further, the vehicle going straight is given a positive sign while the vehicle turning left is assigned a negative sign to distinguish the role of each vehicle in this unique interaction. Note that scenarios such as both vehicles turning right at the intersection are not considered as vehicles do not interact. Also, each scenario exhibits rotational invariance due to rotational symmetry in the symmetric four-way intersection meaning (WE, NE) and (NS, ES) are identical two-vehicle interactions but rotated 90 degrees.

Training Value Function

In this example, we train separate V_{GT} for each scenario $\forall m \in \mathbb{I}_1^8$ with distinct datasets only containing samples with corresponding scenario encoding. In the current implementation, we are constrained to lightweight MLP architectures for fast gradient computations necessary for real-time planning (i.e. $\geq 10\ Hz$) and have trained separate V_{GT} models for each scenario. In future work, we aim to accelerate the solution of (3.12) by using faster NLP solvers to enable the use of more complex architectures for representing V_{GT}, which would improve generalization across multiple scenarios and intersection geometries, thereby enhancing the scalability of our method.

Additional Simulation Results

We further highlight the differences in how V_{GT} and V_{MP} guide the short-horizon MPC solutions through an intersection from the same initial condition in different scenarios. In Fig. 3.10, we show Vehicles 1 (green) on route *NS* and Vehicle 2 (blue) on *ES* navigating through an un-signalized intersection starting from rest, where both vehicles are controlled using identical MPC policies (3.12) with the same V_{GT}. As both vehicles approach the intersection at $t = 4\ s$, the velocity of vehicle 1 and 2 are 2.3 and 4.6 m/s, illustrating that the V_{GT} is guiding Vehicle 1 to approach the intersection at a slower velocity to yield to

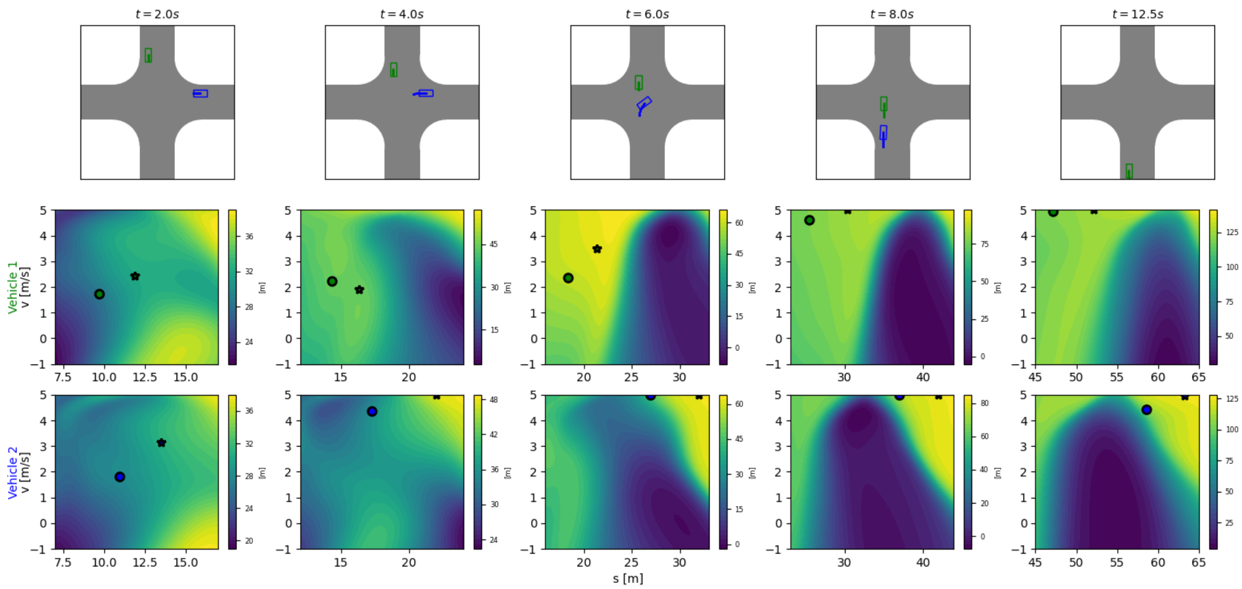


Figure 3.10: In a simulation experiment, Vehicle 1 (green) on route *NS* and Vehicle 2 (blue) on route *ES* navigate through an intersection using V_{GT} (Top). Colored squares represent each vehicle's planned trajectories. The middle and bottom rows display the contour of V_{GT} as perceived by Vehicle 1 and 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t + N$.

Vehicle 2, allowing both vehicles to pass the intersection by $t = 12.5\ s$. The blue regions shown in Fig. 3.10 represent undesirable terminal states that lead to collision or gridlocks, which the MPC policies avoid. In contrast, Vehicle 1 and Vehicle 2 are both approaching the intersection at 3.5 m/s at $t = 4\ s$ when guided by a naive V_{MP} in Fig. 3.11. The short-sightedness of the MPC policy and unaware of the interaction with other agents, the safety controller intervenes to avoid collision when guided by V_{MP} at $t = 6\ s$. Lastly, we show Vehicles 1 (green) on route *SN* and Vehicle 2 (blue) on *EW* navigating through an un-signalized intersection starting from rest, where both vehicles are controlled using identical MPC policies (3.12) with the same V_{GT} in Fig. 3.12. Interestingly, the level curves for Vehicle 1's V_{GT} at $t = 4\ s$ show a preference for slowing down within its reachable set in the $s - v$ state space while accelerating (high s, v region) reflect undesirable states. At $t = 6\ s$, Vehicle 1 V_{GT}'s topology changes and shows a preference for accelerating only after Vehicle 2 is passing the intersection.

In contrast, in Fig. 3.13 when both MPC planners are guided by V_{MP}, it prefers to maximize its progress without considering the impact of their actions on other agents as illustrated by linear level sets in the middle and bottom row of Fig. 3.13. Consequently, the vehicles become stationary and reach a gridlock.

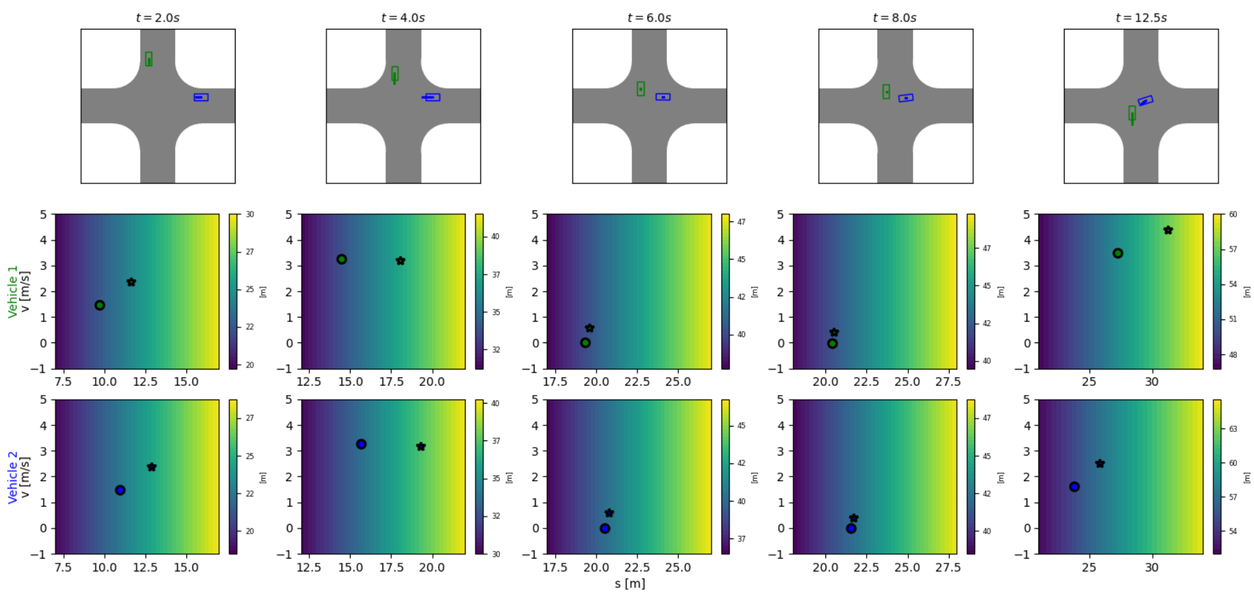


Figure 3.11: In a simulation experiment, Vehicle 1 (green) on route *NS* and Vehicle 2 (blue) on route *ES* navigate through an intersection using V_{MP} (Top), causing infeasibility of the MPC policy and requiring the safety controller to intervene. Colored squares represent each vehicle's planned trajectories. The middle and bottom rows display the contour of V_{MP} as perceived by Vehicle 1 and 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t + N$. The initial conditions are identical to those in Fig. 3.10

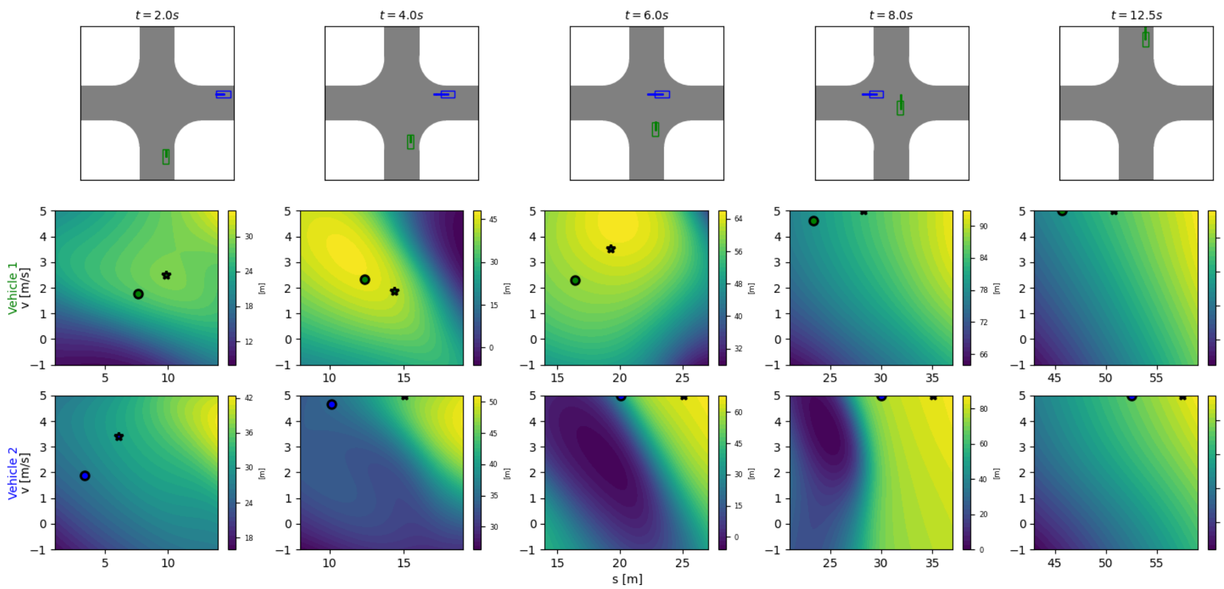


Figure 3.12: In a simulation experiment, Vehicle 1 (green) on route *SN* and Vehicle 2 (blue) on route *EW* navigate through an intersection using V_{GT} (Top). Colored squares represent each vehicle's planned trajectories. The middle and bottom rows display the contour of V_{GT} as perceived by Vehicle 1 and 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t + N$.

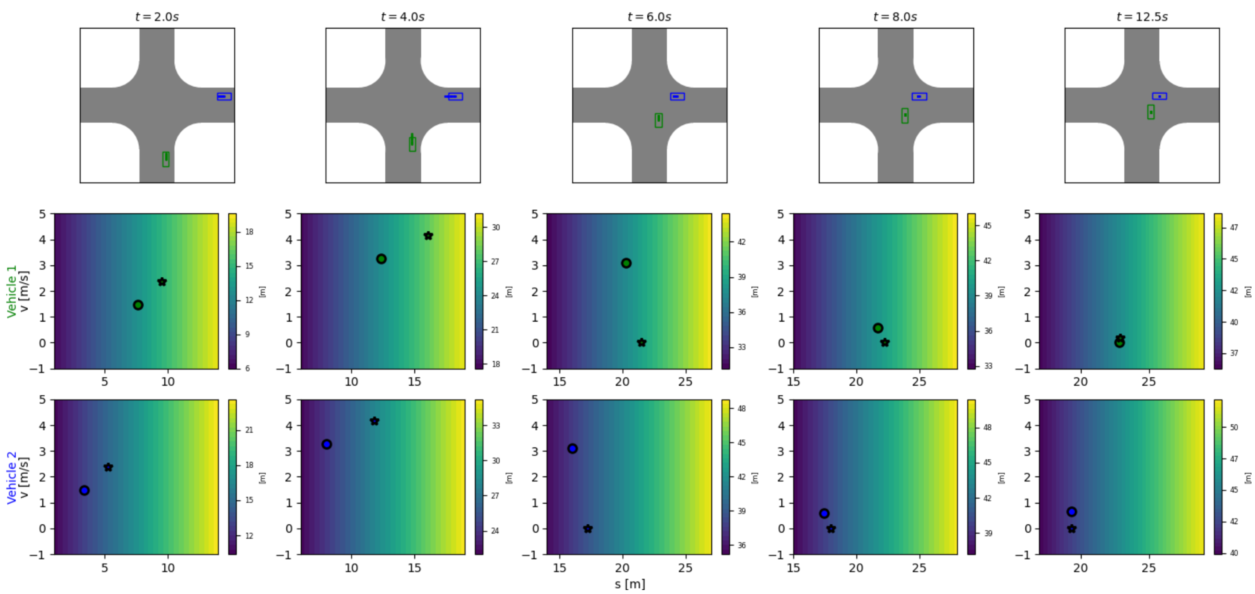


Figure 3.13: In a simulation experiment, Vehicle 1 (green) on route *SN* and Vehicle 2 (blue) on route *EW* navigate through an intersection using V_{MP} (Top), reaching a gridlock. Colored squares represent each vehicle's planned trajectories. The middle and bottom rows display the contour of V_{MP} as perceived by Vehicle 1 and 2, respectively. Colored circles indicate current states at each time instance, and colored stars denote the planned terminal state at time $t + N$. The initial conditions are identical to those in Fig. 3.12

Chapter 4

Learning-Guided Scalable MPC using Duality-based Criterion: Constraint Reduction

This chapter is adapted from a previous publication: Hansung Kim, Siddharth H. Nair, and Francesco Borrelli. "Scalable Multi-modal Model Predictive Control via Duality-based Interaction Predictions". In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 1499–1504. DOI: 10.1109/IV55156.2024.10588718

This chapter shifts from the multi-agent coupling challenges addressed in Part I to the scalability bottleneck introduced in Chapter 1: the combinatorial growth of collision-avoidance constraints when stochastic MPC must handle multi-modal predictions of surrounding vehicles. As the number of vehicles and predicted modes increases, the constraint count scales as $O(N \cdot M^V)$, rendering real-time computation impractical. We use Lagrangian duality to interpret which vehicles and which of their predicted modes interact with the ego vehicle along the prediction horizon, and derive a sensitivity-based screening criterion for removing inactive constraints. To predict these interactions at runtime, we propose RAID-Net, an attention-based recurrent neural network trained via imitation learning on optimal dual solutions. The resulting hierarchical algorithm solves a reduced MPC problem containing only the relevant constraints, achieving an order-of-magnitude speedup in solve time.

4.1 Introduction

Autonomous driving research on motion planning has graduated to address challenges in complex, interaction-driven urban scenarios over the past few years. In these settings, Autonomous Vehicles (AVs) must navigate safely through highly uncertain environments, while observing and reacting to heterogeneous traffic agents: human-driven and other AVs navigating and making their own decisions. Accounting for these agents during motion planning presents intricate challenges, demanding robust and real-time solutions for safe navigation in urban environments.

In urban driving scenarios, the motion planning for AVs in the presence of uncertain, multi-modal agents poses a significant challenge, leading to the development of various solutions for planning and behavior prediction. These solutions can be categorized into three broad approaches: (i) Hierarchical Prediction and

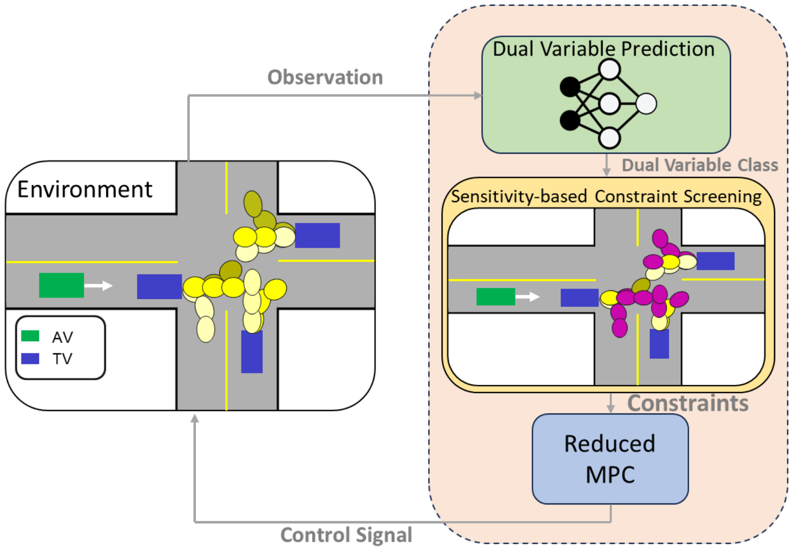


Figure 4.1: Our hierarchical architecture for motion planning with duality-based interaction prediction. Given the environment observation, we classify which *vehicles (blue) and their maneuvers can be eliminated/screened (magenta ellipses)* for solving a reduced, real-time MPC problem for the AV (green rectangle).

Planning [83, 73]: where a sophisticated prediction architecture provides forecasts of the surrounding vehicles which is used for motion planning, (ii) Model-based Integrated Planning and Prediction [25, 79]: where planning and behavior prediction are simultaneously obtained by game-theoretic and joint-optimization approaches for all vehicles, and (iii) End-to-End Learning-based Prediction and Control [57, 38, 81]: where a sophisticated neural network (NN), trained using imitation/reinforcement learning algorithms on realistic datasets, implicitly and jointly forecasts the behavior of surrounding agents and a motion plan for the AV. Each of these approaches suffers from either scalability for complex driving scenarios or interpretability and safety of the computed motion plans. For instance in (i), [73] employs Gaussian mixture models to explicitly express multi-modal predictions and positional uncertainty of surrounding vehicles for multi-modal motion planning. While this approach showcases robust navigation capabilities in multi-modal traffic scenarios, it focuses on interactions in simple traffic situations with limited agents and does not scale for real-time applicability in complex scenarios with many surrounding vehicles and their multi-modal predictions. The game theoretic approaches in (ii) are generally computationally intractable for traffic scenarios with many agents, which is further exacerbated when the games are multi-modal/mixed. The End-to-End approaches in (iii) are generally scalable to complex traffic scenarios but lack interpretability in their predictions. Given the traffic scene, it is unclear how to interpret which agents and which of their possible maneuvers are relevant for a safe motion plan for the AV. In [57], an attention-based architecture discerns relevant vehicles at the current time-step when controlling an AV in a simulated traffic intersection, but does not explicitly consider safety or long-term interactions. Hence, there is a pressing need for a scalable motion planning method that incorporates multi-modal predictions of surrounding agents while safely navigating complex interaction-driven scenarios for autonomous driving.

In this chapter, we present a scalable real-time hierarchical motion planning algorithm given multi-modal predictions in complex, interaction-driven traffic scenarios. In the design stage we use a optimization-based planner and its Lagrangian dual to deduce which surrounding vehicle and its multi-modal predictions/maneuvers along the prediction horizon are *interacting* with the AV. We design a Recurrent Neural Network (RNN) [65] with attention mechanisms to predict the interactions between the AV and its surrounding vehicles. The RNN is trained from data from the MPC solutions using a novel constraint screening criterion which uses ideas from sensitivity analysis as depicted in Fig. 4.1. The novel attention-based RNN architecture for interaction prediction is interpretable using Lagrange duality and generalizable to varying numbers of surrounding vehicles. We demonstrate the real-time applicability of our proposed algorithm in numerical simulations of traffic intersections with interactive surrounding vehicles.

The chapter is structured as follows. Section 4.3 formulates Stochastic Model Predictive Control (MPC) for motion planning with multi-modal predictions, discussing scalability issues. In Section 4.4, we present a duality-based interpretation of the interaction between the AV and surrounding vehicles and introduce an RNN architecture for predicting interactions with multi-modal forecasts. A hierarchical algorithm for motion planning is outlined, utilizing NN predictions and a dual sensitivity-based constraint screening mechanism. Section 5.4 evaluates our hierarchical MPC algorithm at a simulated traffic intersection. Finally, Section 4.6 concludes the chapter and discusses future research directions.

4.2 Preliminaries

Notations

The index set $\{k_1, k_1+1, \dots, k_2\}$ is denoted by $\mathbb{I}_{k_1}^{k_2}$. For a proper cone $\mathbb{K}$ and $x, y \in \mathbb{K}$, the dual cone of $\mathbb{K}$ is given by the convex set $\mathbb{K}^* = \{y | y^\top x \geq 0, \forall x \in \mathbb{K}\}$. $||\cdot||_p$ denotes the $p-$norm. For any matrix $A \in \mathbb{R}^{n\times m}$, we denote $[A]_{\mathbb{S}}$ to be the rows of A with indices in $\mathbb{S} \subset \mathbb{I}_1^n$. For any pair of vectors $x, y \in \mathbb{R}^n$, we denote the Hadamard product as $x \circ y = [[x]_1[y]_1, .., [x]_n[y]_n] \in \mathbb{R}^n$.

4.3 Problem Formulation

Vehicle Prediction Models

Let the AV be described by a linear time-varying dynamics

$$x_{t+1} = A_t x_t + B_t u_t + E_t w_t \tag{4.1}$$

where $x_t \in \mathbb{R}^{n_x}, u_t \in \mathbb{R}^{n_u}, w_t \in \mathbb{R}^{n_x}$ are the state, input and process noise respectively and A_t, B_t, E_t are the system matrices at time t. The process noise w_t is assumed to be a zero mean Gaussian, $\mathcal{N}(0, \Sigma_t)$.

Now suppose that there are V Target Vehicles (TVs), each described by *multi-modal*, affine time-varying discrete-time predictions over a horizon N at time t,

$$\begin{aligned} o^i_{k+1|t,j} = T^i_{k|t,j} o^i_{k|t,j} + q^i_{k|t,j} + E^i_{k|t,j} n^i_{k|t,j}, \\ \forall k \in \mathbb{I}_t^{t+N}, i \in \mathbb{I}_1^V, j \in \mathbb{I}_1^M \end{aligned} \tag{4.2}$$

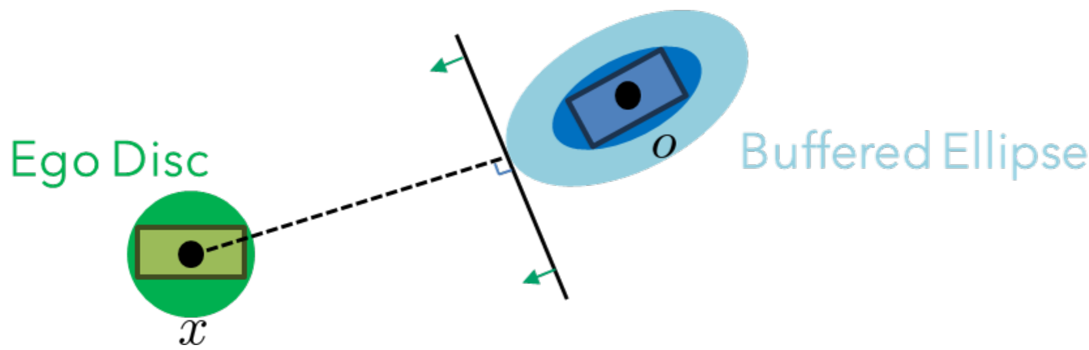


Figure 4.2: Collision avoidance geometry: the ego vehicle (green disc, center x) and the target vehicle (dark ellipse, center o) buffered by the ego disc's radius (light blue ellipse). The separating half-space constraint (solid line) is constructed at the boundary of the buffered ellipse, ensuring the ego center x remains on the collision-free side.

where $o^i_{k|t,j}, n^i_{k|t,j} \in \mathbb{R}^{n_o}$ are the position and process noise predictions for TV i in mode j and time t, at the kth time step and $T^i_{k|t,j}, q^i_{k|t,j}, E^i_{k|t,j}$ are the time, mode-dependent system matrices and process noise $n^i_{k|t,j} \sim \mathcal{N}(0, \Sigma^i_{k|t,j})$.

Notice that each TV prediction has M modes, which amounts to a total of M^V mode configurations of all the TVs. We call each mode configuration a *scenario* and define the function $\bar{j} : \mathbb{I}_1^V \times \mathbb{I}_1^{M^V} \to \mathbb{I}_1^M$ such that $\bar{j}(i, m)$ returns the mode of TV i in scenario m.

Constraints

The AV is subject to polytopic state-input constraints given by $\mathbb{XU}_t = \{(x, u)|\ F^x_t x + F^u_t u \leq f_t\}$, for time t and affine collision avoidance constraints given as $\mathbb{CA}^i_{k|t,j} = \{(x, o)|\ L^{x,i}_{k|t,j} x + L^{o,i}_{k|t,j} o \leq l^i_{k|t,j}\ \}$ for vehicle i in mode j at time step k, which are obtained by appropriate inner-approximations of the free space as half spaces as illustrated in Fig. 4.3 [15, 72]. The ego vehicle body is approximated as a disc and the target vehicle body as an ellipse, buffered by the ego disc's radius to account for the ego's spatial extent [74]. Collision then occurs if and only if the ego disc's center enters the buffered ellipse, reducing the collision avoidance constraint to a half-space constraint on the ego vehicle's center. Over the horizon N, this affine constraint is time-varying, with each half-space computed from the ego's planned trajectory and the target vehicle's predicted trajectory at the corresponding time step.

Stochastic MPC formulation

We compute a state-feedback control $u_t = \pi(x_t, o_t)$ for the controlled by solving the following finite-horizon stochastic optimal control problem towards tracking a given reference $\{x^{\text{ref}}_k, u^{\text{ref}}_k\}^{t+N}_{k=t}$. Specifically, we use affinely parameterized feedback policies $\pi_{\theta^i_{k|t,j}}(\mathbf{x}_t, \mathbf{o}_t)$ in (4.4) for the control actions along the prediction horizon for time k and scenario m,

$$u_{k|t,m} = h_{k|t} + \sum_{i=1}^{V} K^i_{k|t,\bar{j}(i,m)}(o^i_{k|t,\bar{j}(i,m)} - \mathbb{E}[o^i_{k|t,\bar{j}(i,m)}]) \tag{4.3}$$

where variables $\theta^i_{k|t,j} = \{h_{k|t}, K^i_{k|t,j}\}$ and the second term in (4.3) encodes state feedback for the obstacles' states. This term allows the AV to *react* to different realizations of the obstacles' trajectories, thus greatly enhancing the feasibility of the MPC problem (especially in the presence of large uncertainties). Using this feedback policy parameterization, the constrained finite-time optimization problem becomes

$$\min_{\boldsymbol{\theta}_t} \sum_{m=1}^{M^V} \mathbb{E}\left[\sum_{k=t}^{t+N-1} ||Q(x_{k+1|t,m} - x^{\text{ref}}_k)||_2^2 + ||R(u_{k|t,m} - u^{\text{ref}}_k)||_2^2\right] \tag{4.4a}$$

$$\text{s.t.} \quad x_{k+1|t,m} = A_k x_{k|t,m} + B_k u_{k|t,m} + E_k w_{k|t}, \tag{4.4b}$$

$$o^i_{k+1|t,j} = T^i_{k|t,j} o^i_{k|t,j} + q^i_{k|t,j} + E^i_{k|t,j} n^i_{k|t,j}, \tag{4.4c}$$

$$\mathbb{P}((x_{k|t,m}, u_{k|t,m}) \notin \mathbb{XU}_k) \leq \epsilon, \tag{4.4d}$$

$$\mathbb{P}((x_{k|t,m}, o^i_{k|t,\bar{j}(i,m)}) \notin \mathbb{CA}^i_{k|t,\bar{j}(i,m)}) \leq \epsilon, \tag{4.4e}$$

$$u_{k|t,m} = h_{k|t} + \sum_{i=1}^{V} K^i_{k|t,\bar{j}(i,m)}(o^i_{k|t,\bar{j}(i,m)} - \mathbb{E}[o^i_{k|t,\bar{j}(i,m)}]), \tag{4.4f}$$

$$u_{t|t,m} = u_{t|t,1},\ x_{t|t,m} = x_t,\ o_{t|t,j} = o_t, \tag{4.4g}$$

$$\forall k \in \mathbb{I}_t^{t+N-1}, j \in \mathbb{I}_1^M, m \in \mathbb{I}_1^{M^V}, i \in \mathbb{I}_1^V$$

where the decision variables

$$\boldsymbol{\theta}_t = \{(h_{k|t}, \{\{K^i_{k|t,j}\}_{i=1}^V\}_{j=1}^M)\}_{k=t}^{t+N} \tag{4.5}$$

parameterize the feedback policies (4.4f) of the controlled agent. The second term in (4.4f) encodes state feedback for the obstacles' states, to allow the AV to *react* to different realizations of the obstacles' trajectories, thus greatly enhancing the feasibility of the MPC problem (especially in the presence of large uncertainties). The Stochastic MPC (SMPC) is given by the optimal solution of (4.4) as

$$u_t = \pi_{\text{MPC}}(x_t, o_t) = u^\star_{t|t,1}. \tag{4.6}$$

The constraint $u_{t|t,m} = u_{t|t,1}$ is called the *non-anticipatory* constraint. It enforces that the control applied at time t is independent of scenario m, encoding the fact that the AV doesn't know the *true modes* of the V TVs at time t. The problem has $\mathcal{O}(NMV)$ decision variables $\boldsymbol{\theta}_t$ but exponentially growing $\mathcal{O}(NM^V)$ constraints, arising from the multi-modal predictions and collision avoidance constraints. The combinatorial explosion of the collision avoidance constraints when handling multi-modal forecasts is explained in the next section. Since the solution to (4.4) is sought for real-time applications at frequencies $\geq 10Hz$, complex scenarios with many obstacles and modes can render the computation of (4.6) impractical.

The goal of this article is to accelerate the solution of (4.4) for real-time applications. In practice, not all collision avoidance constraints need to be enforced for solving the optimization problem (4.4). We use Lagrange duality to interpret which TVs and their corresponding modes *interact* with the AV along the prediction horizon. Leveraging this insight, we use imitation learning to train an attention-based RNN architecture for predicting which TVs, and which modes should be considered in (4.4).

Multi-Modal Prediction Models and the Combinatorial Explosion

State-of-the-art trajectory prediction models for autonomous driving produce multi-modal, probabilistic forecasts that capture the inherent uncertainty in human driving behavior. A vehicle approaching an intersection may turn left, proceed straight, or turn right; a vehicle on a highway may maintain its lane, merge, or decelerate. Modern prediction architectures represent this uncertainty by outputting a distribution over multiple possible future trajectories for each agent, typically parameterized as a Gaussian Mixture Model (GMM) or a discrete set of trajectory modes with associated probabilities.

Trajectron++ [83] is a graph-structured recurrent model that forecasts trajectories of multiple agents using a Conditional Variational Autoencoder (CVAE) decoder, producing a GMM over future positions for each agent conditioned on the scene context and agent interactions. Wayformer [75], proposed by Waymo, uses a transformer-based encoder-decoder architecture that processes heterogeneous scene inputs (road geometry, lane connectivity, traffic light state, agent history) through attention mechanisms and outputs multiple trajectory hypotheses per agent following the MultiPath++ regression-based decoding scheme [90]. Both models have demonstrated strong performance on large-scale benchmarks such as the Waymo Open Motion Dataset and Argoverse.

A critical property of these models is that they produce *marginal* predictions: each agent's multi-modal forecast is generated independently, conditioned on the observed scene but not on the future behavior of other agents. That is, for agent i, the prediction model outputs M trajectory modes $\{o_j^i\}_{j=1}^M$ with associated probabilities $\{p_j^i\}_{j=1}^M$, where $\sum_j p_j^i = 1$. These modes represent the marginal distribution over agent i's future trajectory. The prediction model does not output a joint distribution over the futures of all agents simultaneously—doing so would require modeling the full combinatorial space of multi-agent behaviors, which is itself intractable for current architectures and datasets.

When these marginal predictions are used for motion planning, however, the planner must reason about the joint future of all agents to ensure safety. If the ego vehicle must avoid collisions with V surrounding vehicles, each with M predicted modes, it must in principle consider all M^V possible combinations of modes across agents—each combination representing one possible joint future scenario. For each scenario, collision-avoidance constraints must be imposed at every time step k along the prediction horizon N, yielding $O(N \cdot M^V)$ constraints in total. For the problem sizes considered in Chapters 4 and 5 ($V = 3$, $M = 2$, $N = 14$), this produces 312 collision-avoidance constraints. For more realistic settings with $V = 5$ vehicles and $M = 3$ modes, the constraint count exceeds 30,000, rendering the optimization intractable at real-time rates.

This combinatorial explosion is a direct consequence of the gap between marginal prediction and joint planning: prediction models provide per-agent distributions, but safe planning requires reasoning over the joint space of all agents' futures. Recent work has begun to address this gap by developing joint prediction models that output scene-level modes rather than per-agent marginal modes. Scene Transformer [76] uses a unified transformer architecture to jointly predict futures for all agents, producing K scene-consistent trajectory sets where each set specifies one future trajectory per agent. JFP [63] explicitly models pairwise interactions in the decoder to generate consistent multi-agent futures. M2I [87] takes a factored approach, conditioning one agent's prediction on the other's to produce interaction-consistent trajectory pairs. If such joint models were used in the planning pipeline, the combinatorial explosion would be eliminated by construction: instead of M^V mode combinations, the planner would receive K scene-level scenarios directly, yielding $O(N \cdot K)$ constraints regardless of the number of agents.

However, joint prediction models face several fundamental limitations that prevent them from replacing marginal models in practice. First, they suffer from *mode collapse*: recent work [24] demonstrated that joint prediction models frequently fail to cover all relevant interaction modes, particularly in safety-critical scenarios. A model may correctly predict that one vehicle yields while the other proceeds, but fail to represent the symmetric alternative where the roles are reversed—precisely the mode that matters most for safe planning, as the planner must hedge against both possibilities. The Waymo Open Motion Dataset benchmark reflects this difficulty: joint metrics (scene-level minADE, minFDE, miss rate) are substantially harder than their marginal counterparts because the minimum is taken over the whole scene rather than per-agent, meaning a single poorly predicted agent degrades the entire score [26]. Second, the number of scene-level modes K that a joint model can output is fundamentally limited by architecture and training—typically $K = 6$ or $K = 8$—whereas the true joint behavior space grows exponentially with the number of agents. This means joint models must implicitly compress the exponential space into a small fixed set of modes, inevitably losing coverage of low-probability but safety-critical futures. Third, joint models can produce *counterfactual forecasts*: scene-consistent modes that are internally coherent but assign probability mass to joint futures that would not arise under the agents' actual decision-making processes, because the model has no access to the agents' true objectives or constraints [87]. Finally, from a systems perspective, marginal prediction models remain dominant in deployed autonomous driving stacks due to their maturity, computational efficiency, and modularity—the prediction module can be developed, validated, and updated independently of the planner.

For these reasons, the combinatorial scaling problem arising from marginal multi-modal predictions remains a practically relevant and theoretically interesting challenge. In this work, we accept the marginal predictions as given and efficiently identify which of the M^V constraint combinations are active, removing the rest to accelerate the computation of the underlying optimal control problem for real-time applications.

4.4 Duality-based Interaction Prediction using Imitation Learning

Convex MPC formulation and Duality

All the chance constraints can be deterministically reformulated as second-order cone (SOC) constraints in $\boldsymbol{\theta}_t$ because the process noise w_t and $n^i_{k|t,j}$ are assumed to be Gaussian[9, Chapter 5], [72]. We write the MPC optimization problem compactly as the following SOCP:

$$\begin{aligned} \min_{\boldsymbol{\theta}_t} \quad & \frac{1}{2}||\mathcal{Q}_t\boldsymbol{\theta}_t||_2^2 + \mathcal{C}_t^\top \boldsymbol{\theta}_t \\ \text{s.t}\ & \mathcal{A}_t\boldsymbol{\theta}_t + \mathcal{R}_t \in \mathbb{K} := (\bigotimes_{s=1}^{n_c} \mathbb{K}_s) \times \mathbb{K}_{xu} \end{aligned} \tag{4.7}$$

where $\mathcal{A}_t, \mathcal{R}_t, \mathcal{Q}_t, \mathcal{C}_t$ represent the predictions models and constraints, $n_c = N \cdot M^V$, $\mathcal{Q}_t \succ 0$ and $\mathbb{K}_s, \forall s \in \mathbb{I}_1^{n_c}$ are the cones corresponding to the collision avoidance and $\mathbb{K}_{xu}$ is the cone that corresponds to all the state-input constraints. The exponential number of state-input constraints in (4.4d) can be reduced to $\mathcal{O}(NMV)$ for $\mathbb{K}_{xu}$ by employing standard constraint-tightening techniques from robust optimization 4.7. For the $\mathcal{O}(NM^V)$ collision avoidance constraints, we would like to solve a reduced version of (4.7) involving only the *necessary* collision avoidance constraints. We quantify this necessity using Lagrange duality. Consider the dual problem of (4.7),

$$\begin{aligned} \min_{\boldsymbol{\mu}_t, \boldsymbol{\eta}_t} \ & [\boldsymbol{\mu}_t^\top \ \boldsymbol{\eta}_t^\top]\mathcal{R}_t + \frac{1}{2}||\mathcal{Q}_t^{-1}(\mathcal{A}_t^\top[\boldsymbol{\mu}_t^\top \ \boldsymbol{\eta}_t^\top]^\top - \mathcal{C}_t)||_2^2 \\ \text{s.t}\ & \boldsymbol{\mu}_t \in \bigotimes_{s=1}^{n_c} \mathbb{K}_s^*, \boldsymbol{\eta}_t \in \mathbb{K}_{xu}^* \end{aligned} \tag{4.8}$$

Notice that the conic feasible set of the dual problem is independent of time t, and is comprised of the standard SOCs and positive orthants, for which closed-form projections can be computed efficiently[1]. If strong duality holds, then the optimal primal and dual solutions $(\boldsymbol{\theta}_t^\star, \boldsymbol{\mu}_t^\star, \boldsymbol{\eta}_t^\star)$ satisfy the KKT conditions:

$$\begin{aligned} & \mathcal{Q}_t^2\boldsymbol{\theta}_t^\star + \mathcal{C}_t - \mathcal{A}_t^\top \begin{bmatrix} \boldsymbol{\mu}_t^\star \\ \boldsymbol{\eta}_t^\star \end{bmatrix} = 0 \\ & \mathcal{A}_t\boldsymbol{\theta}_t^\star + \mathcal{R}_t \in \mathbb{K}, \ \ \boldsymbol{\mu}_t^\star \in \bigotimes_{s=1}^{n_c} \mathbb{K}_s^*, \quad \boldsymbol{\eta}_t^\star \in \mathbb{K}_{xu}^* \\ & [\mathcal{A}_t\boldsymbol{\theta}_t^\star + \mathcal{R}_t]_{xu} \circ \boldsymbol{\eta}_t^\star = 0, \\ & [\mathcal{A}_t\boldsymbol{\theta}_t^\star + \mathcal{R}_t]_s \circ [\boldsymbol{\mu}_t^\star]_s = 0 \ \ \forall s \in \mathbb{I}_1^{n_c} \end{aligned} \tag{4.9}$$

where the $[\boldsymbol{\mu}_t^\star]_s$ denotes dual variable belonging to sth cone $\mathbb{K}_s^*$, and $[\mathcal{A}_t\boldsymbol{\theta}_t^\star + \mathcal{R}_t]_{xu}$ denotes the rows of $\mathcal{A}_t\boldsymbol{\theta}_t^\star + \mathcal{R}_t$ that correspond the state-input constraints. Observe from the last equation that if $[\boldsymbol{\mu}_t^\star]_s = 0$, then $[\mathcal{A}_t\boldsymbol{\theta}_t^\star + \mathcal{R}_t]_s \in \text{int}(\mathbb{K}_s)$ and this constraint can be eliminated from (4.7).

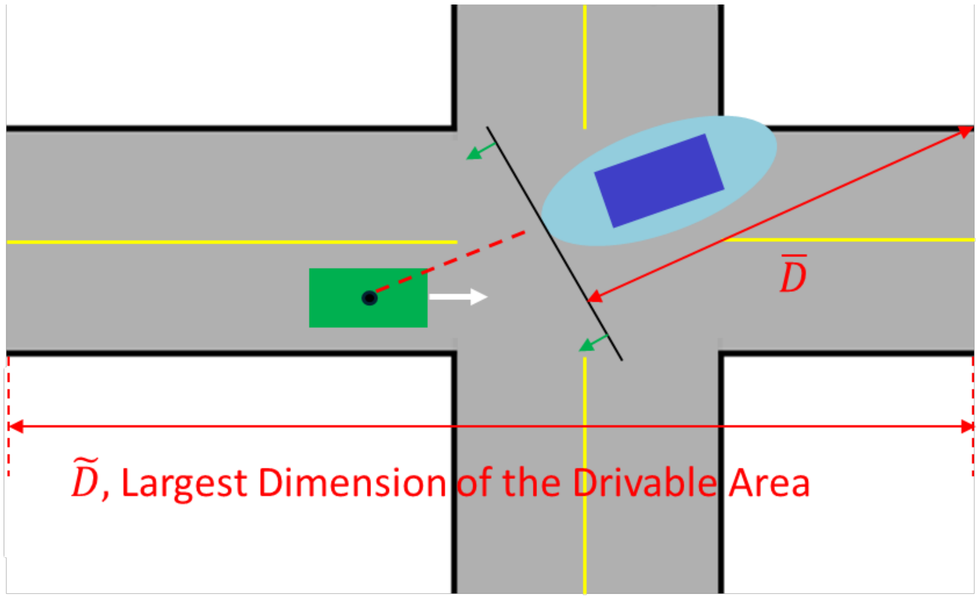


Figure 4.3: Illustration of $\bar{D}$, the largest dimension of the drivable area, used to bound the separating half-space constraint distance. When no collision constraint is active, the half-space is placed at a distance $\bar{D}$ from the ego vehicle (green), ensuring the constraint remains feasible and inactive without affecting the planned trajectory.

Now suppose that we are given a dual feasible candidate $\hat{\boldsymbol{\mu}}_t \in \bigotimes_{s=1}^{n_c} \mathbb{K}_s^*$. Let $S^i[\hat{\boldsymbol{\mu}}_t]$ denote the duals corresponding to collision avoidance constraints with vehicle i, $S_m[\hat{\boldsymbol{\mu}}_t]$ denote the duals corresponding to collision avoidance constraints in a particular mode configuration scenario m and $[\mathcal{A}_t\boldsymbol{\theta}+\mathcal{R}_t]_{xu}$ be the rows corresponding to the state-input constraints. We will use ideas from sensitivity analysis of conic programs to eliminate vehicles, and scenarios to construct a reduced MPC problem as follows. Define the maximum possible violation of the collision avoidance constraints across all vehicles for an AV trajectory that satisfies state and input constraints,

$$\begin{aligned}\bar{D} := \max_{i,\boldsymbol{\theta}^i} \min_{\boldsymbol{\delta}^i} \quad & ||S^i[\mathcal{A}_t\boldsymbol{\theta}^i+\mathcal{R}_t]-\delta^i||_2\\ \text{s.t} \quad & i\in\mathbb{I}_1^V, [\mathcal{A}_t\boldsymbol{\theta}^i+\mathcal{R}_t]_{xu}\in\mathbb{K}_{xu}, \delta^i\in S^i[\mathbb{K}]\end{aligned}$$

The quantity $\bar{D}$ is guaranteed to be finite because (i) the AV and TV trajectories are bounded by dynamical and actuation constraints, and (ii) the vehicle geometries are compact sets. $\bar{D}$ need not be explicitly computed; any $\tilde{D} > \bar{D}$ suffices. A conservative choice is $\tilde{D} = N \cdot M \cdot W$, where W is the largest dimension of the drivable area, as illustrated in Fig. 4.4.

Let δ be an acceptable deviation in the optimal cost of (4.4). By the *shadow price* interpretation of the duals $S^i[\hat{\boldsymbol{\mu}}_t]$ [13, Chapter 5], if $\|S^i[\hat{\boldsymbol{\mu}}_t]\| \leq \frac{\delta}{\bar{D}}$, then violating the collision avoidance constraints for TV i by at most $\bar{D}$ changes the optimal cost by at most $\frac{\delta}{\bar{D}} \cdot \bar{D} = \delta$. Hence, TV i can be safely ignored without significantly affecting the motion plan. Similarly, if $\|S_m[\hat{\boldsymbol{\mu}}_t]\| \leq \frac{\delta}{\bar{D}\cdot V}$, scenario m can be pruned, changing the cost by at most δ across all V TVs in that scenario.

Recurrent Attention for Interaction Duals Network (RAID-Net)

The optimal dual variables $\boldsymbol{\mu_t^\star} \in \bigotimes_{s=1}^{n_c} \mathbb{K}_s^*$ contain information about active constraints of the primal optimization problem (4.7) predicted at time t. The continuous vector can be converted into a binary vector $\tilde{\boldsymbol{\mu}}_t^\star \in \{0,1\}^{n_c}$ by applying the screening rules described in 4.4. If $[\boldsymbol{\mu_t}]_s > 0$, then the corresponding constraint is active and $[\tilde{\boldsymbol{\mu}}_t^\star]_s = 1$. Otherwise, $[\tilde{\boldsymbol{\mu}}_t^\star]_s = 0 \quad \forall s \in \mathbb{I}_1^{n_c}$.

We propose Recurrent Attention for Interaction Duals Network (RAID-Net) that predicts the dual class $\tilde{\boldsymbol{\mu}}_t$ given the observation (ob_t) of the environment, which may contain continuous or discrete information about the environment.

Expert Dataset

With access to a simulation environment, the optimal dual solutions $\{\boldsymbol{\mu_t^\star}\}_{t=0}^T$ (i.e. expert actions in IL nomenclature) are obtained by solving (4.7) at each step t along the trajectory with length T. Then, it is converted into the binary class labels $\{\tilde{\boldsymbol{\mu}}_t^\star\}_{t=0}^T$. The optimal dual class labels and observation of the environment pairs are stored into a dataset $\mathcal{D} = \{(ob_t, \tilde{\boldsymbol{\mu}}_t^\star)\}_{t=0}^B$, where B is the user-defined dataset size.

Input Normalization and Encoding

We assume that the observation of the environment is available and includes information about the AV's states, previous control input, and TVs' states along with semantic information about the AV and TVs' modes.

The observation is normalized using environment and system information such as maximum states and inputs (restricted by the environment and system limitations), the maximum number of modes per vehicle, etc. In the context of complex interactive autonomous driving scenarios, employing graph encoding of the scene is crucial as it allows for a comprehensive representation of the intricate relationships and interactions among various entities within the scene. Nodes in the graph represent vehicles including the AV, while directed edges (originating from the AV) capture the dynamic relationships and interaction between the AV and TVs. We utilize an AV-centric graph encoding using time-to-collision (TTC) between the AV to the TVs to represent the edges. In RAID-Net the observation is separated per i-th vehicle: $ob_t^i \quad \forall i \in \mathbb{I}_0^V$, where $i = 0$ represents the AV and augmented with the TTC graph encoder. Subsequently, a transformer-based encoder embeds the augmented observations, producing a scene representation feature vector denoted as $f_i \in \mathbb{R}^{d_{em}}$. We define $d_{em} = 2 \times |ob|$, where $|ob|$ represents the dimension of the observation space.

The encoder exhibits permutation invariance, indicating that the network's output remains unchanged even when the input order is interchanged. This invariance is achieved by summing the feature vectors $\{f_i\}_{i=0}^V$ over all vehicles before feeding them into the decoder as depicted in Fig. 4.1.

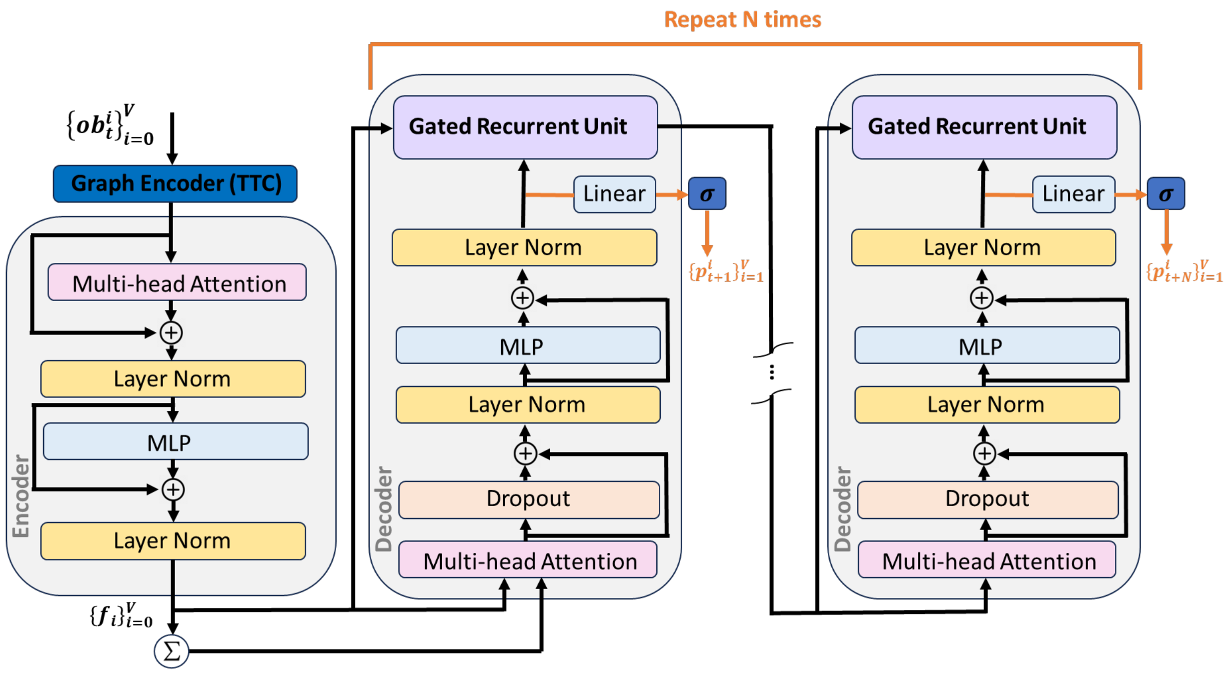


Figure 4.4: Schematic of RAID-Net for predicting relevant constraints for (4.4). RAID-net is invariant to the number and order of TVs, and has a MPC horizon-independent memory footprint because of its recurrent architecture.

Network Architecture

The normalized and encoded inputs undergo N sequential passes through a shared parameter attention-based gated recurrent network of decoders, as depicted in Fig. 1. The decoder integrates a Multi-Head Attention (MHA) mechanism to capture various relationships among the encoded inputs [13]. Each head of the attention mechanism focuses on distinct interactions between AVs and TVs. Refer to [91] for detailed explanations of the MHA network architecture. The MHA network in the decoder has embedding dimensions of d_{em} and n_h, representing the number of heads. Additionally, the decoder includes dropout and layer normalization layers to address overfitting and stabilize training [85, 6], along with a multi-layer perceptron (MLP) to capture complex data relationships within the embedded dimension.

The hidden states and the input to the decoder network are passed to a gated recurrent unit (GRU) which applies an input-dependent gating mechanism to modify the hidden state that will be passed on to the subsequent decoder (See Fig. 4.4). The recurrent structure is popular in obstacle motion prediction as it captures temporal dependencies [83] of the AV's interactions with the TVs and prevents the vanishing gradient problem that is prevalent in standard RNNs[21]. Also, the recurrent structure renders the RAID-Net independent of the prediction horizon (N) as decoders share parameters. A linear projection layer lifts the sequence of embedded states to predict the class-1 probabilities of duals corresponding to the collision avoidance constraints with TV i at time $t+1$: $\boldsymbol{p^i_{t+1}} \in \mathbb{R}^M$, where the m-th element corresponds to the class-1 probability for the m-th mode configuration of the TVs. Collecting the $\{\boldsymbol{p^i_{t+1}}\}_{i=1}^V$ from each decoder, we get

$$\boldsymbol{p_t} = [\{\boldsymbol{p^i_{t+1}}\}_{i=1}^V, ..., \{\boldsymbol{p^i_{t+N}}\}_{i=1}^V] \in \mathbb{R}^{n_c} \tag{4.10}$$

Further details of the RAID-Net architecture can be found in [48]

Loss function

The binary cross-entropy loss function for training the RAID-Net, $\ell(\boldsymbol{p}, \tilde{\boldsymbol{\mu}}^\star)$ is

$$\tfrac{-1}{n}\sum_{i=1}^{n}\sum_{s=1}^{n_c}(w_p[\tilde{\boldsymbol{\mu}}_i^\star]_s \log[\boldsymbol{p}_i]_s + (1-[\tilde{\boldsymbol{\mu}}_i^\star]_s)\log(1-[\boldsymbol{p}_i]_s), \tag{4.11}$$

where $[\boldsymbol{p}_i]_s$ and $[\tilde{\boldsymbol{\mu}}^\star{}_i]_s$ are s-th component of the dual class-1 probability prediction and the s-th component of the target class label of the i-th sample, respectively. n is the training batch size and w_p is the weight corresponding to positive class. By choosing w_p appropriately, the learned classifier can conservatively over-predict class 1 and account for a potential imbalance in a dataset (i.e. number of class 0 $\gg$ number of class 1). *This is critical for safety as increasing w_p will reduce false negative classifications of active constraints.*

Hierarchical MPC with RAID-Net

For online deployment, we use the RAID-Net in a hierarchical structure with the low-level MPC planner as visualized in Fig. 4.1 and formalized in Algorithm 2.

The RAID-Net classifier to predict the dual classes is

$$\pi^{\text{RAIDN}}(\tilde{\boldsymbol{\mu}}_t|ob_t) = \begin{cases}[\tilde{\boldsymbol{\mu}}_{\boldsymbol{t}}]_s = 1 & \forall s \in \mathbb{I}_1^{n_c} \text{ if } [\boldsymbol{p_t}]_s \geq 0.5\\ [\tilde{\boldsymbol{\mu}}_{\boldsymbol{t}}]_s = 0 & \text{otherwise}\end{cases} \tag{4.12}$$

where $\boldsymbol{p_t}$ is the RAID-Net output representing the probabilities of the dual classes being 1, given the observation of the environment at time t. Next, we define a set that contains all the indices of constraints that are predicted to be active.

$$\hat{\mathbb{S}} := \{s \in \mathbb{I}_1^{n_c} |\ [\pi^{\text{RAIDN}}(\tilde{\boldsymbol{\mu}}_t|ob_t)]_s = 1\} \tag{4.13}$$

To recover a dual feasible candidate $\hat{\boldsymbol{\mu}}_t$, we solve the reduced linear system via least-squares,

$$[\mathcal{A}_t]_{\hat{\mathbb{S}}\cup xu}\mathcal{Q}_t^{-1}[\mathcal{A}_t]_{\hat{\mathbb{S}}\cup xu}^\top \begin{bmatrix}[\bar{\boldsymbol{\mu}}_t]_{\hat{\mathbb{S}}}\\ \boldsymbol{\eta}_t\end{bmatrix} = [\mathcal{A}_t]_{\hat{\mathbb{S}}\cup xu}\mathcal{Q}_t^{-1}\mathcal{C}_t - [\mathcal{R}_t]_{\hat{\mathbb{S}}\cup xu} \tag{4.14}$$

to obtain the unconstrained minimizer, $\bar{\boldsymbol{\mu}}_t$, of the *reduced* dual problem (by eliminating all duals not in $\hat{\mathbb{S}}$). Then we project $[\bar{\boldsymbol{\mu}}_t]_{\hat{\mathbb{S}}}$ onto the corresponding cones to get $[\hat{\boldsymbol{\mu}}_t]_{\hat{\mathbb{S}}}$ and set the other elements of $\hat{\boldsymbol{\mu}}_t$ to be 0. Finally, we perform the sensitivity-based vehicle and scenario selection from Section 4.4 to get

$$\mathbb{S} := \hat{\mathbb{S}} \bigcap_{\substack{\forall i \in \mathbb{I}_1^V:\\ ||S^i[\hat{\boldsymbol{\mu}}_t]||_2 > \frac{\delta}{D}}} \mathbb{S}^i \bigcap_{\substack{\forall m \in \mathbb{I}_1^{M^V}:\\ ||S_m[\hat{\boldsymbol{\mu}}_t]||_2 > \frac{\delta}{D\cdot V}}} \mathbb{S}_m, \tag{4.15}$$

where $\mathbb{S}^i, \mathbb{S}_m \subset \mathbb{I}_1^{n_c}$ are the index sets of collision avoidance cones corresponding to vehicle i and scenario m, respectively. Thus, we can now define the reduced problem

$$(P_r): \quad \begin{aligned} \min_{\boldsymbol{\theta}_t} \quad & \frac{1}{2}||\mathcal{Q}_t\boldsymbol{\theta}_t||_2^2 + \mathcal{C}_t^\top \boldsymbol{\theta}_t \\ \text{s.t } & [\mathcal{A}_t\boldsymbol{\theta}_t + \mathcal{R}_t]_{\mathbb{S}\cup xu} \in \mathbb{K} := (\bigotimes_{s\in\mathbb{S}} \mathbb{K}_s) \times \mathbb{K}_{xu} \end{aligned} \tag{4.16}$$

The Reduced MPC (ReMPC) is given by the optimal solution of (4.16) as

$$u_t = \pi_{\text{ReMPC}}(\mathcal{A}_t, \mathcal{R}_t, \mathcal{Q}_t, \mathcal{C}_t, \mathbb{S}) = u^\star_{t|t,1}. \tag{4.17}$$

Algorithm 2 Hierarchical MPC (HMPC): Hierarchical Motion Planning with Duality-based Interaction Prediction

---

Require: π^{RAIDN}
  Set the task horizon T
  while $t < T$ **do**
    $ob_t \leftarrow$ Observation from the environment
    $\mathcal{A}_t, \mathcal{R}_t, \mathcal{Q}_t, \mathcal{C}_t \leftarrow ob_t$ ▷ Constructed from observation
    $\hat{\mathbb{S}} \leftarrow$ Computed using (4.13) from $\pi^{\text{RAIDN}}(\tilde{\boldsymbol{\mu}}_t|ob_t)$
    $\mathbb{S} \leftarrow$ Computed from sensitivity criterion (4.15)
    $u_t \leftarrow \pi_{\text{ReMPC}}(\mathcal{A}_t, \mathcal{R}_t, \mathcal{Q}_t, \mathcal{C}_t, \mathbb{S})$
    Apply u_t to system
    $t \leftarrow t + 1$

4.5 Numerical Example: Planning at a Traffic Intersection

Simulation Environment

We created a customized Python simulation environment for unsignalized traffic intersection based on OpenAI gymnasium [14]. The simulation environment has four node zones: *N, S, W, and E* as shown in Fig. 4.5. From each node zone, vehicles are randomly generated in the following order: Firstly, the ego vehicle is placed at the starting node in *W* zone, starting with a random target node (*N* or *E* defining 2 modes for ego vehicle). Note that the *S* node isn't considered a target for the vehicle starting in *W* zone as it leads to minimal interaction between vehicles. Next, V target vehicles are randomly spawned. V is a random integer from the interval [1,V_{max}], where V_{max} is the maximum possible number of target vehicles defined by the user. We arbitrarily set $V_{max} = 3$.

Each of these target vehicles originates from distinct zones (*W, S, and E*) and is assigned a randomly determined target node based on the available modes outlined in Table 4.1. The target vehicle spawning from the *E* zone, serving as cross-traffic for the ego vehicle, is provided with

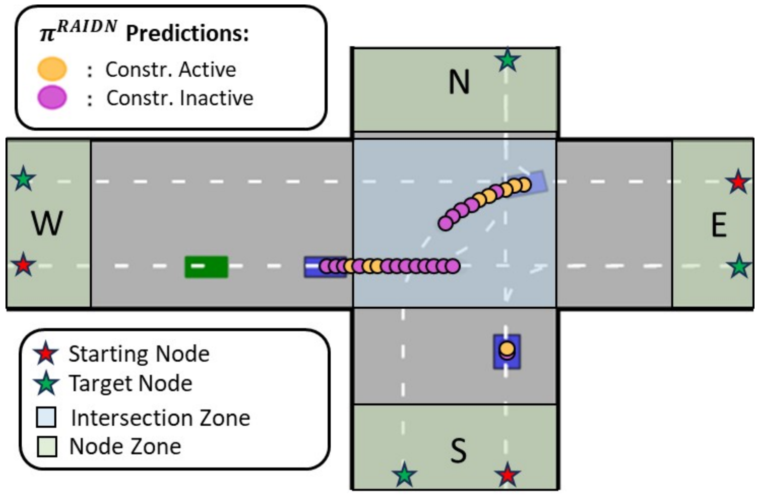


Figure 4.5: An example scene in the custom unsignalized intersection environment. The ego vehicle (green rectangle) is approaching the intersection and interacts with the target vehicles (blue rectangles).The active and inactive constraint predictions from the π^{RAIDN} are depicted as yellow and magenta ellipses, respectively.

four modes for diverse interactive scenarios, while target vehicles starting from other zones have two modes each. The total possible mode configuration of the target vehicles in the scenario is then $M = 16$, which leads to $13 \cdot 16 \cdot 3 = 624$ collision avoidance constraints for (4.7) when $N = 14$.

Given the starting and target node, each vehicle follows a fixed reference trajectory marked with white dashed lines in Fig. 4.5. The vehicle states are $x = [s, v] \in \mathbb{R}^2$, where s and v are displacement along the reference trajectory and velocity, respectively. The target vehicle starting from the *W* node begins 8 meters ahead of the ego vehicle, moving at an initial velocity of 8 m/s. Target vehicles from the *E* zone start with an initial velocity of 8 m/s (7 if slow), while those from the *S* zone begin with 7 m/s.

Table 4.1: Simulator Vehicle's Modes

Start Node	Target Node	# Modes
W	*E*, *N*	2
S	*N*, *E*	2
E	*W*, *W* (slow), *S*, *N*	4

The target vehicle resets to its starting node upon reaching its target node, ending the simulation episode if a collision occurs or when the ego vehicle reaches its target node. Target vehicles employ a Frenet-frame Intelligent Driver Model (fIDM) for safe and human-like acceleration, considering virtual projections of surrounding vehicles on the controlled vehicle's reference trajectory, akin to the generalized Intelligent Driver Model in [54]. The fIDM also features logic gates that prioritize vehicles within intersections, emulating human-like driving behavior. For instance, the approaching vehicle will yield to the surrounding vehicle that is already in the interaction zone as shown in Fig. 4.5. After computing acceleration inputs for the ego vehicle and target vehicles using the relevant motion planner and fIDM, they are simulated forward by a time step (Δt) along their respective reference trajectories using the kinematic bicycle model.

The observation vector available to the ego vehicle from the simulator is represented as $ob_t = [x_t, u_{t-1}, mode_{ev}, \{o_t^i\}_{i=1}^{V_{max}}, \{mode_i\}_{i=1}^{V_{max}}, \{TTC_i\}_{i=0}^{V_{max}}] \in \mathbb{R}^{17}$, where x_t, u_{t-1}, and $mode_{ev}$ denote the ego states, control input at previous time step, and EV's mode, respectively. o_t^i represents the state of the i-th target vehicle, $mode_i$ denotes its mode, and TTC_i denotes the time-to-collision between the ego vehicle and the i-th target vehicle, with $TTC_0 = 0$. When $V < V_{max}$, we spawn dummy vehicles outside the traffic scene for the rest.

RAID-Net Imitation Learning

We use a behavior cloning algorithm [69] to train the RAID-Net to mimic the expert (optimal dual solutions to (4.7)). Data is collected by rolling out trajectories in the aforementioned simulator using the expert policy (4.7) with prediction horizon $N = 14$ and sampling time $\Delta t = 0.2s$. Then, $(ob_t, \tilde{\boldsymbol{\mu}}_{\boldsymbol{t}}^{\star})$ pairs at each time step t are recorded. During the data collection phase, the optimization problem is formulated in CasADi[2] and solved using IPOPT[16] to extract optimal dual solutions. By randomly generating scenarios, 120,315 data points are collected. We split the dataset into training (85$\%$) and test (15$\%$) datasets.

In dual-class classification for predicting active constraints, minimizing false negatives is crucial for safety. While low precision is acceptable, high recall is desired, which represents the model's ability to find all positive classes. To counter dataset imbalance and introduce a positive-class bias in the model, we set $w_p = 4$ in (4.11). Additionally, we address dataset imbalance using a weighted random sampler, assigning a higher weight to the under-represented class to achieve balanced class representation in the training batch.

The RAID-Net is constructed using PyTorch and trained using the training dataset and the loss function (4.11). The parameters used to construct the RAID-Net are reported in [48]. We train the RAID-Net for 3000 epochs using a batch size of 1024 and the Adam with a constant learning rate of 0.001 and $\beta_1, \beta_2 = (0.9, 0.99)$ [62]. The training process takes in total of 4 hours. [2]

Numerical Results

In this section, we first present the imitation learning results of the RAID-Net. Secondly, we compare the performance of our proposed **HMPC** policy 2 to the full MPC policy (4.7) ($N = 14, \Delta t = 0.2$ s for both policies) across 100 randomly generated scenarios in numerical simulations [2]. During the evaluation, the optimization problems are formulated in CasADi, and the MPC SOCPs are solved using Gurobi [34].

[2]Training and experiments were run on a computer with a Intel i9-9900K CPU, 32 GB RAM, and a RTX 2080 Ti GPU.

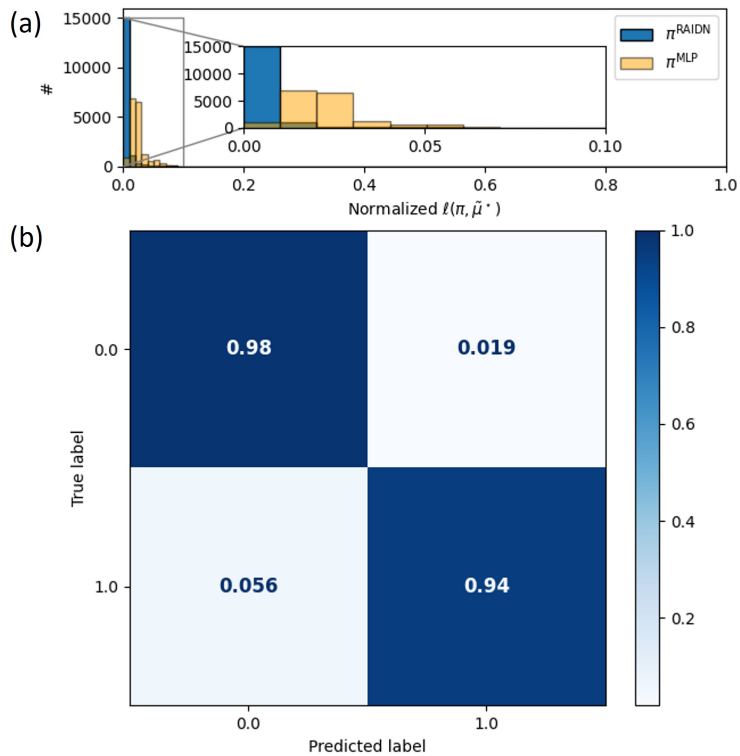


Figure 4.6: RAID-Net evaluation results on a test dataset: (a) histogram of the normalized loss value of RAID-Net versus MLP neural network, (b) normalized confusion matrix of RAID-Net

RAID-Net Evaluation

The trained RAID-Net model was evaluated on the test dataset. We compare the normalized loss statistics of RAID-Net with those of an MLP NN (π^{MLP}) trained with the same configuration as RAID-Net, using the following parameters: 6 layers of 128 hidden units, as shown in Fig. 4.6a. We report the normalized confusion matrix of RAID-Net in Fig. 4.6b. The RAID-Net model adeptly captures intricate interactions between the AV and TVs, surpassing the capabilities of an MLP NN. Further, the RAID-Net model achieved a recall of 0.94 and a precision of 0.44—in other words, it correctly predicts an interactive constraint 44% of the time. RAID-Net training is configured to trade off precision for high recall, since false-positive classifications are not critical to safety. Evaluated on the test dataset, it correctly predicts 98.1% of the total interactive duals (Class 1) with a low 0.056 false negative rate. This performance showcases the proposed architecture's effectiveness in capturing complex interactions between the EV and TVs with multi-modal predictions.

Table 4.2: Performance metrics across all policies for Sec. 4.5.

Performance metric	Full MPC (4.7)	HMPC (2)
Feasibility (%)	98.97	**99.79**
Collision (%)	**0**	4.0
Avg. Constraints Enforced (%)	100	**17.45**
Avg. Solve Time (s)	0.92 ± 0.18	$\mathbf{0.063 \pm 0.073}$
Avg. RAID-Net Query Time (s)	–	$\mathbf{0.013 \pm 0.003}$
Avg. Total Computation Time (s)	0.92 ± 0.18	$\mathbf{0.076 \pm 0.076}$
Avg. normalized task completion time w.r.t. Full MPC	1	**0.91**

Policy Comparison

The performance metrics over 100 runs are recorded in Table 4.2. We record $\%$ of time steps where MPC infeasibility and collision with a target vehicle were detected, along with average $\%$ of constraints that were imposed in the MPC. Furthermore, average times for solving (4.7) and (4.16) as well as the average times for querying the RAID-Net. Finally, the average task completion (reaching the target node) time for **HMPC** normalized by that of the full MPC is reported. The **HMPC** algorithm imposes only the collision avoidance constraints corresponding to the yellow multi-modal prediction of the target vehicle. A video demonstrating the **HMPC** algorithm in multiple scenarios can be found here: https://youtu.be/-pRiOnPb9_c.

Discussion

On average, the solve time, including the RAID-Net query time for the **HMPC** algorithm, is **12 times faster** than that of the full MPC as highlighted in the 6-th row of Table 4.2. This significant acceleration in solving time is attributed to the reduced number of constraints using our proposed algorithm. The **HMPC** algorithm enables real-time applications at frequencies $\geq 10Hz$ by predicting interactions and selectively imposing only safety-critical multi-modal collision avoidance constraints with multiple target vehicles, which is not feasible in real-time otherwise. Additionally, **HMPC** algorithm had slightly higher average feasibility $\%$ which is also attributed to imposing fewer constraints. Considering that only 1.52$\%$ of the total constraints were active in the test dataset, a prediction of 17.45$\%$ for active constraints is a conservative estimation that was deliberately achieved for safety. The **HMPC** algorithm achieved a 0.91 normalized task completion time, indicating a 9$\%$ faster task completion compared to control with the full MPC. The **HMPC**-AV completes the task faster as it is optimistic, imposing fewer constraints, while the full MPC-AV exhibits a more conservative behavior.

Out of 100 runs, the AV collided with a TV four times with **HMPC** compared to zero collisions with full MPC. This resulted from false negative classifications of dual variables, leading **HMPC** to screen out incorrect constraints.

The results highlight the advantages of screening inactive constraints from the MPC formulation for motion planning in highly interactive traffic scenarios to accelerate the computation of MPC. Numerical simulations revealed that, in practice, very few constraints are active in MPC formulation for motion planning with multi-modal collision avoidance constraints. With the proposed

RAID-Net architecture, the model can predict which TVs, and which modes along the prediction horizon interact with the AV given the graph encoding of the scene. By imposing only the relevant, interactive constraints, we achieve over 12 times improvement in solution time while ensuring safety with a high probability.

4.6 Chapter Summary

We proposed a hierarchical approach for scalable real-time MPC for complex, multi-modal traffic scenarios, consisting of 1) RAID-Net, an attention-based RNN architecture that predicts feasible duals of the MPC problem for identifying relevant AV-target vehicle interactions along the MPC horizon using a sensitivity-based criterion, and 2) a reduced SMPC problem that eliminates irrelevant collision avoidance constraints for computational efficiency. Our approach achieves a 12x speed-up in MPC solve times at a simulated traffic intersection. Current drawbacks are the lack of guarantees on the screened collision avoidance constraints and generalizability of RAID-Net to other intersection topologies. We aim to remedy these concerns by further exploiting the duality and convexity properties of (4.7) in Chapter 5.

4.7 Appendix

Constraint Tightening for Reducing Multi-modal State-Input constraints

First, we enforce $O(NMV)$ joint chance constraints

$$\mathbb{P}\left[\frac{-1}{V}\gamma \leq K^i_{k|t,j}(o^i_{k|t,j} - \mathbb{E}[o^i_{k|t,j}]) \leq \frac{1}{V}\gamma\right] \geq 1-\beta \tag{4.18}$$

for some $\beta < \epsilon$ and reactive control authority $\gamma \in \mathbb{R}^{n_u}$ within input constraints. Define the *tightened* state-input constraints $\tilde{\mathbb{XU}}_k = \{(x,u) | F^x_k x + F^u_k u \leq \tilde{f}_k\}$ where

$$\tilde{f}_k = f_k - \max_{\substack{-\gamma\leq z_l\leq\gamma \\ l=t,..,k}} F^x_k \sum_{l=t}^{k}\prod_{v=l}^{k} A_v B_l z_l + F^u_k z_k$$

which can be computed in closed-form using the dual norm[9, Chapter 1]. Let the nominal predicted state for time p be $\bar{x}_{p|t} = \prod_{v=t}^{k} A_v x_t + \sum_{l=t}^{p}\prod_{v=l}^{p} A_v B_l h_{l|t}$. The quantity $\bar{f}_k$ ensures that if $(\bar{x}_{k|t}, h_{k|t}) \in \tilde{\mathbb{XU}}_K$ and $\frac{-1}{V}\gamma \leq K^i_{k|t,j}(o^i_{k|t,j} - \mathbb{E}[o^i_{k|t,j}]) \leq \frac{1}{V}\gamma$, then $(x_{k|t,m}, u_{k|t,m}) \in \mathbb{XU}_k$ for *any* $m \in \mathbb{I}_1^{M^V}$. Then we replace the exponentially many probabilistic state-input constraints in (4.4) using:

$$\begin{aligned}
&\mathbb{P}\left[\frac{-1}{V}\gamma \leq K^i_{k|t,j}(o^i_{k|t,j} - \mathbb{E}[o^i_{k|t,j}]) \leq \frac{1}{V}\gamma\right] \geq 1-\beta,\\
&\mathbb{P}[(\bar{x}_{k+1|t}, h_{k|t}) \notin \tilde{\mathbb{XU}}_k] \leq \epsilon - \beta,\ \forall k \in \mathbb{I}_t^{t+N}, i \in \mathbb{I}_1^V, j \in \mathbb{I}_1^M &(4.19)\\
\Longrightarrow &\mathbb{P}((x_{k+1|t,m}, u_{k|t,m}) \notin \mathbb{XU}_k) \leq \epsilon, \forall m \in \mathbb{I}_1^{M^V}, k \in \mathbb{I}_t^{t+N}. &(4.20)
\end{aligned}$$

Thus, the $\mathcal{O}(NMV)$ state-input constraints (4.19) can be enforced to inner-approximate the $\mathcal{O}(NM^V)$ state-input constraints in (4.20).

Chapter 5

Learning-Guided Scalable MPC using Convex Duality-based Criterion: Safe Variable and Constraint Reduction

Chapter 4 demonstrates that duality-based screening can dramatically reduce computation, but provides no formal guarantee that removed constraints remain satisfied. This chapter addresses that limitation, completing the duality-based screening framework introduced in Chapter 1 with certifiable safety guarantees. We present SHIELD, a hierarchical algorithm that reduces both the decision-variable dimension and the constraint set in ℓ_1-regularized convex programs. From strong convexity and Lagrangian duality, we derive certificates that *safely* discard constraints and decision variables while guaranteeing that all removed constraints remain satisfied and all removed variables are null. To further accelerate the proposed algorithm, we propose a transformer-based deep neural network to guide the dual certificate inference. We validate SHIELD on stochastic model predictive control (SMPC) in complex, multi-modal traffic scenarios, comparing against a full-dimensional SMPC policy. Numerical simulations demonstrate order-of-magnitude computational speedups while preserving feasibility and closed-loop safety, highlighting the practicality of certifiably safe, lightweight MPC in complex driving scenarios.

5.1 Introduction

Deep learning and other data-driven techniques have grown rapidly in modern control, with broad adoption in robotics, power systems, chemical processes, and manufacturing. Learning is commonly used for model identification, uncertainty quantification, controller performance improvement, and accelerating online control computation. This last direction has become particularly active in optimization-based control, especially Model Predictive Control (MPC). At each control step, MPC solves a constrained optimization problem to compute the next input while accounting for dynamics, physical limits, and safety margins. However, this repeated optimization must run at real-time rates—often milliseconds—which becomes challenging as problem complexity grows,

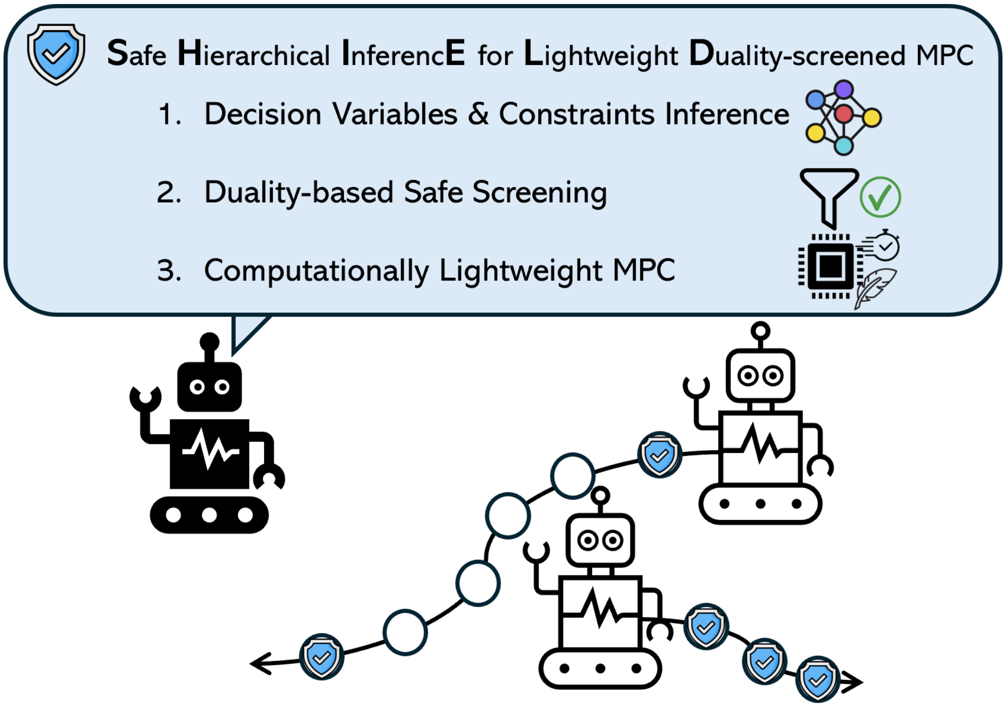


Figure 5.1: An illustration of the Safe Hierarchical Inference for Lightweight Duality-Screened MPC (SHIELD). The controlled robot (black) applies the SHIELD algorithm to infer about which constraints and decision variables are safe to ignore for computationally lightweight model predictive control.

e.g., longer horizons, richer models, and hundreds of inequality constraints. In dynamic, uncertain environments, such computational latency can directly degrade performance or compromise safety.

Recent supervised-learning frameworks address this computational bottleneck via the "learning to optimize" paradigm [82, 3], embedding neural priors into solver pipelines to achieve runtime gains, particularly in large-scale or embedded control systems. These methods exploit structure in MPC solutions and have been studied for convex and mixed-integer convex problems. Broadly, they fall into two approaches trading off speed and optimality. The first learns warm-starts, predicting initial guesses to reduce solver iterations [86, 20, 66]. The second predicts complete or partial solutions, learning mappings from problem parameters to optimizers [101, 80, 18, 19]. Warm-starting is typically more robust to mispredictions and preserves certifiable optimality by solving the full problem; however, in time-varying settings it is difficult to predict warm-starts that consistently reduce iterations. Solution-prediction methods can be fast (via direct prediction or reduced online solves after eliminating variables/constraints), and recent work quantifies suboptimality [101, 80]; nevertheless, mispredictions can cause suboptimality or infeasibility. We therefore target a robust approach for accelerating real-time convex MPC with time-varying constraints, variables, and cost, as in multi-robot coordination, energy-aware fleet control, and autonomous driving.

Building on Chapter 4 and [48], we propose a hierarchical learning-based approach for scalable constrained optimal control in dynamic scenarios. Our approach has two components: (i) an attention-based deep neural network (DNN), trained on time-varying data, that predicts *marginally important variables and constraints* for optimal control at run-time; and (ii) a Lagrange duality-based safety screen that certifies whether inferred variables/constraints can be safely eliminated without violating original constraints. By marginally important, we mean variables/constraints whose presence or absence does not affect the optimal planned trajectory.

The result is a lightweight optimal control problem with fewer decision variables and constraints, enabling fast run-time solves. Rather than relying on *a priori* statistical guarantees for the neural network, we use the duality-based certification algorithm online to maintain robustness under train-to-test distribution shift in time-varying scenarios.

Sec. 5.2 formulates a generic time-varying convex optimal control problem. Sec. 5.3 presents our **S**afe **H**ierarchical **I**nferenc**E** for **L**ightweight, **D**uality-screened (SHIELD) MPC algorithm for safe computational acceleration, and Sec. 5.4 demonstrates its effectiveness in multi-modal traffic MPC.

5.2 Problem Formulation

We consider a convex, finite-horizon optimal control problem that must be solved at a high rate to enable real-time robot control. In a standard MPC setup, at each time step t, the robot solves the following optimization problem:

$$\min_{\mathbf{x}_t, \mathbf{u}_t} \quad f_t^0(\mathbf{x}_t, \mathbf{u}_t) \tag{5.1a}$$

$$\text{s.t.} \quad f_t^i(\mathbf{x}_t, \mathbf{u}_t) \leq 0, \quad \forall i \in \mathbb{I}_1^m \tag{5.1b}$$

$$h_t^i(\mathbf{x}_t, \mathbf{u}_t) = 0, \quad \forall i \in \mathbb{I}_1^p \tag{5.1c}$$

to compute feedback policy $\pi_t(x_t) = u^\star_{t|t}$, where $x_{k|t} \in \mathbb{R}^{n_x}$ and $u_{k|t} \in \mathbb{R}^{n_u}$ are state and input at prediction step k from time t. Let $\mathbf{x}_t$ and $\mathbf{u}t$ stack states and inputs over horizon N, and denote $k_1, \ldots, k_2$ by $\mathbb{I}{k_1}^{k_2}$. We assume f_t^0 is strongly convex in $(\mathbf{x}, \mathbf{u})$. Functions f_t^i in (5.1b) are convex inequality constraints, and h_t^i in (5.1c) encode linear dynamics and initial conditions. (5.1) has m inequality and p equality constraints.

Computing $\pi_t(\cdot)$ by solving (5.1) at high rate becomes challenging at large problem scales, even on modern CPUs. In heterogeneous multi-agent, multi-modal settings, constraints can grow combinatorially with the number of modes and surrounding agents, leading to prohibitive solve times [48, 73]. Likewise, centralized (and some distributed) optimization-based multi-robot controllers scale poorly with the number of robots [37, 53, 59]. When constraints and decision variables grow superlinearly with horizon, agents, or scenario complexity, even fast solvers may not deliver low-latency control.

Our objective is to accelerate computation of $\pi_t(\cdot)$ in real-time while certifying safety, i.e., satisfaction of the original constraints in (5.1). Using certifiable screening, we solve a reduced convex program that provides a feasible solution to the original problem by construction. Certifiability is defined as preserving feasibility of the original convex program (5.1).

5.3 Safe Duality-screened Lightweight Optimal Control

ℓ_1 Regularized Optimization Problem

Without loss of generality, we adopt an affine control policy, $u_{k|t} = \pi_{\boldsymbol{\theta}_t}\big(x_{k|t},\, z_t\big)$, where $u_{k|t}$ is the control at prediction step k given information at time t, $x_{k|t}$ is the predicted state, and z_t collects environment parameters (e.g., obstacle states). We optimize over the policy parameters $\boldsymbol{\theta}_t$. Such parameterizations are standard in MPC; in uncertain multi-modal settings they combine state feedback with affine disturbance feedback (ADF), improving feasibility and reducing conservatism [73]. For open-loop input-sequence optimization, the feedback parameters are zero and $\boldsymbol{\theta}_t$ is the open-loop sequence.

Let the decision variable $\boldsymbol{\theta}_t$ collect the control policy parameters across the prediction horizon. We add N additional equality constraints for $u_{k|t} = \pi_{\boldsymbol{\theta}_t}\big(x_{k|t},\, z_t\big)$ to h_t^i represented by (5.2d) and add l_1 regularization to the cost function:

$$
\begin{aligned}
\min_{\mathbf{x}_t^i, \mathbf{u}_t^i, \boldsymbol{\theta}_t} \quad & f_t^0(\mathbf{x}_t, \mathbf{u}_t) + \lambda \|S\boldsymbol{\theta}_t\|_1 && \text{(5.2a)}\\
\text{s.t.} \quad & \bar{f}_t^i(\mathbf{x}_t, \mathbf{u}_t) \le 0, \quad \forall i \in \mathbb{I}_1^{m-c} && \text{(5.2b)}\\
& \tilde{f}_t^i(\mathbf{x}_t, \mathbf{u}_t) \le 0, \quad \forall i \in \mathbb{I}_1^{c} && \text{(5.2c)}\\
& h_t^i(\mathbf{x}_t, \mathbf{u}_t) = 0, \quad \forall i \in \mathbb{I}_1^{p+N} && \text{(5.2d)}
\end{aligned}
$$

where S is a user-defined selection matrix to choose which control policy parameters to sparsify. The l_1 penalty promotes sparsity in the decision variables and feature selection, which enables optimization over decision variables in control-task importance [17]. We partition (5.1b) into (i) $\bar{f}_t^i$, *immutable*—always enforced (physical limits, non-negotiable safety: actuator/state limits; friction/flight envelopes; traffic rules/road bounds) and (ii) $\tilde{f}_t^i$, *screenable*—relevant only when near-active (collision avoidance with distant agents/obstacles; keep-out/no-fly regions irrelevant to current path). For instance, constraint $\tilde{f}_t^i \le 0(\mathbf{x}_t, \mathbf{u}_t)$ can be screened out (i.e. removed from optimization problem solved at time t) if a strict margin $\tilde{f}_t^i(\mathbf{x}_t, \mathbf{u}_t) < 0$ for i at the optimal solution is certified.

This provides an opportunity to reduce the c constraints where the number of inequality constraints $m \ge c \gg 1$ limits the fast computation of (5.2).

For a linear system, given the initial condition on $x_{t|t}$, and the affine control policy parameterization $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, and composing the functions with the affine map $\boldsymbol{\theta}_t \mapsto (\mathbf{x}_t(\boldsymbol{\theta}_t), \mathbf{u}_t(\boldsymbol{\theta}_t))$, we can rewrite

(5.2) as follows via recursive substitution:

$$
\begin{aligned}
\min_{\boldsymbol{\theta}_t} \quad & f_t^0(\boldsymbol{\theta}_t) + \lambda \|S\boldsymbol{\theta}_t\|_1 && \text{(5.3a)}\\
\text{s.t.} \quad & \bar{f}_i(\boldsymbol{\theta}_t) \le 0, \quad \forall i \in \mathbb{I}_1^{m-c} && \text{(5.3b)}\\
& \tilde{f}_t^i(\boldsymbol{\theta}_t) \le -\zeta, \quad \forall i \in \mathbb{I}_1^c && \text{(5.3c)}\\
& h_t^i(\boldsymbol{\theta}_t) = 0, \quad \forall i \in \mathbb{I}_1^p, && \text{(5.3d)}
\end{aligned}
$$

where $\zeta > 0$ is a *safety-margin* that we desire on constraint satisfaction. We tighten the *screenable* constraints to enable a design of screening condition that guarantees at most ζ violation of the tightened constraint, ensuring $\tilde{f}_t^i(\boldsymbol{\theta}_t) \le 0$.

Assumption 1. *$f_t^0(\boldsymbol{\theta}_t) \in C^2(\mathbb{R}^n)$ is $\underline{\sigma}$-strongly convex w.r.t. $\boldsymbol{\theta}_t$ and has $\bar{\sigma}$-Lipschitz continuous gradient.*

Problem (5.3) after the epigraph reformulation is

$$
\begin{aligned}
\min_{\boldsymbol{\theta}_t, s} \quad & f_t^0(\boldsymbol{\theta}_t) + \lambda \mathbf{1}^\top s && \text{(5.4a)}\\
\text{s.t.} \quad & \bar{f}_t^i(\boldsymbol{\theta}_t) \le 0, \quad \forall i \in \mathbb{I}_1^{m-c} && \text{(5.4b)}\\
& \tilde{f}_t^i(\boldsymbol{\theta}_t) \le -\zeta, \quad \forall i \in \mathbb{I}_1^c && \text{(5.4c)}\\
& h_t^i(\boldsymbol{\theta}_t) = 0, \quad \forall i \in \mathbb{I}_1^p && \text{(5.4d)}\\
& S\boldsymbol{\theta}_t - s \le 0, -S\boldsymbol{\theta}_t - s \le 0, s \ge 0. && \text{(5.4e)}
\end{aligned}
$$

Let the dual variables $\mu_i, \eta_i, \nu_i, g_1, g_2, \gamma$ correspond to $\tilde{f}_t^i, \bar{f}_t^i, h_t^i$, and (5.4e), respectively. Let $\Phi := f_t^0 + \sum \mu_i(\tilde{f}_t^i + \zeta) + \sum \eta_i \bar{f}_t^i + \sum \nu_i h_t^i$. The dual problem is given by

$$
\min_{\mu \ge 0, \eta \ge 0, \nu, \|g\|_\infty \le \lambda} \quad -\inf_{\boldsymbol{\theta}_t} \{\Phi(\boldsymbol{\theta}_t; \mu, \eta, \nu) + (S^\top g)^\top \boldsymbol{\theta}_t\} \tag{5.5}
$$

For well-defined convex optimization problems, Φ can be computed explicitly. For example, if f_t^0 is quadratic and $\tilde{f}_t^i, \bar{f}_t^i, h_t^i$ are affine in $\boldsymbol{\theta}_t$ and (5.5) becomes a quadratic program.

The optimal dual solutions $(\boldsymbol{\theta}_t^\star, \mu^\star, \eta^\star, \nu^\star, g^\star)$ are given by the KKT conditions:

$$
\begin{aligned}
&\mathbf{0} \in \partial f_t^0(\boldsymbol{\theta}_t^\star) + \sum_i \mu_i^\star \partial \tilde{f}_t^i(\boldsymbol{\theta}_t^\star) + \sum_i \eta_i^\star \partial \bar{f}_t^i(\boldsymbol{\theta}_t^\star) && \text{(5.6)}\\
&+ \sum_i \nu_i^\star \partial h_t^i(\boldsymbol{\theta}_t^\star) + S^\top g^\star, g^\star \in \lambda \partial \|S\boldsymbol{\theta}_t^\star\|_1 \\
&\tilde{f}_t^i(\boldsymbol{\theta}_t^\star) \le -\zeta, \ \mu_i^\star \ge 0, \ \forall i \in \mathbb{I}_1^c, h_t^i(\boldsymbol{\theta}_t^\star) = 0, \ \forall i \in \mathbb{I}_1^p \\
&\bar{f}_t^i(\boldsymbol{\theta}_t^\star) \le 0, \ \eta_i^\star \ge 0, \ \forall i \in \mathbb{I}_1^{m-c}, \ \|g^\star\|_\infty \le \lambda \\
&\mu_i^\star(\tilde{f}_t^i(\boldsymbol{\theta}_t^\star) + \zeta) = 0, \ \forall i \in \mathbb{I}_1^c, \ \eta_i^\star \bar{f}_t^i(\boldsymbol{\theta}_t^\star) = 0, \ \forall i \in \mathbb{I}_1^{m-c} \\
&{g^\star}^\top S\boldsymbol{\theta}_t^\star = \lambda \|S\boldsymbol{\theta}_t^\star\|_1
\end{aligned}
$$

Observe from the last two equations that

1. If $\mu_i^\star = 0$, then $\tilde{f}_t^i(\boldsymbol{\theta}_t^\star) < -\zeta$ and this constraint can be removed from (5.3),
2. If $\|g_i^\star\|_\infty < \lambda$, then $[\mathcal{S}\boldsymbol{\theta}_t^\star]_i = 0$ and the variables corresponding to it can be removed from (5.4).

where g_i is the dual variable corresponding to the i-th components of the ℓ_1 regularization term (5.4a) and $[S\boldsymbol{\theta}_t^\star]_i$ denotes the i-th component of $S\boldsymbol{\theta}_t^\star$. As small nonzeros from tolerances/degeneracy make a hard zero-test brittle, we seek for more robust and useful constraint screening condition.

Constraint Screening Condition

We utilize perturbation/sensitivity analysis of convex programs to derive constraint-removal conditions. Suppose we obtain solution $\bar{\boldsymbol{\theta}}_t$ after removing constraint $\tilde{f}_t^i$. The violation of constraint $\tilde{f}_t^i$, denoted δ_i, is analogous to constraint perturbations in [13]. Let $p_t(\delta_i)$ be the objective value of (5.15) at $\bar{\boldsymbol{\theta}}_t$, and $p_t(0)$ the optimal value of original problem (5.4).

Assumption 2. *Strong duality holds for* (5.4) *under Slater's condition [13].*

By the global sensitivity result in [13, Ch. 5],

$$p_t(0) - p_t(\delta_i) \le \delta_i \mu_i^\star, \tag{5.7}$$

which quantifies how the optimal value changes when relaxing/removing constraint i.

Assumption 3. *Each constraint $\tilde{f}_t^i(\cdot)$ is L_i-Lipschitz.*

Even if the reduced convex program (5.15) obtained by removing $\tilde{f}_t^i(\cdot)$ yields a solution that is infeasible for the tightened problem (5.4), it remains within the safety margin ζ and satisfies the original constraints (5.2c), as formalized below.

Proposition 1. *Let Assumptions 1–3 hold and the solution to* (5.2) $\boldsymbol{\theta}_t^\star$ *be strictly feasible with margin $\zeta \ge \min_i\{-\tilde{f}_t^i(\boldsymbol{\theta}_t^\star)\} > 0$ for all i. Denote $f_t^0 =$* (5.4a) *and define*

$$\varepsilon_{\text{crit}} \; := \; \frac{\sigma}{2} \min_i \left(\frac{\zeta}{L_i}\right)^2.$$

Then, any $\boldsymbol{\theta}_t$ with $|f_t^0(\boldsymbol{\theta}_t) - f_t^0(\boldsymbol{\theta}_t^\star)| \le \varepsilon_{\text{crit}}$ is feasible for the original constraints (5.2c)*, i.e., $\tilde{f}_t^i(\boldsymbol{\theta}_t) \le 0$ for all i.*

Proof. Let $|f_t^0(\boldsymbol{\theta}_t) - f_t^0(\boldsymbol{\theta}_t^\star)| \le \epsilon$ and by $\underline{\sigma}$-strong convexity,

$$f_t^0(\boldsymbol{\theta}_t) - f_t^0(\boldsymbol{\theta}_t^\star) \; \ge \; \frac{\sigma}{2}\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^\star\|^2 \quad \Rightarrow \quad \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^\star\| \; \le \; \sqrt{\tfrac{2\epsilon}{\underline{\sigma}}}.$$

Since $\tilde{f}_t^i$ is L_i-Lipschitz,

$$\tilde{f}_t^i(\boldsymbol{\theta}_t) \; \le \; \tilde{f}_t^i(\boldsymbol{\theta}_t^\star) + L_i\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^\star\| \; \le \; -\zeta + L_i\sqrt{\tfrac{2\epsilon}{\underline{\sigma}}} \le 0.$$

Thus, $\tilde{f}_t^i(\boldsymbol{\theta}_t) \le 0$ holds whenever $\epsilon \le \frac{\underline{\sigma}\zeta^2}{2L_i^2}$ and enforcing this for all i gives $\epsilon \le \epsilon_{\text{crit}}$, which proves feasibility. $\square$

Corollary 1. *From Proposition 1 and* (5.7)*, we set $\boldsymbol{\delta}_i$ to ζ and get $|p_t(0) - p_t(\zeta)| \leq \zeta \|\mu_i^\star\|_2$ where the solution to the ζ-perturbed primal is $\bar{\boldsymbol{\theta}}_t$. Thus, if $\|\mu_i^\star\|_2 \leq \epsilon/\zeta$ (i.e., perturbation is ζ and the change in the optimal cost is bounded by ϵ) and $\epsilon \leq \frac{\sigma\zeta^2}{2} \min_i \frac{1}{L_i^2}$, then the i-th constraint can be safely removed and the original constraint $\tilde{f}_t^i(\bar{\boldsymbol{\theta}}_t) \leq 0$ is satisfied.*

Because the derived screening conditions in Sec. 5.3 apply to optimal duals, we generalize the inequalities and Cor. 1 to any feasible dual via a dual optimality-gap bound, enabling certifiable safe screening with an approximate dual computed from DNN class predictions and a linear equation solve.

Duality-based Safe Screening Certification

We derive certifiably *safe* screening conditions for constraints and decision variables in (5.4) by leveraging strong convexity of the primal and strong duality.

Proposition 2. *Under Assumption 1, the dual objective $d(\mu, \eta, \nu, g)$ in* (5.5) *is $\bar{\rho}$-smooth and $\underline{\rho}$-strongly convex over the feasible set $\mathcal{Y}$* [1]*.*

In the quadratic-program case where the primal cost f_t^0 has Hessian $Q_t \succ 0$, the dual Hessian is Q_t^{-1}, and

$$\bar{\rho} = \lambda_{\max}(Q_t^{-1}), \qquad \underline{\rho} = \lambda_{\min}(Q_t^{-1}).$$

For brevity, we stack the dual variables as $y := [\mu, \eta, \nu, g]$.

Definition 1. *The projected gradient for the dual is*

$$\nabla^\dagger d(y) = y - [y - \nabla d(y)]_{\mathcal{Y}}, \ \forall y \in \mathcal{Y}$$

where $\mathcal{Y}$ is the dual feasible set, which is nonempty, closed, and convex from (5.5)*. The $[x]_{\mathcal{Y}}$ is a convex projection operator.*

The gradient of the dual cost can be computed explicitly offline and may be a parametric function, which can be quickly evaluated online. Note, at the optimal dual solution $y^\star$, $\nabla^\dagger d(y^\star) = 0$ [93]. Then, the following inequality estimates the distance of a feasible dual y from optimality [93]:

$$\|y - y^\star\|_2 \leq \frac{1 + \bar{\rho}}{\underline{\rho}} ||\nabla^\dagger d(y)||_2. \tag{5.8}$$

Since $\nabla^\dagger d(y^\star) = 0$, the bound (5.8) is tight. Define

$$gap(\mu, \eta, \nu, g) := \frac{1 + \bar{\rho}}{\underline{\rho}} ||\nabla^\dagger d(\mu, \eta, \nu, g)||_2. \tag{5.9}$$

[1] We use the minimization form of the dual in (5.5) (i.e., the negative of the standard concave dual), hence convexity.

Theorem 1. *Given a feasible dual solution* (μ, η, ν, g) *to* (5.5),

1. *If* $\|\mu_i\|_2 + \mathit{gap}(\mu, \eta, \nu, g) \leq \epsilon/\zeta$ *and* $\epsilon \leq \frac{\sigma\zeta^2}{2} \min_i \frac{1}{L_i^2}$, *then* $\tilde{f}_t^i$ *can be safely removed from* (5.4).

2. *If* $\|g_i\|_\infty + \mathit{gap}(\mu, \eta, \nu, g) < \lambda$, *then the variables* $[S\boldsymbol{\theta}_t]_i$ *comprising the policy parameters are zero and thus can be safely removed from* (5.4).

Proof. Statement (1) follows from Corollary 1:

$$\begin{aligned}\|\mu_i^\star\|_2 &\leq \|\mu_i\|_2 + \|\mu_i^\star - \mu_i\|_2 \\ &\leq \|\mu_i\|_2 + \mathit{gap}(\mu, \eta, \nu, g) < \epsilon/\zeta\end{aligned}$$

and if $\frac{\sigma\zeta^2}{2} \min_i \frac{1}{L_i^2}$, then $\tilde{f}_t^i$ can be *safely* removed. If $\epsilon \leq \|g_i\|_\infty + \text{gap}(\mu, \eta, \nu, g) < \lambda$ some $i \in \mathbb{I}_1^{|S\boldsymbol{\theta}_t|}$, then we have

$$\begin{aligned}\|g_i^\star\|_\infty &\leq \|g_i\|_\infty + \|g_i^\star - g_i\|_2 \\ &\leq \|g_i\|_\infty + \mathit{gap}(\mu, \eta, \nu, g) < \lambda\end{aligned}$$

and consequently, the variables $[S\boldsymbol{\theta}_t]_i = 0$ from (5.6) and can be *safely* removed from (5.4). □

SHIELD Algorithm

SHIELD (Safe Hierarchical Inference for Lightweight Duality-screened MPC) is a two-stage algorithm that accelerates the solution of (5.4) by safely eliminating inactive constraints and decision variables at each time step. The first stage estimates a feasible dual solution via a neural-network classifier followed by a reduced unconstrained dual solve and projection. The second stage applies the safe screening theorem (Theorem 1) using the dual optimality-gap bound (5.9) to certify which primal variables and constraints can be pruned, and solves the resulting reduced problem (5.15). The full procedure is summarized in Algorithm 3.

Feasible Dual Solution Estimation

Given any feasible dual solution $(\hat{\mu}, \hat{\eta}, \hat{\nu}, \hat{g})$, we apply Theorem 1 to *safely* eliminate variables and constraints using the dual optimality-gap upper bound in (5.9). When (5.4) is large, directly solving the dual (5.5) can be expensive; instead, we minimize the *unconstrained* dual objective and project the minimizer onto the convex dual-feasible set of (5.5) to obtain a feasible dual estimate. We then use (5.9) together with Theorem 1 to certify when this estimate is reliable enough to prune variables and constraints. Even this unconstrained solve can be costly at scale (Sec. 5.4). To further accelerate, we use a neural-network classifier to preselect a subset of dual variables, yielding a reduced-dimensional dual problem.

The dual classifier $\pi_\zeta^{\text{class}}(\mathbf{z}_t)$, specific to a choice of ζ, predicts the dual class $(\tilde{\mu}, \tilde{g})$ from features $\mathbf{z}_t$ encoding the environment and parameters of (5.4). The feature design, architecture, and training procedure are problem-dependent. The resulting class vector categorizes the i-th dual as follows:

$$\mu_i = \begin{cases} > 0, & \tilde{\mu}_i = 1, \\ 0, & \tilde{\mu}_i = 0, \end{cases} \quad \|g_i\|_\infty = \begin{cases} \lambda, & \tilde{g}_i = 1, \\ < \lambda, & \tilde{g}_i = 0. \end{cases} \tag{5.10}$$

Using the classifier outputs, we reduce the dual by selecting only $\tilde{\mu}_i = 1$ and $\tilde{g}_i = 0$ entries. Let S^r_μ and S^r_g be the corresponding selection matrices; the reduced variables are

$$\mu^r = S^r_\mu \, \mu, \tag{5.11a}$$

$$g^r = S^r_g \, g. \tag{5.11b}$$

After solving the reduced unconstrained dual for (μ^r, η, ν, g^r), we lift (μ^r, g^r) to full dimension by inserting zeros at indices i with $\tilde{\mu}_i = 0$ and λ at indices with $\tilde{g}_i = 1$, consistent with (5.10). Finally, we project the full augmented solution onto the convex, closed dual-feasible set to obtain $(\hat{\mu}, \hat{\eta}, \hat{\nu}, \hat{g})$.

Reduced ℓ_1 Regularized Convex Optimization

Given the feasible dual estimate $(\hat{\mu}, \hat{\eta}, \hat{\nu}, \hat{g})$ and dual class predictions $(\tilde{\mu}, \tilde{g})$, we apply Theorem 1 to *safely* prune variables and constraints. Let $n_\theta := \dim(\boldsymbol{\theta}_t)$ and

$$\mathcal{I} = \{ i \in \mathbb{I}_1^{n_\theta} \mid \|\hat{g}_i\|_\infty + \text{gap}(\hat{\mu}, \hat{\eta}, \hat{\nu}, \hat{g}) < \lambda \wedge \tilde{g}_i = 0 \}, \tag{5.12}$$

which is the index set of variables satisfying the screening condition in Theorem 1.

Let $r := |\mathcal{I}|$. We write $\bar{\boldsymbol{\theta}}_t = S_\mathcal{I} \boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_t = E_\mathcal{I} \bar{\boldsymbol{\theta}}_t$ where

$$(S_\mathcal{I})_{k,i} = \begin{cases} 1, & i = \mathcal{I}(k), \\ 0, & \text{otherwise}, \end{cases} \qquad S_\mathcal{I} \in \{0,1\}^{r \times n_\theta}, \tag{5.13}$$

and the embedding matrix

$$E_\mathcal{I} := S_\mathcal{I}^\top \in \{0,1\}^{n_\theta \times r}. \tag{5.14}$$

The reduced optimization over $\bar{\boldsymbol{\theta}}_t \in \mathbb{R}^r$ is

$$\min_{\bar{\boldsymbol{\theta}}_t} \quad f_t^0(E_\mathcal{I} \bar{\boldsymbol{\theta}}_t) + \lambda \|S E_\mathcal{I} \bar{\boldsymbol{\theta}}_t\|_1 \tag{5.15a}$$

$$\text{s.t.} \quad \bar{f}_t^i(E_\mathcal{I} \bar{\boldsymbol{\theta}}_t) \leq 0, \quad \forall i \in \mathbb{I}_1^{m-c} \tag{5.15b}$$

$$\tilde{f}_t^i(E_\mathcal{I} \bar{\boldsymbol{\theta}}_t) \leq -\zeta, \quad \forall i \in \mathcal{K} \tag{5.15c}$$

$$h_t^i(E_\mathcal{I} \bar{\boldsymbol{\theta}}_t) = 0, \quad \forall i \in \mathbb{I}_1^p, \tag{5.15d}$$

where the reduced optimization problem includes only the relevant decision variables and constraints denoted by $\mathcal{I}$ and

$$\mathcal{K} = \{ i \in \mathbb{I}_1^c \mid \|\mu_i\|_2 + gap(\hat{\mu}, \hat{\eta}, \hat{\nu}, \hat{g}) \leq \epsilon/\zeta \wedge \tilde{\mu}_i = 0 \}. \tag{5.16}$$

Algorithm 3 SHIELD: Safe Hierarchical Inference for Lightweight Duality-screened MPC

Require: (5.4), (5.5), and $\pi_\zeta^{\text{class}}, \epsilon, \zeta$.

1: $(\tilde{\mu}, \tilde{g}) \leftarrow \pi_\zeta^{\text{class}}(\mathbf{z}_t)$ [Dual Classification]
2: Construct S_μ^r and S_g^r using (5.10) [Selection Matrix]
3: $(\mu^r, g^r) \leftarrow (S_\mu^r \mu,\ S_g^r g)$ using (5.11) [Dual Sparsification]
4: Solve the *unconstrained* version of (5.5) in (μ^r, η, ν, g^r) and lift (μ^r, g^r) to full dimension via (5.11) [Linear Eq. Solve]
5: $(\hat{\mu}, \hat{\eta}, \hat{\nu}, \hat{g}) \leftarrow [(\mu, \eta, \nu, g)]_{\mu \geq 0,\ \eta \geq 0,\ \|g\|_\infty \leq \lambda}$ [Projection]
6: $\mathcal{I} \leftarrow (5.12)$, $\mathcal{K} \leftarrow (5.16)$, $E_\mathcal{I} \leftarrow (5.14)$
7: Formulate (5.15) using $E_\mathcal{I}$, $\mathcal{K}$, and (5.4) [Reduced Problem]
8: **Output:** $\bar{\boldsymbol{\theta}}_t^\star \leftarrow$ Solve (5.15)

5.4 Numerical Example: Stochastic MPC with Multi-modal Predictions

In this section, we implement Algorithm 3 to accelerate a convex SMPC reformulation with multi-modal, uncertain predictions of surrounding vehicles. As the numbers of modes (M) and vehicles (V) increase, constraints and decision variables scale combinatorially [73, 48]. We use $N = 14$, $M = 2$, $V = 3$, and $\Delta t = 0.1\,\text{s}$, yielding 312 collision-avoidance constraints and 234 decision variables, challenging high-frequency computation. In simulation, we compare two policies with the same parameters: (i) *Full MPC*—solving the original convex program (5.4); and (ii) *Reduced MPC*—solving the screened/reduced program (5.15). We implement and evaluate our policies in the nuPlan simulator [35].

Multi-modal Trajectory Prediction Model

The open-source implementation of Wayformer [75], an attention-based multi-modal trajectory prediction model proposed by Waymo, was trained using the nuPlan Mini v1.2 dataset [35] using the UniTraj framework [29].

ℓ_1 Regularized Convex MPC Formulation

We add the ℓ_1 regularization term to the second-order cone program (SOCP) from [48, Eq. (6)] to promote decision variable sparsity and enable screening. Given the multi-modal prediction of V vehicles, the resulting SOCP over $\boldsymbol{\theta}_t$ is

$$\begin{aligned} \min_{\boldsymbol{\theta}_t} \quad & \frac{1}{2}||\mathcal{Q}_t\boldsymbol{\theta}_t||_2^2 + \mathcal{C}_t^\top\boldsymbol{\theta}_t + \lambda \sum_{k\in\mathbb{I}_{t+1}^{t+N},\, i\in\mathbb{I}_1^V} ||\mathcal{S}_{k|t}^i\boldsymbol{\theta}_t||_1 \\ \text{s.t}\; & \mathcal{A}_t\boldsymbol{\theta}_t + \mathcal{R}_t \in \mathbb{K} := (\bigotimes_{s=1}^{n_c}\mathbb{K}_s) \times \mathbb{K}_{xu} \end{aligned} \tag{5.17}$$

where $\mathcal{A}_t, \mathcal{R}_t, \mathcal{Q}_t, \mathcal{C}_t$ denote the prediction models and constraints; $\mathcal{S}_{k|t}^i$ is the selection matrix that collects the affine disturbance feedback gain terms in $\boldsymbol{\theta}_t$ [73]. $n_c = \mathcal{O}(N \cdot V \cdot M^V)$, $\mathcal{Q}_t \succ 0$. The $\mathbb{K}_s$ (for all $s \in \mathbb{I}_1^{n_c}$) are cones associated with collision-avoidance (*screenable*) constraints, while $\mathbb{K}_{xu}$ is the cone for all state–input (*immutable*) constraints. See [48] for further details. The dual SOCP of (5.17) is

$$\tfrac{1}{2}\Big\|\mathcal{Q}_t^{-1}\Big(\mathcal{A}_t^\top \begin{bmatrix}\boldsymbol{\mu}_t\\ \boldsymbol{\eta}_t\end{bmatrix} - \mathcal{C}_t - \lambda\sum_{k=t+1}^{t+N}\sum_{i=1}^{V}\mathcal{S}_{k|t}^i\, g_{k|t}^i\Big)\Big\|_2^2 + \mathcal{R}_t^\top \begin{bmatrix}\boldsymbol{\mu}_t\\ \boldsymbol{\eta}_t\end{bmatrix} \tag{5.18}$$

$$\begin{aligned} \min_{\boldsymbol{\mu}_t,\boldsymbol{\eta}_t,\{g_{k|t}^i\}} \quad & (5.18) \\ \text{s.t.}\;\; & \boldsymbol{\mu}_t \in \bigotimes_{s=1}^{n_c}\mathbb{K}_s^*, \quad \boldsymbol{\eta}_t \in \mathbb{K}_{xu}^*, \\ & \|g_{k|t}^i\|_\infty \le \lambda,\ \forall\, k \in \mathbb{I}_{t+1}^{t+N}, i \in \mathbb{I}_1^V. \end{aligned} \tag{5.19}$$

Dual Classifier: Training Data Collection

We use nuPlan [35] to simulate traffic scenarios and collect training data as follows: First, given the scene encoding of road agent history, traffic lights, and a vectorized map, Wayformer [75] forecasts the multi-modal trajectory of surrounding agents. Then, we formulate (5.17) in CasADi [2], replacing $\mathbb{K}_s$ with ζ-tightened cones ($\zeta = 0.5$) and solve using Gurobi [34]. The feature is $\mathbf{z}_t = \{\mathbf{o}_t^i - \mathbf{x}_t^{EV}\}_{i=1}^V$,where $\mathbf{o}_t^i$ denotes the i-th vehicle's multi-modal predictions (top M modes over horizon N) and $\mathbf{x}_t^{EV}$ is the ego planned trajectory. At each time t, we log features $\mathbf{z}_t$ together with the optimal dual variables, and convert them into class labels $(\tilde{\mu}_t^\star, \tilde{g}_t^\star)$. We collect over $33{,}000$ (features, labels) pairs across 300 scenarios from the nuPlan Mini v1.2 dataset [35], and split the data 85 / 15% for training and evaluation, respectively.

Dual Classifier: Architecture Design

The classifier uses a Transformer-style [91], decoder-only architecture *without positional encodings* to preserve permutation equivariance to vehicle ordering. Input features $\mathbf{z}_t$ are linearly embedded and processed by multi-head attention blocks with layer normalization, residual connections, and feedforward layers capture inter-vehicle interactions. Decoder is unrolled for N steps to produce class-1 probabilities (as in RAID-Net [48]), modeling temporal dependencies via recurrence. Training uses weighted binary cross-entropy with AdamW [62] optimizer and cosine-annealing schedule [48].

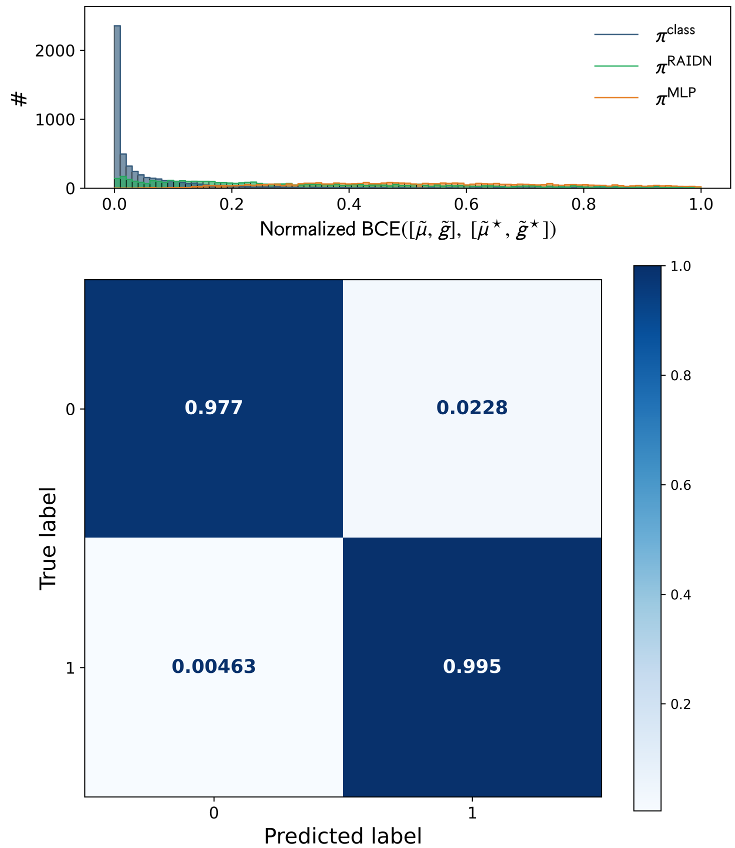


Figure 5.2: Per-sample normalized binary cross-entropy (top) and normalized confusion matrix (bottom) for the (μ, g)-dual classification, evaluated on the evaluation data split. We compare our classifier π_ζ^{class} with an MLP π_ζ^{MLP} and RAID-Net π_ζ^{RAIDN} [48].

Numerical Results

Classifier Evaluation

We evaluate on the *evaluation split*, comparing normalized BCE of $\pi^{\text{class}}\zeta$ (1.1M) against an MLP $\pi^{\text{MLP}}\zeta$ (1.4M) and RAID-Net $\pi^{\text{RAIDN}}\zeta$ (4.6M) in Fig. 5.2. $\pi^{\text{class}}\zeta$ attains lowest normalized BCE. Its confusion matrix (Fig. 5.2) shows recall 0.99, precision 0.97, and accuracy 0.96, indicating few false negatives and improved interaction modeling over baselines under multi-modal predictions.

Closed-loop Simulation

In nuPlan [35], we evaluate *Full MPC* (5.17) (not ζ tightened) and *Reduced MPC* (Alg. 3) with $\epsilon = 0.01$, a conservative underestimate of ϵ_{crit} in Proposition 1. We run 25,000 solves across

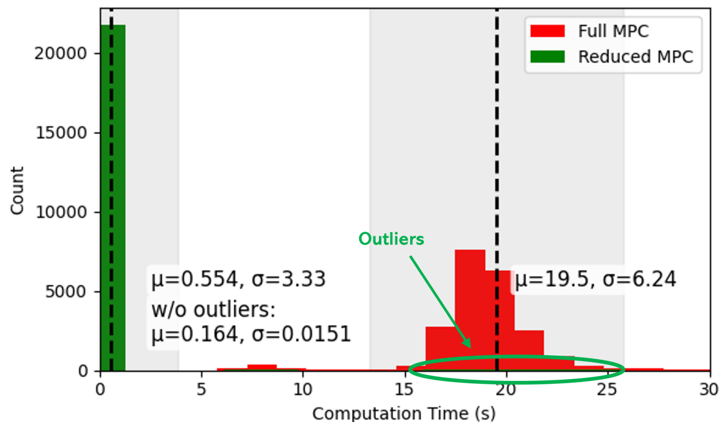


Figure 5.3: Distributions of total computation times for *Full MPC* (5.17) and *Reduced MPC* (Alg. 3) over 25,000 nuPlan steps (SMPC with multi-modal prediction). Shaded bands show $\pm 1\sigma$. The circled region contains outliers for *Reduced MPC* computation time ($\sim 6\%$ of total count).

$150+$ traffic scenarios (e.g., lane merges, unprotected left turns). Table 5.1 reports feasibility, collision rate, average fraction of constraints enforced (constraint keep-rate in *Reduced MPC*), and average fraction of ADF gains kept. We also report solve times and auxiliary runtimes: classifier inference, CasADi parametric calls (dual construction/solve), optimal-dual estimation, and duality-gap computation in SHIELD. These auxiliaries are not yet optimized and can be reduced further. All MPC formulations evaluated in this chapter, both the full and reduced problems, are warm-started using the shifted solution from the previous time step. The computational gains reported therefore reflect the benefit of constraint and variable screening beyond what warm-starting alone provides: the full MPC baseline already benefits from warm-starting, yet its solve times remain substantially higher than the screened formulation, confirming that the bottleneck is problem size rather than initialization quality.

As summarized in Table 5.1, both policies remained feasible and incurred zero collisions. Under the proposed method, the *Reduced MPC* enforced 6.80% of collision-avoidance constraints on average while retaining 3.19% of the ADF gains. Average solve time decreased to 0.498 ± 3.33 s for the *Reduced MPC*, compared to 19.53 ± 6.24 s for the *Full MPC*. The large variance is driven by rare outliers ($\sim 6\%$) where the dual estimate prevents screening and the solve reverts to near-full size, which may happen when the RAID-Net predictions are inaccurate. In a small fraction of time steps, the screening procedure removes few or no constraints, resulting in solve times comparable to the full MPC. These cases arise primarily when the ego vehicle is in close proximity to multiple interacting agents simultaneously, such that many scenario-indexed collision avoidance constraints are simultaneously active or near-active. In such configurations, the dual gap bound is insufficiently tight to certify inactivity, because the true optimal dual variables associated with these constraints are large. Geometrically, these correspond to states where the ego vehicle's fea-

Table 5.1: Closed-loop Simulation Results

Performance Metric	**Full MPC** (5.17)	**Reduced MPC** **Alg. 3**
Feasibility (%)	100	100
Collision (%)	0	0
Avg. Constraints Enforced (%)	100	$\mathbf{6.80 \pm 16.44}$
Avg. ADF Gain Kept (%)	100	$\mathbf{3.19 \pm 13.81}$
Avg. Solve Time (Gurobi) [s]	19.53 ± 6.24	$^{\dagger}\mathbf{0.498 \pm 3.33}$
Avg. CasADi Func. Calls [s]	—	0.0345 ± 0.0159
Avg. Classifier Query Time [s]	—	0.0049 ± 0.0031
Avg. Dual Approx. Time [s]	—	0.0083 ± 0.0041
Avg. Total Computation Time [s]	19.53 ± 6.24	$^{*}\mathbf{0.554 \pm 3.33}$

† Total solve time excluding outliers: 0.109 ± 0.01 s.
* Total computation time excluding outliers: 0.164 ± 0.0151 s.

sible set is tightly constrained from multiple directions — precisely the situations where constraint reduction is least available regardless of method. Notably, these outlier steps do not compromise safety: when screening is inconclusive, the method defaults to solving the full problem, so the worst-case behavior is equivalent to standard MPC. The computational overhead of the failed screening attempt (dual estimation and screening test) is small relative to the QP solve, so the cost of conservatism in these cases is marginal.

Excluding outliers, the *Reduced MPC* achieves 0.109 ± 0.01 s; including auxiliary overhead (Fig. 5.3), it is $35\times$ faster overall and $119\times$ faster without outliers.

Furthermore, the dual-approximation time (i.e., solving the unconstrained dual problem) using (5.11) with classifier-selected variables is $0.0083 \pm 0.0041\,\mathrm{s}$, compared to $0.0182 \pm 0.0087\,\mathrm{s}$ for the full-dimensional unconstrained dual solve without the classifier. This speedup is critical for real-time control and highlights the benefit of data-driven dimension reduction for high-dimensional dual approximation.

Fig. 5.4 illustrates the constraint screening and ADF gain reduction in two real-world nuPlan scenarios: (a) an unprotected left turn and (b) a lane merge. In both cases, the left column shows the *Reduced MPC* solution and the right column shows the *Full MPC* solution, with colored circles indicating the screening status of each predicted mode for each target vehicle. In scenario (a), the AV must navigate an unprotected left turn in the presence of oncoming traffic. The *Reduced MPC* retains only the collision avoidance constraints corresponding to the modes that are active in the *Full MPC* solution, indicated by the red and yellow circles. The light blue circles denote screened-out modes whose constraints and ADF gains are deemed inactive, confirming that the screening criterion correctly identifies the safety-relevant subset of the multi-modal predictions without discarding any constraints that would affect the AV's planned trajectory.

In scenario (b), the AV must merge into an adjacent lane occupied by multiple target vehicles traveling at highway speeds. The conflict zone spans a larger spatial region, and multiple target

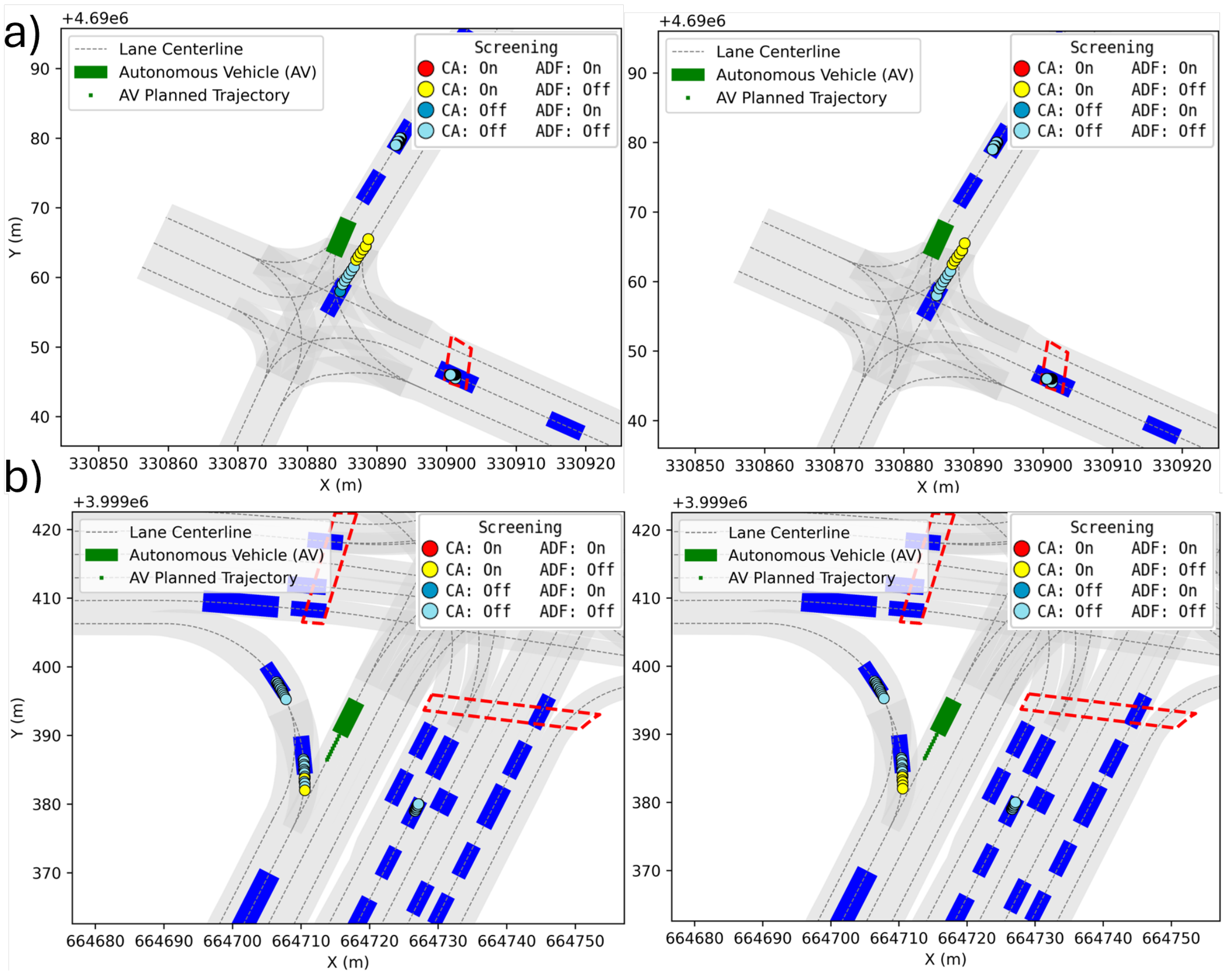


Figure 5.4: Two sampled real-world scenarios from nuPlan controlled by the *Reduced MPC* (left) and the *Full MPC* (right): (a) unprotected left turn, and (b) lane merge. Colored circles denote multi-modal predictions from a Wayformer model [75], with screening status indicated by color: *red* (CA: On, ADF: On), *yellow* (CA: On, ADF: Off), *teal* (CA: Off, ADF: On), and *light blue* (CA: Off, ADF: Off). Red dashed outlines mark the target vehicles for which collision avoidance constraints are retained in the *Reduced MPC*.

vehicles contribute active constraints. The *Reduced MPC* retains a subset of the *Full MPC* active constraint set, including the entry and exit timing constraints that govern when the AV commits to and completes the merge. These boundary constraints largely determine the yield behavior and gap acceptance decision, such that the *Reduced MPC* closely reproduces the *Full MPC* closed-loop trajectory while operating on a significantly smaller constraint set. In both scenarios, the screening procedure preserves the safety-critical constraints and ADF gains while eliminating redundant decision variables, reducing computational load without degrading closed-loop performance. To further quantify the closed-loop similarity between the *Reduced MPC* and *Full MPC*, we compute the Average Displacement Error (ADE) and Final Displacement Error (FDE) between their respective planned trajectories. In scenario (a), the *Reduced MPC* achieves an ADE of $0.15\,\mathrm{m}$ and FDE of $0.88\,\mathrm{m}$, while in scenario (b) the trajectories are nearly identical, with an ADE of $0.006\,\mathrm{m}$ and FDE of $0.018\,\mathrm{m}$. These results confirm that despite retaining only 6.80% of the collision avoidance constraints and 3.19% of the ADF gains on average, the *Reduced MPC* closely replicates the *Full MPC* closed-loop behavior at $35\times$ lower computation time. Code and videos are available at: `https://github.com/MPC-Berkeley/SHIELD_mpc`.

5.5 Chapter Summary

We proposed a safe hierarchical inference framework for lightweight duality-based screening that provably reduces the decision dimension and constraint set of an ℓ_1-regularized convex program, enabling high-rate computation with ζ-tightening. Leveraging strong convexity and Lagrangian duality, we derived certification conditions to satisfy the *original* constraints while removing irrelevant constraints and variables. We validated the approach on stochastic MPC with multi-modal predictions in complex, interactive traffic scenarios. The safe screening certificates derived in this work rely on strong convexity of the objective, Lipschitz continuity of the gradient, and Slater's constraint qualification. These assumptions hold for the quadratic-cost, linearly-constrained MPC formulations considered in our experiments and are satisfied by a broad class of practical MPC problems that use linearized dynamics and polytopic constraint sets. However, they exclude non-convex formulations arising from nonlinear vehicle dynamics, non-convex collision geometry, or mixed-integer decision variables such as pass/stop maneuver selection. Extending the certification framework to these settings is nontrivial: without strong duality, the gap bound in Theorem 1 does not hold, and the screening test may produce false negatives (failing to screen) or, more critically, could not be applied at all without alternative bounding mechanisms. Possible avenues include applying screening after convexification within a sequential convex programming loop, or deriving relaxation-based dual bounds for mixed-integer programs, though both directions require substantial theoretical development and are left to future work.

Chapter 6

Energy-Aware Lane Planning for Connected Electric Vehicles

This chapter is adapted from a previous publication: Hansung Kim et al. "Energy-Aware Lane Planning for Connected Electric Vehicles in Urban Traffic: Design and Vehicle-in-the-Loop Validation". In: *2025 IEEE 64th Conference on Decision and Control (CDC)* (2025), pp. 2132–2137. URL: https://api.semanticscholar.org/CorpusID:277452507

This chapter addresses the long-horizon planning challenge introduced in Chapter 1: jointly optimizing speed and lane-change decisions over the extended horizons enabled by vehicle connectivity yields a combinatorially large planning problem that standard MPC cannot solve in real time. Most eco-driving strategies focus solely on longitudinal speed control within a single lane. This neglects the significant impact of lateral decisions, such as lane changes, on overall energy efficiency—especially in environments with traffic signals and heterogeneous traffic flow. To address this gap, we propose a novel energy-aware motion planning framework that jointly optimizes longitudinal speed and lateral lane-change decisions using vehicle-to-infrastructure (V2I) communication. Our approach estimates long-term energy costs using a graph-based approximation and solves short-horizon optimal control problems under traffic constraints for real-time edge computation. Using a data-driven energy model calibrated to an actual battery electric vehicle, we demonstrate with vehicle-in-the-loop experiments that our method reduces total energy consumption by up to 24% compared to a human driver, highlighting the potential of connectivity-enabled real-time planning for sustainable urban autonomy.

6.1 Introduction

Connected and Automated Vehicles (CAVs) can improve safety, traffic flow, and energy efficiency by coordinating with traffic signals and nearby vehicles through vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication [33]. Eco-driving strategies leveraging Signal Phase and Timing (SPaT) data from connected traffic lights have demonstrated 11–16% energy savings [58, 36], but these methods typically assume fixed-lane driving and optimize only longitudinal motion [5, 7, 40]. In practice, lane changes also influence efficiency by avoiding slower vehicles or aligning with favorable traffic signals [7]. Yet most eco-driving studies overlook this factor, limiting their applicability to realistic multi-lane scenarios. We extend eco-driving to include lane-change decisions for energy savings for CAVs. We focus on how connectivity (V2I/V2V) can help an energy-aware lane selection strategy, extending conventional eco-driving to a multi-lane context.

In this chapter, we propose a novel motion planning framework that integrates lane-change decisions with energy-efficient speed optimization, enabled by V2I/V2V communication. Our approach uses a hierarchical control structure: a high-level decision module selects the most energy-efficient lane (leveraging SPaT data and V2V traffic information), and a low-level trajectory planner computes a safe and smooth trajectory to that lane. We validate the proposed system in a vehicle-in-the-loop (VIL) testing environment, embedding a real CAV into a virtual traffic simulation. This setup provides a real evaluation of energy savings and driving performance with physical vehicle dynamics and realistic surrounding traffic, bridging the gap between pure simulation and real-world public road deployment. As with all literature on this topic, the impact of deploying this technology at scale on global energy improvement is not studied in this work.

RELATED WORKS

Eco-Driving with V2I Connectivity

A large body of research has explored energy-efficient driving using V2I communication, particularly along signalized corridors. Optimization-based eco-driving controllers have been proposed to leverage SPaT data over corridors with multiple traffic lights. For example, in [4], the authors apply Pontryagin's Minimum Principle to develop an eco-driving controller, demonstrating fuel savings of up to 36% compared to a human-modeled driver through VIL experiments. In [8], the authors design a real-time eco-driving controller formulated as a receding-horizon optimal control problem and demonstrate its effectiveness through real-world road testing, achieving a 31% energy reduction at the cost of increased travel time. These works demonstrate that CAVs can significantly improve efficiency through longitudinal speed planning, when leveraging infrastructure information. However, these studies assume a fixed lane. The controllers react to surrounding traffics and red lights, but do not consider proactive lane changes to avoid deceleration.

Lane Changes in Eco-Driving

Only recently have researchers begun to incorporate lateral maneuvers into eco-driving. In [23], the authors note that most studies emphasize longitudinal control and neglect lane changes, which can reduce efficiency when following slower vehicles. To address this, they propose a graph-search strategy that selects lane sequences based on predicted traffic, though the effect of traffic lights is not considered. Similarly, [100] designs an overtaking strategy in which an energy-optimal speed trajectory is first computed, and a lane change is triggered if a slower vehicle disrupts that plan. These studies indicate that lane-changing can enhance longitudinal eco-driving, yet their validation remains limited to simulation. Few works have demonstrated lane-changing eco-driving in real vehicles or high-fidelity VIL platforms, aside from our previous work [42].

Contributions

This work's contributions to energy-efficient motion planning for connected, automated urban driving are as follows

- Formulation of the lane selection problem in urban environments as a joint behavior planning task over lateral and longitudinal actions, using SPaT and surrounding traffic information.
- A hybrid lane selection algorithm that combines short-horizon optimal control with a graph-based approximation of long-term energy costs, supported by a data-driven energy model calibrated on a Hyundai IONIQ5.
- Vehicle-in-the-loop experiments demonstrating up to 24% motion energy reduction compared to human driving and 7% savings over a lane-keeping eco-driving baseline.
- Highlights the role of eco-driving technologies in electric vehicles, where regenerative braking diminishes the benefit of traditional stop-and-go minimization. This emphasizes the need for a refined understanding of energy-optimal driving behavior for electric vehicles.

In summary, this work presents a modular framework for energy-aware lane selection in connected electric vehicles and provides experimental insights that support its potential for improving energy efficiency in urban driving scenarios. Understanding average savings on daily commutes is complex and outside the scope of this work.

6.2 Problem Formulation

In this section, we formulate the problem of energy-efficient lane selection for connected, automated urban driving. We adopt the hierarchical control architecture introduced in [42] and focus on designing an energy-efficient lane selector for battery electric vehicles (BEVs) in urban environments. The architecture, illustrated in Fig. 6.1, and the corresponding vehicle-in-the-loop (VIL) setup are detailed in [42]. We assume that a route from point A to point B is given and that the traffic light cycles are deterministic.

Lane Selection Problem

This paper focuses on a two-lane, one-way road, as illustrated in Fig. 6.2, although the proposed methods can be extended to roads with more than two lanes. We assume that the CAV can perfectly observe surrounding vehicles' states within a certain radius and, by leveraging V2I connectivity, can receive SPaT information for the next N_{TL} traffic lights along its route. The information available to the lane selector includes the controlled vehicle's state, denoted as $x^{EV}(t) = [s(t), v(t)]^\top$,

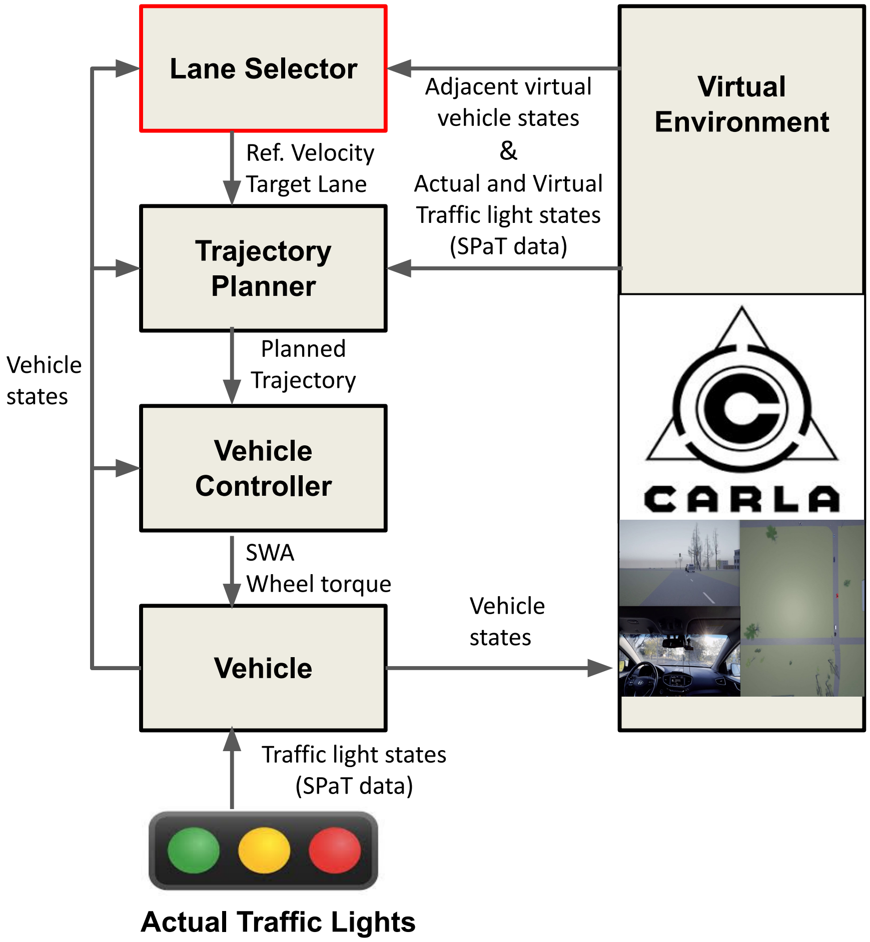


Figure 6.1: Vehicle-in-the-loop control architecture [42]

where $s(t)$ and $v(t)$ are the longitudinal displacement along the centerline of the given route and the longitudinal velocity, respectively. Additionally, the vehicle's lane information is encoded as a categorical variable $lane(t) \in \{0, 1\}$. The surrounding vehicles (SV) within the detection range are collected into

$$o(t) = [x^i(t), lane^i(t)]_{i=1}^{N_{SV}}, \tag{6.1}$$

where N_{SV} is the number of detected SVs. Lastly, the SPaT of the next N_{TL} upcoming traffic lights is collected into

$$SPaT(t) = \{s_{tl}^i, p_{tl}^i(t), t_{tl}^i(t)\}_{i=1}^{N_{TL}}, \tag{6.2}$$

where $s_{tl}^i \in \mathbb{R}$, $p_{tl}^i(t) \in \{r, g, y\}$, $t_{tl}^i(t) \in \mathbb{R}$ denote the longitudinal position of the i-th traffic light along the centerline of the route, the current signal phase (red, green, or yellow), and the remaining

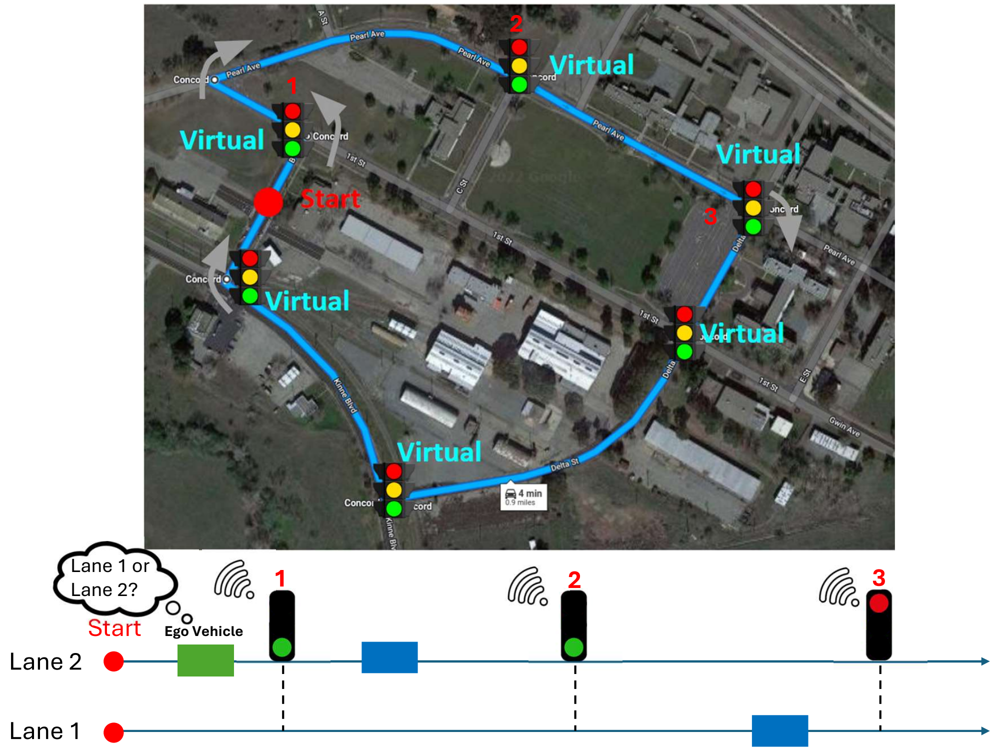


Figure 6.2: Lane selection problem in urban road

time in the current phase, respectively. The duration of the full cycle of traffic signals—green (t_g^i), yellow (t_y^i), and red (t_r^i)—are assumed to be estimated from historic SPaT data. Let $\mathcal{C}_{\text{pass}}$ denote the lane selector's decision regarding whether the controlled vehicle will pass the upcoming traffic light and which lane it should proceed in. In the case of two lanes,

$$\mathcal{C}_{\text{pass}} \in \mathcal{LS} := \{\text{PASS0, PASS1, NONPASS0, NONPASS1}\}, \tag{6.3}$$

where PASS0 and PASS1 denote the lane selector's decision to pass the upcoming traffic light in lane 0 and lane 1, respectively. NONPASS0 and NONPASS1 denote the decision to not pass the upcoming traffic light in lane 0 and lane 1, respectively. The subset of passing decisions is denoted by $\mathcal{LS}_{\text{PASS}} := \{\text{PASS0}, \text{PASS1}\}$, and the subset of nonpassing decisions is denoted by $\mathcal{LS}_{\text{NONPASS}} := \{\text{NONPASS0}, \text{NONPASS1}\}$.

Now, the lane selection problem is to design a mapping function

$$\mathcal{C}_{\text{pass},t} = f(x^{\text{EV}}(t), o(t), SPaT(t)), \tag{6.4}$$

where $\mathcal{C}_{\text{pass},t}$ is the lane selection decision at time t.

Energy-efficient Lane Selection

For energy-efficient lane selection, it is crucial to understand how the lane selector's current decision influences the vehicle's long-horizon energy consumption, in the context of current traffic information. To achieve this, we first develop a model for the vehicle's energy consumption.

Data-driven Energy Consumption Model [40]

We approximate the energy consumption of the battery electric vehicle (BEV) given a longitudinal trajectory. Specifically, we assume that the energy consumption is a function of the longitudinal displacement along the centerline and the longitudinal acceleration of the vehicle. For more details, refer to Sec. II.C. in [40]. By recording the vehicle's state-of-charge (SOC) and longitudinal motion, we fit a quadratic regression model to estimate the energy consumption rate ΔE, denoted as $\mathcal{E}(v, a)$, where v and a are the vehicle's longitudinal velocity and acceleration. The quadratic regression model is parameterized as

$$\mathcal{E}(v, a) = z^\top P z + q^\top z + r \tag{6.5}$$

where $z = [v, a]$. The parameters $P \in \mathbb{R}^{2\times 2}$, $q \in \mathbb{R}^2$, and $r \in \mathbb{R}$ are constants, with $P \succ 0$, making (6.5) a convex quadratic function.

The resulting regression and the corresponding measured energy consumption data are shown in Fig. 6.3. The mean error is approximately 1% [41]. The observed discrepancy between the model and the measured data is attributed to unmodeled non-linear effects such as aerodynamic drag, rolling resistance, and variations in motor and battery efficiency. Note that this approximate energy consumption model is used to reason about energy consumption for lane selection, but not used to measure the actual energy consumption of the vehicle. We directly read and record the SOC from the vehicle's CAN signals. Given the energy consumption model of the vehicle, we can reason about long-horizon energy-efficiency in designing an energy-efficient lane selector:

$$\mathcal{C}^{\text{eco}}_{\text{pass},t} = f(x^{\text{EV}}(t), o(t), SPaT(t), \mathcal{E}). \tag{6.6}$$

6.3 Proposed Lane Selection Method

In this section, we outline the proposed method for energy-efficient lane selection leveraging V2I connectivity. Let the globally energy optimal lane selection strategy for following a given route be $\{\mathcal{C}^{\star,\text{eco}}_{\text{pass},t}, \mathcal{C}^{\star,\text{eco}}_{\text{pass},t+1}, \dots, \mathcal{C}^{\star,\text{eco}}_{\text{pass},t+T}\}$, where T is the task horizon. In general, solving for this globally energy optimal lane change strategy for a large T is intractable due to imperfect forecasts of the NPC vehicles and vehicle dynamics modeling errors. Thus, we approximate it by solving for a locally energy optimal lane selection strategy in a receding horizon fashion. Every Δt_{LS}, we compute a new locally energy optimal lane change strategy and reference velocity profile for the trajectory planner.

The proposed method is a hybrid approach that solves $|\mathcal{LS}|$ short finite-horizon optimal control problems up to the upcoming traffic light (i.e. one for each pass/nonpass and lane choice combination). The energy cost beyond the upcoming traffic light—associated with passing or not passing

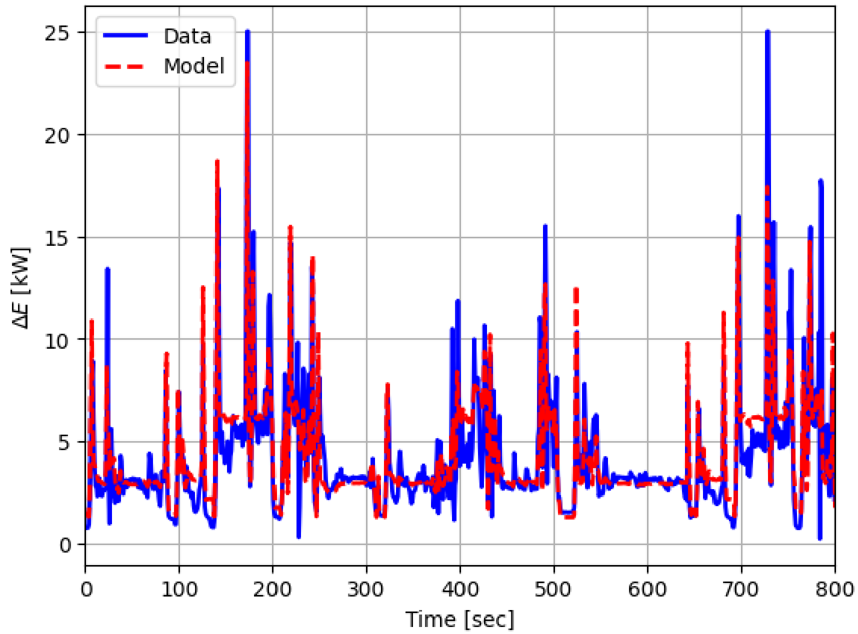


Figure 6.3: Comparison of quadratic energy consumption model and measured data

the next N_{TL} traffic lights and selecting the appropriate lane—is then approximated using a graph search method and incorporated as a terminal cost-to-go. Finally, the lane selection decision with the minimum total cost is chosen.

Optimization and Graph Search Method

In this method, we formulate a finite-horizon optimal control problem that considers the signal and phase of the next upcoming traffic light, as well as the safety constraints with respect to other NPC vehicles, based on $\mathcal{C}_{pass}$.

Vehicle Model

We model the longitudinal dynamics of the vehicle as a double integrator:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \tag{6.7}$$

where $u(t) = a(t)$ is the longitudinal acceleration of the vehicle. We discretize the dynamics (6.7) using the sampling time Δt of 0.5 seconds. The discretized dynamics are denoted as

$$x_{k+1} = Ax_k + Bu_k, \tag{6.8}$$

where x_k and u_k are state and control input at time step k, respectively.

Graph Search Terminal Cost Approximation

We simplify the lane selection problem for passing the N_{TL} traffic lights by assuming the vehicle travels in a fixed lane and at a constant velocity between each traffic light (nodes) and formulating it as a directed graph defined as

$$G = (V, E), \tag{6.9}$$

where $V = \{V_1, \mathcal{V}_2, \dots, \mathcal{V}_{N_{TL}}, V_{-1}\}$ and $\mathcal{V}_i$, $\forall i \in \mathbb{I}_1^{N_{TL}}$, represents the subset of nodes corresponding to each traffic light as depicted in Fig. 6.4. Each node in $\mathcal{V}_i$ represents the velocity at the point of passing the i-th traffic light. We grid the passing velocities from $v^G_{min} = 5$ m/s to $v^G_{max} = 10$ m/s with a grid size of 1 m/s. The starting and terminal nodes V_1 and V_{-1} represent the first traffic light ahead and the ego vehicle's position after passing the N_{TL}-th traffic light, respectively. The solution of the optimal control problem determines the passing velocity at the 1st traffic light. For simplicity, the passing velocity at the terminal node is set equal to that of the starting node.

The edges represent the cost of stopping at a traffic light—specifically, the vehicle must stop at the j-th traffic light if the signal is not green at the time it arrives at the traffic light (t_j), i.e., if $p^j_{tl}(t_j) \neq g$—as well as the approximate change in kinetic energy due to velocity transitions between nodes. The edge cost for the directed connection $V_i \to V_j$ is computed as follows:

$$c(V_i, V_j) = v(V_i)^2 - v(V_j)^2 - \mathbb{H}_{\{p^j_{tl}(t_j) \neq g\}} v(V_j)^2, \tag{6.10}$$

where $v(V)$ represents the passing velocity value of the node and ℓ denotes the ℓ-th traffic light that the node V_j represents. $tl^{\ell,p}$ at the time the ego vehicle reaches the traffic light is computed based on the travel time between nodes assuming the vehicle travels at a constant velocity between traffic lights. $\mathbb{H}(\cdot)$ is the Heaviside step function. The connections from nodes in $\mathcal{V}_{N_{TL}}$ to the terminal node V_{-1} are added to penalize the velocity deviation from the velocity at the point of passing the first traffic light. Alternatively, the data-driven energy model in (6.5) can be used to approximate the energy consumption resulting from changes in node velocities, assuming constant acceleration. However, for simplicity, this work adopts the kinetic energy model to compute edge costs. Note that the graph (6.9) satisfies additive property meaning $c(V_a, V_c) = c(V_a, V_b) + c(V_b, V_c)$ for any node connections $V_a \to V_b$ and $V_b \to V_c$. Additivity allows us to exploit dynamic programming and optimal substructure, formulating the problem as a path-finding problem and applying Dijkstra's algorithm [22] to compute $\min c(V_1, V_{-1})$ given $v(V_1)$. We define the minimum sum of the edge cost from the starting node to the terminal node as

$$J^\star_{graph}(v(V_1)) = \min_{\{V_i \in \mathcal{V}_i\}_{i=2}^{N_{TL}}} c(V_1, V_{-1}), \tag{6.11}$$

and use it to approximate the energy cost from the 1st traffic light to the N_{TL}-th traffic light, assuming the ego vehicle travels in the fixed lane and at a constant velocity.

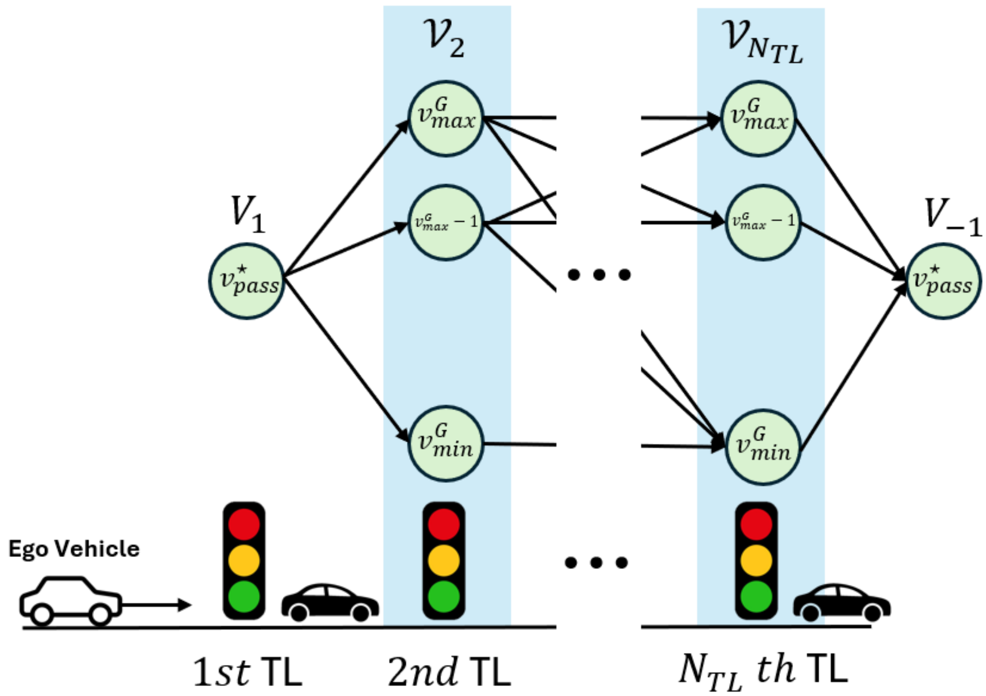


Figure 6.4: Illustration of a graph for navigation through multiple traffic lights. The graph approximates the pass or nonpass decision for a fixed lane and velocity

Constraints

The ego vehicle is subject to state and input constraints defined below

$$\mathcal{X} := \{x \in \mathbb{R}^2 |\, 2 \leq v \leq 11, \text{where } x = [s, v]^\top\}, \tag{6.12}$$
$$\mathcal{U} := \{u \in \mathbb{R} |\, -4 \leq u \leq 2\}. \tag{6.13}$$

Collision avoidance constraints with respect to the NPC vehicles are also considered, and their set representation is denoted as $\mathcal{F}^{LK}(s^{\text{front}}, v^{\text{front}})$, where s^{front} and v^{front} denote the states of the vehicle immediately preceding the ego vehicle in the same lane. We assume that the NPC vehicles maintain constant velocity during the prediction horizon N. For more details, refer to [42].

Additionally, conditional on the next traffic light's current signal phase at time, $p_{tl}^{\text{next}}(t)$, the ego vehicle is subject to traffic light passing or stopping constraints as follows

$$\mathcal{P}(s_{tl}^{\text{next}}) = \{x \in \mathbb{R}^2 | s \geq s_{tl}^{\text{next}} + d\}, \tag{6.14}$$
$$\mathcal{S}(s_{tl}^{\text{next}}) = \{x \in \mathbb{R}^2 | s_{tl}^{\text{next}} - d \geq s\}, \tag{6.15}$$

where s_{tl}^{next} is the next traffic light's relative longitudinal position along the centerline of the given route and d is the user-defined margin for controlling the conservativeness of the lane selection planner. In this work, we use $d = 3$ m. Further, these traffic light passing and stopping constraints are time-varying constraints, which means the time steps that these constraints are applied depend on the pass or nonpass decision $\mathcal{C}_{\text{pass}}$ and current phase of the next traffic light $p_{tl}^{\text{next}}(t)$. The lower and upper bounds on the time for pass and stop constraints are as follows

$$\bar{t}_{\text{pass}}(\mathcal{C}_{\text{pass}}, p_{tl}^{\text{next}}(t)) = CT(\phi, \mathcal{C}_{\text{pass}}) - t_g^{\text{margin}}, \tag{6.16a}$$

$$\underline{t}_{\text{pass}}(\mathcal{C}_{\text{pass}}, p_{tl}^{\text{next}}(t)) = CT(\phi, \mathcal{C}_{\text{pass}}) + t_g^{\text{margin}}, \tag{6.16b}$$

$$\bar{t}_{\text{stop}}(\mathcal{C}_{\text{pass}}, p_{tl}^{\text{next}}(t)) = CT(\phi, \mathcal{C}_{\text{pass}}), \tag{6.16c}$$

$$\underline{t}_{\text{stop}}(\mathcal{C}_{\text{pass}}, p_{tl}^{\text{next}}(t)) = 0, \tag{6.16d}$$

where $\bar{t}_{\text{pass}}$ and $\underline{t}_{\text{pass}}$ denote the upper and lower bounds of the time interval during which the pass constraint is active; analogous definitions apply to the stop constraint. The parameter t_g^{margin} is a user-defined safety margin that ensures the vehicle passes through the traffic light within the green phase. In this work, we set $t_g^{\text{margin}} = 2$ s. Additionally, the function CT denotes the cycle time adjustment, which accounts for the current signal phase of the upcoming traffic light and the decision to either pass or stop. Let $\phi \in \{0, 1, 2\}$ be the categorical encoding of the next traffic light's signal phase, corresponding to green, yellow, and red respectively: $[0 \mapsto g,\ 1 \mapsto y,\ 2 \mapsto r]$. The next traffic light's cycle durations is Cycle $:= [t_g^{\text{next}}, t_y^{\text{next}}, t_r^{\text{next}}]$, where t_g^{next}, t_y^{next}, and t_r^{next} denote the durations of the green, yellow, and red phases respectively—which can be obtained from the traffic light's SPaT data. The CT function is defined as follows:

$$CT(\phi, \mathcal{C}_{\text{pass}}) := t_{tl}^{\text{next}}(t) + \sum_{i=1}^{\delta(\phi, \mathcal{C}_{\text{pass}})} \text{Cycle}[(\phi + i) \bmod 3], \tag{6.17}$$

$$\delta(\phi, \mathcal{C}_{\text{pass}}) = \begin{cases} 0 & \text{if } \mathcal{C}_{\text{pass}} \in \mathcal{LS}_{\text{PASS}} \\ 2 & \text{if } \mathcal{C}_{\text{pass}} \in \mathcal{LS}_{\text{NONPASS}} \wedge \phi = 2 \text{ (r)} \\ 1 & \text{if } \mathcal{C}_{\text{pass}} \in \mathcal{LS}_{\text{NONPASS}} \wedge \phi = 0 \text{ (g)} \\ 2 & \text{if } \mathcal{C}_{\text{pass}} \in \mathcal{LS}_{\text{NONPASS}} \wedge \phi = 1 \text{ (y)}. \end{cases} \tag{6.18}$$

where $t_{tl}^{\text{next}}(t)$ is the remaining signal time on the current signal phase of the next traffic light.

Optimal Control Problem

We solve the following constrained finite-time optimal control problem for every $\mathcal{C}_{\text{pass}}$ candidates in the set (6.3) and collect the optimal cost $J^{\star}_{\text{ftocp}}(x(t), u(t), \mathcal{C}_{\text{pass}})$:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{k=t}^{t+N-1} [\mathcal{E}(v_{k|t}, u_{k|t})] + J^{LK}_{\text{smooth}}(\{x_{k|t}\}_t^{t+N-1}) \tag{6.19a}$$

$$\text{s.t.} \quad x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \tag{6.19b}$$

$$x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \tag{6.19c}$$

$$(x_{k|t}, u_{k|t}) \in \mathcal{F}^{LK}(s^{\text{front}}_{k|t}, v^{\text{front}}_{k|t}), \tag{6.19d}$$

$$x_{t|t} = x(t),\ \forall k \in \mathbb{I}_t^{t+N-1} \tag{6.19e}$$

$$x_{k|t} \in \mathcal{P}(tl^{\text{next},s}),\ \forall k \in \mathbb{I}_{[t+\underline{t}_{\text{pass}}(\mathcal{C}_{\text{pass}}, tl^{\text{next},p})]/\Delta t}^{[t+\bar{t}_{\text{pass}}(\mathcal{C}_{\text{pass}}, tl^{\text{next},p})]/\Delta t} \tag{6.19f}$$

$$x_{k|t} \in \mathcal{S}(tl^{\text{next},s}),\ \forall k \in \mathbb{I}_{[t+\underline{t}_{\text{stop}}(\mathcal{C}_{\text{pass}}, tl^{\text{next},p})]/\Delta t}^{[t+\bar{t}_{\text{stop}}(\mathcal{C}_{\text{pass}}, tl^{\text{next},p})]/\Delta t} \tag{6.19g}$$

where $x_{k|t}$ denotes the decision variable at time step k predicted at time t and N is the prediction horizon. In this work, we use $N = 140$ to achieve 1 Hz lane selection planning with the available hardware. The $\mathbb{I}_{k_1}^{k_2}$ notation denotes the set $\{k_1, k_1 + 1, \ldots, k_2\}$. The optimal control problem has three components for the cost function. The first component is the energy minimization cost which uses the data-driven energy consumption model: $\mathcal{E}$, which is quadratic and convex w.r.t. the decision variables. The second component $J^{LK}_{\text{smooth}}(\{x_{k|t}\}_t^{t+N-1})$ is the stage cost added for smoothness of the longitudinal trajectory and passenger comfort [42]. After the optimal control problem is formulated in CasADi [2] and solved using IPOPT [16], the passing velocity of the first traffic light $v^{\star}_{\text{pass}}$ is extracted from the optimal solution, which is used to initialize the graph (6.9). After computing (6.11), we define

$$J^{\star}(\mathcal{C}_{\text{pass}}) = J^{\star}_{\text{graph}}(v^{\star}_{\text{pass}}) + J^{\star}_{\text{ftocp}}(x(t), u(t), \mathcal{C}_{\text{pass}}). \tag{6.20}$$

We compute the $J^{\star}(\mathcal{C})$ for every $\mathcal{C}_{\text{pass}} \in \mathcal{LS}$. Finally, the optimization and graph search-based energy-efficient lane selector policy is

$$\mathcal{C}^{\text{eco}}_{\text{pass}} = \arg\min_{\mathcal{C}\in\mathcal{LS}}\ J^{\star}(\mathcal{C}) \tag{6.21}$$

The lane selector also provides the reference velocity profile v_{ref}, which represents the energy-efficient speed trajectory corresponding to the selected behavior decision $\mathcal{C}^{\text{eco}}_{\text{pass}}$. The lower-level trajectory planner then computes a motion trajectory that closely tracks this reference velocity while respecting traffic laws and ensuring safety with respect to surrounding vehicles, using a finer time discretization to guarantee dynamic feasibility and reactivity [42].

6.4 Experiments

Experimental Setup

We evaluate the proposed approach using a Vehicle-in-the-Loop (VIL) system that integrates a Hyundai IONIQ 5 with CARLA simulation for realistic traffic and SPaT signals. Fig. 6.5 shows the virtual CARLA simulator map, the satellite image of the physical testing site, and the test vehicle (IONIQ 5). This setup enables robust evaluation of planning and control algorithms in a safe environment while using real energy measurements on the physical vehicle. See [42] for hardware details.

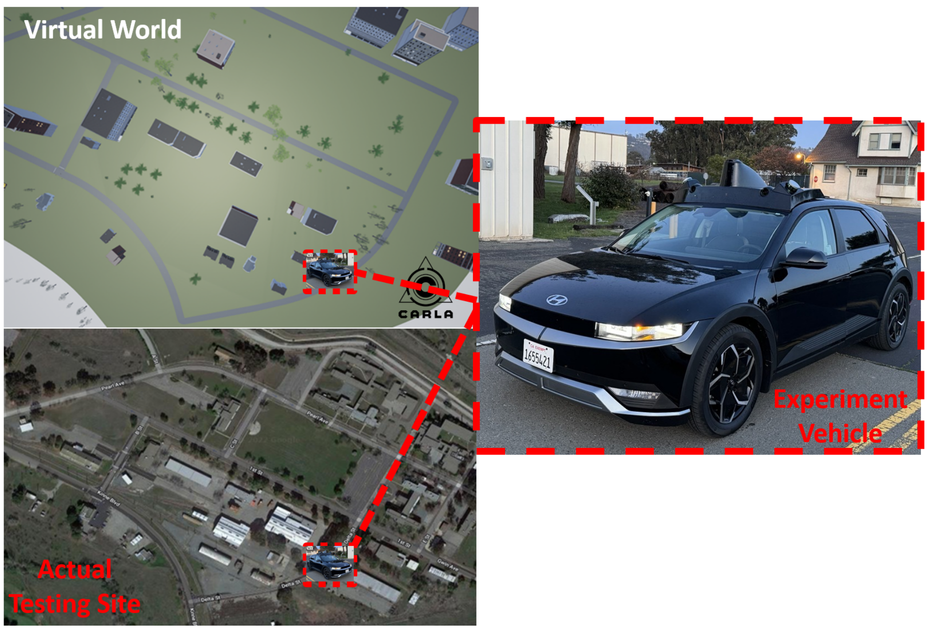


Figure 6.5: The virtual CARLA simulator map, the satellite image of the actual testing site, and the physical test vehicle

Each experiment consists of three laps around the route (approximately 4 km). For comparison, we designed two baseline approaches and conducted experiments to collect the corresponding energy consumption measurements: (i) a human driver observing the virtual environment (Baseline 1), and (ii) an autonomous eco-driving controller that optimizes longitudinal speed without lane changes (Baseline 2), which is essentially a lane-keeping algorithm that optimizes vehicle behavior based on an energy stage cost.

Experiment Results

Fig. 6.6(a) presents the total motion energy consumption, calculated by subtracting auxiliary power usage from the overall energy consumption, while Fig. 6.6(b) shows the total energy usage including auxiliary loads. The human driver (Baseline 1) tended to drive closer to the speed limit and changed lanes as needed, resulting in the shortest average trip time of 522 seconds. In contrast, the proposed algorithm averaged 823 seconds per trip, and Baseline 2 resulted in the longest trip time of 1,047 seconds. These differences in longitudinal behavior are reflected in the velocity profiles shown in Fig. 6.7.

As a result, the proposed algorithm achieved a 39% reduction in motion energy consumption compared to Baseline 1, averaging 580 Wh versus 808 Wh. Furthermore, it achieved a 24% reduction in total energy consumption (including auxiliary power usage) compared to Baseline 1 on average. While one might argue that the energy savings are simply due to the vehicle driving more slowly, the proposed algorithm actively computed energy-efficient velocity profiles and lane selections based on an energy cost function. No lower bound was imposed on the velocity profile, allowing the algorithm to explore the energy-optimal behavior of the electric vehicle. Although Baseline 2 exhibited similar motion energy consumption to the proposed algorithm (within 2% on average), its significantly longer trip time led to higher overall energy usage due to increased auxiliary power consumption, resulting in only a 7% reduction on average. Ultimately, the proposed algorithm demonstrated the lowest total energy consumption as defined by the cost function.

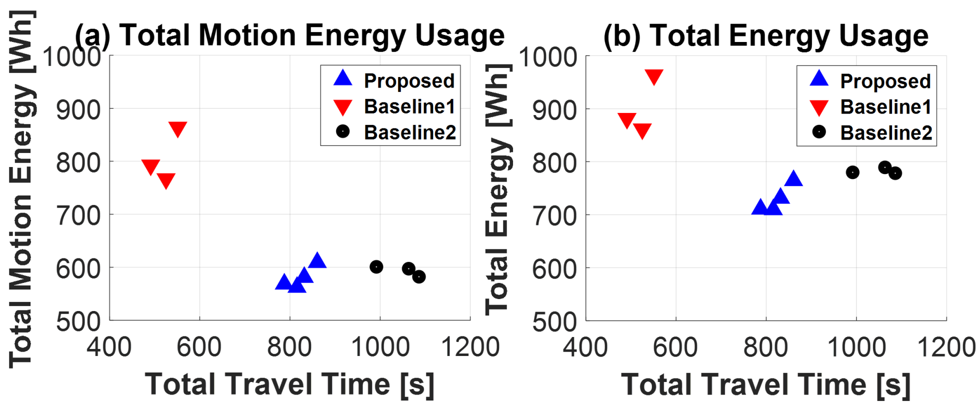


Figure 6.6: Experiment Results: (a) Total motion energy consumption (Overall energy consumption - auxiliary power usage); (b) Overall energy consumption.

6.5 Chapter Summary

This work presented an energy-aware lane selection framework for connected electric vehicles that integrates lane-change decisions with longitudinal speed optimization using V2I communication. Vehicle-in-the-loop experiments with a Hyundai IONIQ5 demonstrated up to 39% reduction in motion energy consumption compared to human driving and 7% savings relative to a lane-keeping eco-driving baseline, without imposing explicit travel-time constraints.

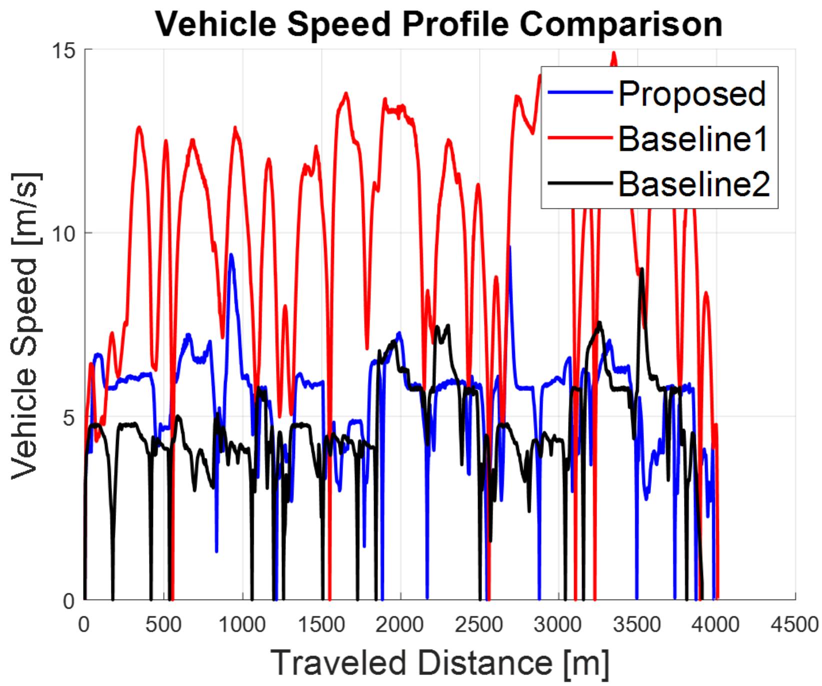


Figure 6.7: Experimental Results: Vehicle velocity profiles of a sample for each method in Fig. 6.6

The results highlight the trade-off between energy efficiency and trip time, and show that lane selection is critical in reducing overall energy use for battery electric vehicles. While demonstrated on a specific route and traffic scenario, the framework illustrates how connectivity-enabled planning can improve the sustainability of autonomous urban driving.

Future work will investigate the generalizability of the method across diverse environments and assess its robustness under varying traffic signal patterns and dynamic interactions with human drivers.

6.6 Acknowledgment

This research work presented herein is funded by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy under DE-AR0000791. We also extend our sincere gratitude to the Hyundai Motor Company's Vehicle Control Development Center and Autonomous Driving Development Center for their valuable technical assistance and support for the experimental vehicle.

Chapter 7

Conclusion

7.1 Dissertation Summary

This dissertation developed computationally efficient and scalable Model Predictive Control algorithms for autonomous driving, addressing three scalability challenges: multi-agent coupling, stochastic multi-modal predictions, and long-horizon connected operation.

Chapters 2 and 3 addressed the challenge of multi-agent coupling, where the ego vehicle must account for surrounding agents whose actions are coupled through shared constraints such as collision avoidance. Chapter 2 considered settings where an explicit coordination mechanism can be designed among connected autonomous vehicles. It introduced a facilitator-based framework for cooperative platoon lane changing, in which a designated facilitator vehicle proactively modifies the surrounding traffic environment to enable lane changes for the platoon. The coordination is implemented through distributed MPC path-planners with three modes governed by finite state machines. Numerical simulations in 200 dense traffic scenarios based on the NGSIM dataset demonstrate over a two-fold increase in lane change completion compared to a passive and opportunistic approach. The distributed formulation scales to additional CAVs without adding constraints to the path planner, with the front/rear vehicle selection having complexity $\mathcal{O}(n \log n)$. Chapter 3 considered settings where no such coordination structure exists and agents must independently reason about their interactions. It presented IGT-MPC, a decentralized algorithm that formulates multi-agent interactions as constrained dynamic games and shifts the computational cost of solving for generalized Nash equilibria offline. A neural network trained on offline GNE solutions serves as the terminal cost-to-go in a short-horizon MPC policy, enabling each agent to implicitly account for strategic interactions without solving the game online. We demonstrated IGT-MPC in competitive head-to-head racing and cooperative unsignalized intersection navigation, showing that the learned value function effectively guides MPC to replicate game-theoretic behaviors. Scaling to more agents and diverse environments remains a key challenge, as richer datasets and more complex architectures increase both data generation and gradient evaluation costs. However, a four-vehicle navigation example showed the IGT-MPC policy successfully emulates the strategic behavior in the potential game GNE solutions. Together, Chapters 2 and 3 demonstrated that interaction-aware MPC can be made tractable through distributed formulations with explicit or implicit mechanisms—finite state machines and learned game-theoretic value functions, respectively—for managing multi-vehicle coupling.

Chapters 4 and 5 addressed the scalability bottleneck arising from stochastic multi-modal predictions. When surrounding vehicles' future behaviors are represented as multi-modal distributions, the number of collision-avoidance constraints grows combinatorially as $O(N \cdot M^V)$, rendering real-time stochastic MPC intractable. Both chapters exploit the insight that most constraints are inactive at the optimum and develop duality-based screening methods to identify and remove them. Chapter 4 introduced RAID-Net, an attention-based recurrent neural network that predicts active collision-avoidance constraints using Lagrangian duality and sensitivity analysis, and a reduced SMPC problem that enforces only the relevant constraints. The approach achieves a 12x speedup in MPC solve times but lacks formal guarantees on the removed constraints. Chapter 5 addressed this limitation by deriving certifiable screening conditions from strong convexity and Lagrangian duality that guarantee all removed constraints remain satisfied and all removed decision variables are zero at optimality. The resulting SHIELD algorithm jointly reduces constraints and decision variables in an ℓ_1 regularized convex program and was validated on stochastic MPC with multi-modal predictions in complex traffic scenarios using the nuPlan simulator. The progression from Chapter 4 to Chapter 5 reflects a methodological arc from heuristic screening to provably safe problem reduction.

Chapter 6 addressed the long-horizon planning challenge that arises when vehicle connectivity extends the information available for planning beyond what onboard sensors can support. Jointly optimizing speed and lane-change decisions over these extended horizons yields a combinatorially large planning problem. We developed a real-time energy-efficient lane planner that combines short-horizon optimal control with a graph-based approximation of long-term energy costs over multiple traffic signals, avoiding the intractability of solving the full long-horizon problem directly. A data-driven energy consumption model, calibrated on a physical battery electric vehicle, serves as the stage cost in the optimal control formulation. We demonstrated over 39% reduction in motion energy consumption compared to a human driver in vehicle-in-the-loop experiments.

In summary, this dissertation introduced computationally efficient algorithms that make optimization-based planning practical and scalable for autonomous driving across the three scalability challenges identified: multi-agent coupling through distributed formulations with explicit and implicit coordination mechanisms, stochastic multi-modal predictions through duality-based constraint and variable screening with certifiable safety guarantees, and long-horizon connected operation through hybrid short-horizon optimal control with graph-based cost approximation.

7.2 Future Research Directions

Scalable Implicit Game-Theoretic MPC via Learned Scene Representations

The IGT-MPC framework in Chapter 3 is currently limited to two-agent interactions in fixed environment geometries. Generalizing to beyond two agents and arbitrary environments requires a scene representation that captures the geometry of the road (lane structure, intersection topology, traffic signals) and the motion history of all surrounding agents. A transformer-based scene encoder, similar in spirit to the architectures used in modern trajectory prediction models, could produce a fixed-dimensional encoding of the variable-size scene context. This encoding would serve as input to the MLP value function, preserving the simple gradient structure needed for real-time numerical optimization in the MPC formulation. Training on GNE solutions across diverse environments and agent counts would enable a single value function to generalize across scenarios, removing the need for scenario-specific models used in the current implementation.

Certifiable Screening for Nonconvex and Mixed-Integer MPC

The SHIELD framework in Chapter 5 relies on strong convexity and conic duality, restricting its applicability to convex programs. Many practical autonomous driving problems involve nonconvex elements such as non-convex collision-avoidance geometry, nonlinear vehicle dynamics, or discrete decisions (e.g., the pass/stop and lane selection decisions in Chapter 6, integer variables in mixed-integer MPC for motion planning). Extending the duality-based certification to these settings — potentially through convex relaxations, sequential convex programming, or mixed-integer duality — would broaden the class of problems amenable to safe, real-time screening and connect the theoretical contributions of Chapters 4–5 to a wider range of robotic and autonomous systems applications.

Combining game-theoretic interaction reasoning (IGT-MPC) with multi-modal uncertainty handling (SHIELD) into a single framework

In the current work, IGT-MPC and SHIELD address two different aspects of the same problem independently. IGT-MPC reasons about how vehicles interact — who yields, who proceeds — but assumes deterministic predictions of the other vehicle's behavior. SHIELD handles the fact that predictions are uncertain and multi-modal, but treats other vehicles as non-interactive exogenous inputs. In reality, both challenges occur simultaneously: a vehicle at an intersection faces an oncoming car that might turn left or go straight, and the ego vehicle's decision to proceed or yield should depend on which mode is realized. Combining the two means embedding a game-theoretic value function that is conditioned on the predicted mode into a stochastic MPC framework, while using duality-based screening to keep the resulting problem computationally tractable. The value function would encode interaction outcomes per mode — for instance, under the left-turn mode I should yield, but under the straight mode I can proceed — and SHIELD would ensure that the expanded problem remains solvable in real time by screening out the mode-interaction combinations that don't matter. This would produce a planner that is simultaneously interaction-aware, uncertainty-aware, and fast enough for deployment.

Bibliography

[1] Alnur Ali, Eric Wong, and J Zico Kolter. “A semismooth Newton method for fast, generic convex programming”. In: *International Conference on Machine Learning*. PMLR. 2017.

[2] Joel A E Andersson et al. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* (2018).

[3] Camilo Gonzalez Arango et al. *Learning-Based Approximate Nonlinear Model Predictive Control Motion Cueing*. 2025. arXiv: `2504.00469 [cs.RO]`. URL: `https://arxiv.org/abs/2504.00469`.

[4] Tyler Ard et al. “Energy-Efficient Driving in Connected Corridors Via Minimum Principle Control: Vehicle-in-The-Loop Experimental Verification in Mixed Fleets”. In: *IEEE Transactions on Intelligent Vehicles* (2023), pp. 1–14. DOI: `10.1109/TIV.2023.3234261`.

[5] Armin Askari et al. “Effect of adaptive and cooperative adaptive cruise control on throughput of signalized arterials”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017, pp. 1287–1292. DOI: `10.1109/IVS.2017.7995889`.

[6] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: `1607.06450 [stat.ML]`.

[7] Sangjae Bae et al. “Design and Implementation of Ecological Adaptive Cruise Control for Autonomous Driving with Communication to Traffic Lights”. In: *2019 American Control Conference (ACC)*. 2019, pp. 4628–4634. DOI: `10.23919/ACC.2019.8814905`.

[8] Sangjae Bae et al. “Ecological adaptive cruise control of plug-in hybrid electric vehicle with connected infrastructure and on-road experiments”. In: *Journal of Dynamic Systems, Measurement, and Control* 144.1 (2022), p. 011109.

[9] A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, Oct. 2009.

[10] Maulik Bhatt, Yixuan Jia, and Negar Mehr. “Strategic decision-making in multi-agent domains: A weighted constrained potential dynamic game approach”. In: *IEEE Transactions on Robotics* (2025).

[11] Richard Bishop. *60 Million Miles And Counting: Robotaxis Shift Into High Gear*. 2024. URL: `https://www.forbes.com/sites/richardbishop1/2024/07/27/60-million-miles-and-counting-robotaxis-shift-into-high-gear/` (visited on 07/27/2024).

[12] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[13] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[14] Greg Brockman et al. *OpenAI Gym*. 2016. eprint: `arXiv:1606.01540`.

[15] Tim Brüdigam et al. "Stochastic model predictive control with a safety guarantee for automated driving". In: *IEEE Transactions on Intelligent Vehicles* (2021).

[16] Richard H. Byrd, Mary E. Hribar, and Jorge Nocedal. "An Interior Point Algorithm for Large-Scale Nonlinear Programming". In: *SIAM Journal on Optimization* 9.4 (1999). DOI: `10.1137/S1052623497325107`.

[17] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd. "Enhancing Sparsity by Reweighted ℓ_1 Minimization". In: *Journal of Fourier Analysis and Applications* 14.5 (Dec. 2008), pp. 877–905. ISSN: 1531-5851.

[18] A. Cauligi et al. *CoCo: Online Mixed-Integer Control via Supervised Learning*. 2021. arXiv: `2107.08143 [cs.RO]`. URL: `https://arxiv.org/abs/2107.08143`.

[19] A. Cauligi et al. *Learning Mixed-Integer Convex Optimization Strategies for Robot Planning and Control*. 2022. arXiv: `2004.03736 [cs.RO]`. URL: `https://arxiv.org/abs/2004.03736`.

[20] Ankush Chakrabarty et al. "Learning Disagreement Regions with Deep Neural Networks to Reduce Practical Complexity of Mixed-Integer MPC". In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2021, pp. 3238–3244.

[21] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: `1406.1078 [cs.CL]`.

[22] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[23] Haoxuan Dong et al. "Flexible Eco-Cruising Strategy for Connected and Automated Vehicles With Efficient Driving Lane Planning and Speed Optimization". In: *IEEE Transactions on Transportation Electrification* 10.1 (2024), pp. 1530–1540. DOI: `10.1109/TTE.2023.3289980`.

[24] Jaap van Eck et al. "Mode Collapse Happens: Evaluating Critical Interactions in Joint Trajectory Prediction Models". In: *arXiv preprint arXiv:2506.23164*. 2025.

[25] Jose Luis Vazquez Espinoza et al. "Deep interactive motion prediction and planning: Playing games with motion prediction models". In: *Learning for Dynamics and Control Conference*. PMLR. 2022.

[26] Scott Ettinger et al. "Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9710–9719.

[27] Francisco Facchinei and Christian Kanzow. "Generalized Nash Equilibrium Problems". In: *Annals of Operations Research* 175.1 (Mar. 2010), pp. 177–211. ISSN: 1572-9338. DOI: 10.1007/s10479-009-0653-x. URL: https://doi.org/10.1007/s10479-009-0653-x.

[28] Paolo Falcone et al. "Predictive active steering control for autonomous vehicle systems". In: *IEEE Transactions on Control Systems Technology* 15.3 (2007), pp. 566–580.

[29] Lan Feng et al. "UniTraj: A Unified Framework for Scalable Vehicle Trajectory Prediction". In: *arXiv preprint arXiv:2403.15098* (2024).

[30] Roya Firoozi, Xiaojing Zhang, and Francesco Borrelli. "Formation and reconfiguration of tight multi-lane platoons". In: *Control Engineering Practice* 108 (Mar. 2021), p. 104714. DOI: 10.1016/j.conengprac.2020.104714.

[31] David Fridovich-Keil et al. "Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 1475–1481. DOI: 10.1109/ICRA40945.2020.9197129.

[32] David Fridovich-Keil et al. "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 1475–1481.

[33] Jacopo Guanetti, Yeojun Kim, and Francesco Borrelli. "Control of connected and automated vehicles: State of the art and future challenges". In: *Annual Reviews in Control* 45 (2018), pp. 18–40. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2018.04.011.

[34] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2023. URL: https://www.gurobi.com.

[35] K. Tan et al. H. Caesar J. Kabzan. "NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles". In: *CVPR ADP3 workshop*. 2021.

[36] Jihun Han et al. "Leveraging Multiple Connected Traffic Light Signals in an Energy-Efficient Speed Planner". In: *2021 American Control Conference (ACC)*. 2021, pp. 1861–1866. DOI: 10.23919/ACC50511.2021.9483424.

[37] Bernardo Hernandez and Paul Trodden. "Distributed model predictive control using a chain of tubes". In: *2016 UKACC 11th International Conference on Control (CONTROL)*. 2016, pp. 1–6.

[38] Yihan Hu et al. "Planning-oriented autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[39] Mrdjan Jankovic. "Instinctive Negotiation by Autonomous Agents in Dense, Unstructured Traffic: A Controls Perspective". In: *Annual Review of Control, Robotics, and Autonomous Systems* 7.Volume 7, 2024 (2024), pp. 97–121. ISSN: 2573-5144. DOI: https://doi.org/10.1146/annurev-control-060923-025701. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-control-060923-025701.

[40] Eunhyek Joa, Eric Yongkeun Choi, and Francesco Borrelli. "Eco-driving under localization uncertainty for connected vehicles on Urban roads: Data-driven approach and Experiment verification". In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 2904–2909. DOI: 10.1109/IV55156.2024.10588864.

[41] Eunhyek Joa, Eric Yongkeun Choi, and Francesco Borrelli. "Energy-efficient Predictive Control for Connected, Automated Driving under Localization Uncertainty". In: *IEEE Transactions on Intelligent Vehicles* (2024), pp. 1–10. DOI: 10.1109/TIV.2024.3442670.

[42] Eunhyek Joa et al. "Energy-Efficient Lane Changes Planning and Control for Connected Autonomous Vehicles on Urban Roads". In: *2023 IEEE Intelligent Vehicles Symposium (IV)*. 2023, pp. 1–6. DOI: 10.1109/IV55152.2023.10186574.

[43] Talha Kavuncu, Ayberk Yaraneri, and Negar Mehr. "Potential iLQR: A potential-minimizing controller for planning multi-agent interactive trajectories". In: *arXiv preprint arXiv:2107.04926* (2021).

[44] Arne Kesting, Martin Treiber, and Dirk Helbing. "Enhanced Intelligent Driver Model to Access the Impact of Driving Strategies on Traffic Capacity". In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 368 (Oct. 2010), pp. 4585–605. DOI: 10.1098/rsta.2010.0084.

[45] Hansung Kim. *GitHub: Implicit-Game-Theoretic-MPC*. 2024. URL: https://github.com/MPC-Berkeley/Implicit-Game-Theoretic-MPC.

[46] Hansung Kim. *Learning Two-agent Motion Planning Strategies from Generalized Nash Equilibrium for MPC*. Nov. 2024. URL: https://youtu.be/9jlz95Nor2I.

[47] Hansung Kim and Francesco Borrelli. "Facilitating Cooperative and Distributed Multi-Vehicle Lane Change Maneuvers". In: *IFAC-PapersOnLine* 56.2 (2023). 22nd IFAC World Congress.

[48] Hansung Kim, Siddharth H. Nair, and Francesco Borrelli. "Scalable Multi-modal Model Predictive Control via Duality-based Interaction Predictions". In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. 2024, pp. 1499–1504. DOI: 10.1109/IV55156.2024.10588718.

[49] Hansung Kim et al. "Energy-Aware Lane Planning for Connected Electric Vehicles in Urban Traffic: Design and Vehicle-in-the-Loop Validation". In: *2025 IEEE 64th Conference on Decision and Control (CDC)* (2025), pp. 2132–2137. URL: https://api.semanticscholar.org/CorpusID:277452507.

[50] Hansung Kim et al. "Learning Two-agent Motion Planning Strategies from Generalized Nash Equilibrium for Model Predictive Control". In: *Proceedings of the 7th Annual Learning for Dynamics & Control Conference*. Ed. by Necmiye Ozay et al. Vol. 283. Proceedings of Machine Learning Research. PMLR, Apr. 2025, pp. 112–123. URL: `https://proceedings.mlr.press/v283/kim25a.html`.

[51] Yeojun Kim, Jacopo Guanetti, and Francesco Borrelli. "Robust Eco Adaptive Cruise Control for Cooperative Vehicles". In: *2019 18th European Control Conference (ECC)*. 2019, pp. 1214–1219. DOI: `10.23919/ECC.2019.8796252`.

[52] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: `https://api.semanticscholar.org/CorpusID:6628106`.

[53] Justin Kottinger, Shaull Almagor, and Morteza Lahijanian. "Conflict-Based Search for Multi-Robot Motion Planning with Kinodynamic Constraints". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 13494–13499.

[54] Karsten Kreutz and Julian Eggert. "Analysis of the Generalized Intelligent Driver Model (GIDM) for Uncontrolled Intersections". In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021.

[55] Stanley Lam and Jayantha Katupitiya. "Cooperative autonomous platoon maneuvers on highways". In: *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2013, pp. 1152–1157. DOI: `10.1109/AIM.2013.6584249`.

[56] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. "ALGAMES: a fast augmented Lagrangian solver for constrained dynamic games". In: *Auton. Robots* 46.1 (Jan. 2022), pp. 201–215. ISSN: 0929-5593. DOI: `10.1007/s10514-021-10024-7`. URL: `https://doi.org/10.1007/s10514-021-10024-7`.

[57] Edouard Leurent and Jean Mercat. *Social Attention for Autonomous Decision-Making in Dense Traffic*. 2019. arXiv: `1911.12250 [cs.LG]`.

[58] Michael W. Levin et al. *Cost-Benefit Analysis of Fuel-Efficient Speed Control Using Signal Phasing and Timing (SPaT) Data: Evaluation for Future Connected Corridor Deployment*. Tech. rep. Final Report. Minnesota, USA: Minnesota Department of Transportation, 2023.

[59] Bai Li et al. "Optimal Cooperative Maneuver Planning for Multiple Nonholonomic Robots in a Tiny Environment via Adaptive-Scaling Constrained Optimization". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1511–1518.

[60] Keqiang Li, Jiawei Wang, and Yang Zheng. "Cooperative Formation of Autonomous Vehicles in Mixed Traffic Flow: Beyond Platooning". In: *IEEE Transactions on Intelligent Transportation Systems* (2022), pp. 1–16. DOI: `10.1109/TITS.2022.3146612`.

[61] Alexander Liniger, Alexander Domahidi, and Manfred Morari. "Optimization-based autonomous racing of 1:43 scale RC cars". In: *Optimal Control Applications and Methods* 36.5 (2015), pp. 628–647.

[62] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2017.

[63] Wenjie Luo et al. "JFP: Joint Future Prediction with Interactive Multi-Agent Modeling for Autonomous Driving". In: *Conference on Robot Learning (CoRL)*. 2023.

[64] Yugong Luo et al. "Cooperative Lane-change maneuver for multiple automated vehicles on a highway". In: *Automotive Innovation* 2.3 (2019), pp. 157–168. DOI: 10.1007/s42154-019-00073-1.

[65] Danilo P Mandic and Jonathon Chambers. *Recurrent neural networks for prediction: learning algorithms, architectures and stability*. John Wiley & Sons, Inc., 2001.

[66] Tobia Marcucci and Russ Tedrake. "Warm Start of Mixed-Integer Programs for Model Predictive Control of Hybrid Systems". In: *IEEE Transactions on Automatic Control* 66.6 (2021), pp. 2433–2448.

[67] Jil McIntosh. *Four Waymo cars cause driverless stand-off in San Fran*. 2024. URL: https://driving.ca/auto-news/crashes/waymo-cars-confused-stand-off-san-francisco (visited on 11/06/2024).

[68] Ali Mesbah. "Stochastic model predictive control: An overview and perspectives for future research". In: *IEEE Control Systems Magazine* 36.6 (2016), pp. 30–44.

[69] Donald Michie, Michael Bain, and Jean Hayes-Michie. "Cognitive models from subcognitive skills". In: *IEE control engineering series* 44 (1990).

[70] Dov Monderer and Lloyd S. Shapley. "Potential Games". In: *Games and Economic Behavior* 14.1 (1996), pp. 124–143. ISSN: 0899-8256. DOI: https://doi.org/10.1006/game.1996.0044. URL: https://www.sciencedirect.com/science/article/pii/S0899825696900445.

[71] Marcello Montanino and Vincenzo Punzo. "Reconstructed NGSIM I80-1. COST ACTION TU0903 - MULTITUDE". In: (2013).

[72] Siddharth H Nair, Eric H Tseng, and Francesco Borrelli. "Collision Avoidance for Dynamic Obstacles with Uncertain Predictions using Model Predictive Control". In: *arXiv preprint arXiv:2208.03529* (2022).

[73] Siddharth H. Nair et al. "Predictive Control for Autonomous Driving With Uncertain, Multimodal Predictions". In: *IEEE Transactions on Control Systems Technology* 33.4 (2025), pp. 1178–1192.

[74] Siddharth H. Nair et al. "Stochastic MPC with Multi-modal Predictions for Traffic Intersections". In: *25th IEEE International Conference on Intelligent Transportation Systems, ITSC 2022, Macau, China, October 8-12, 2022*. IEEE, 2022. DOI: 10.1109/ITSC55140.2022.9921751. URL: https://doi.org/10.1109/ITSC55140.2022.9921751.

[75] Nigamaa Nayakanti et al. *Wayformer: Motion Forecasting via Simple & Efficient Attention Networks*. 2022. URL: https://arxiv.org/abs/2207.05844.

[76] Jiquan Ngiam et al. "Scene Transformer: A unified architecture for predicting future trajectories of multiple agents". In: *International Conference on Learning Representations (ICLR)*. 2022.

[77] Nuro, Inc. *Nuro autonomous delivery*. `https://www.nuro.ai/`. 2023.

[78] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019.

[79] Lasse Peters et al. "Contingency games for multi-agent interaction". In: *IEEE Robotics and Automation Letters* (2024).

[80] Luigi Russo et al. "Learning for Online Mixed-Integer Model Predictive Control With Parametric Optimality Certificates". In: *IEEE Control Systems Letters* 7 (2023), pp. 2215–2220.

[81] Kanghyun Ryu and Negar Mehr. "Integrating predictive motion uncertainties with distributionally robust risk-aware control for safe robot navigation in crowds". In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 2410–2417.

[82] Jacob Sacks and Byron Boots. "Learning to Optimize in Model Predictive Control". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2022, pp. 10549–10556.

[83] Tim Salzmann et al. *Trajectron++: Dynamically-Feasible Trajectory Forecasting With Heterogeneous Data*. 2021. arXiv: `2001.03093 [cs.RO]`.

[84] Shaoshuai Shi et al. *MTR++: Multi-Agent Motion Prediction with Symmetric Scene Modeling and Guided Intention Querying*. 2024. arXiv: `2306.17770 [cs.CV]`. URL: `https://arxiv.org/abs/2306.17770`.

[85] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *J. Mach. Learn. Res.* 15.1 (2014).

[86] Bartolomeo Stellato, Tobias Geyer, and Paul J. Goulart. "High-Speed Finite Control Set Model Predictive Control for Power Electronics". In: *IEEE Transactions on Power Electronics* 32.5 (May 2017), pp. 4007–4020. ISSN: 1941-0107.

[87] Qiao Sun et al. "M2I: From Factored Marginal Trajectory Prediction to Interactive Prediction". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 6543–6552.

[88] Tesla, Inc. *Tesla robotaxi announcement*. `https://www.tesla.com/`. 2024.

[89] U.S. Department of Transportation. *National V2X Deployment Plan*. `https://www.transportation.gov/`. Accessed: 2025. 2024.

[90] Balakrishnan Varadarajan et al. "MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 7814–7821.

[91] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017.

[92] Valerio Vitelli et al. "SafetyNet: Safe Planning for Real-World Self-Driving Vehicles Using Machine-Learned Policies". In: 2021. arXiv: 2109.13602 [cs.RO]. URL: https://arxiv.org/abs/2109.13602.

[93] Po-Wei Wang and Chih-Jen Lin. "Iteration Complexity of Feasible Descent Methods for Convex Optimization". In: *Journal of Machine Learning Research* 15.45 (2014), pp. 1523–1548.

[94] Zijian Wang, Riccardo Spica, and Mac Schwager. "Game Theoretic Motion Planning for Multi-robot Racing". In: *International Symposium on Distributed Autonomous Robotic Systems*. 2018. URL: https://api.semanticscholar.org/CorpusID:68244069.

[95] Waymo LLC. *Waymo One: Our fully autonomous ride-hailing service*. https://waymo.com/waymo-one/. 2020.

[96] Waymo LLC. *Waymo's Safety Report: How We're Building a Safer Driver*. Accessed: 2025. 2017. URL: https://medium.com/waymo/waymos-safety-report-how-we-re-building-a-safer-driver-ce5f1b0d4c25.

[97] Wayve Technologies Ltd. *Wayve: Driving intelligence for any vehicle*. https://wayve.ai/. 2024.

[98] Nick Webb et al. "Waymo's Safety Methodologies and Safety Readiness Determinations". In: (2020). arXiv: 2011.00054 [cs.CY]. URL: https://arxiv.org/abs/2011.00054.

[99] Zach Williams, Jushan Chen, and Negar Mehr. "Distributed Potential iLQR: Scalable Game-Theoretic Trajectory Planning for Multi-Agent Interactions". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 01–07. DOI: 10.1109/ICRA48891.2023.10161176.

[100] Jinsong Yang et al. "A Less-Disturbed Ecological Driving Strategy for Connected and Automated Vehicles". In: *IEEE Transactions on Intelligent Vehicles* 8.1 (2023), pp. 413–424. DOI: 10.1109/TIV.2021.3112499.

[101] Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. *Near-Optimal Rapid MPC using Neural Networks: A Primal-Dual Policy Learning Framework*. 2019. arXiv: 1912.04744 [eess.SY]. URL: https://arxiv.org/abs/1912.04744.

[102] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. "Optimization-Based Collision Avoidance". In: *IEEE Transactions on Control Systems Technology* 29.3 (2021), pp. 972–983. DOI: 10.1109/TCST.2019.2949540.

[103] Edward L. Zhu. "From Cooperation to Competition: Prediction and Planning in Constrained Multi-Agent Settings Using Data-Driven and Model-Based Optimal Control Methods". English. Copyright - Database copyright ProQuest LLC. PhD thesis. University of California, Berkeley, 2024, p. 161. ISBN: 9798384454700. URL: https://www.proquest.com/dissertations-theses/cooperation-competition-prediction-planning/docview/3111087006/se-2.

[104] Edward L. Zhu and Francesco Borrelli. *A Sequential Quadratic Programming Approach to the Solution of Open-Loop Generalized Nash Equilibria for Autonomous Racing*. 2024. arXiv: 2404.00186 [cs.RO]. URL: https://arxiv.org/abs/2404.00186.

[105] Zoox, Inc. *Zoox autonomous vehicle testing*. https://zoox.com/. 2024.