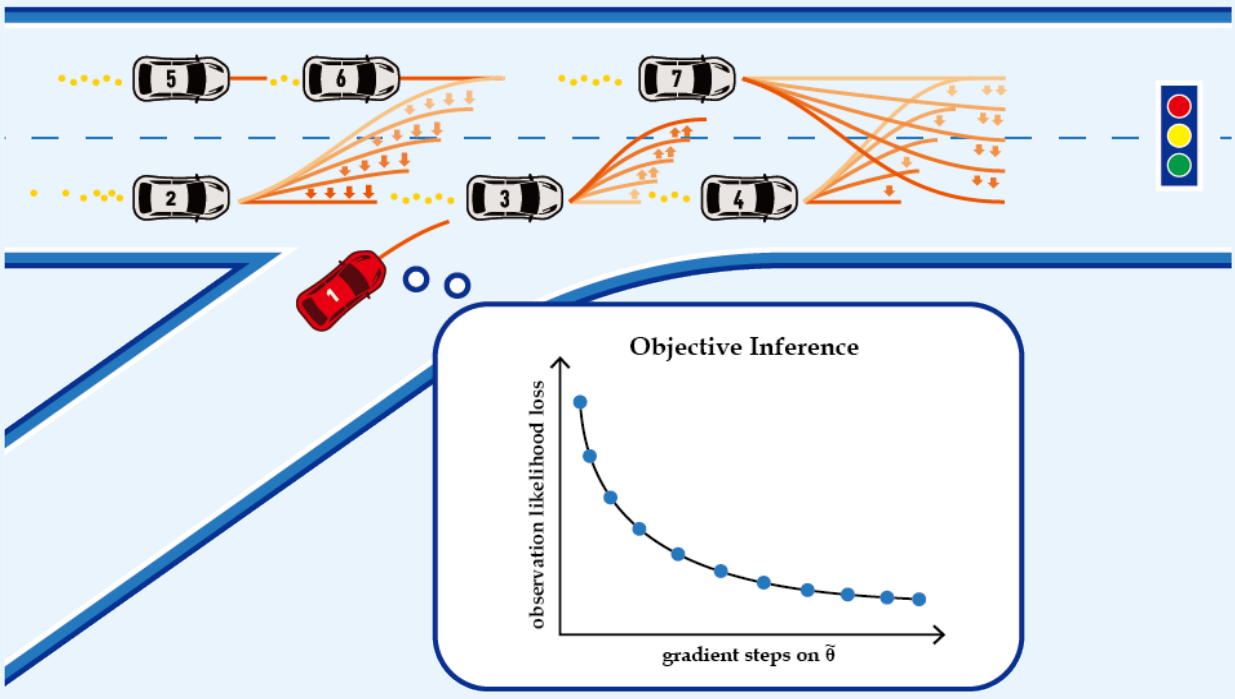
On Game-Theoretic Planning with Unknown Opponents' Objectives

RO57035: RO MSc Thesis

Xinjie Liu

Delft University of Technology



TU Delft

Cognitive Robotics

On Game-Theoretic Planning with Unknown Opponents' Objectives

by

Xinjie Liu

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Master of Science

in

Robotics

at the

Delft University of Technology

Student number:	5456509
Project duration:	July, 2022 – July, 2023
Thesis committee:	Associate Professor Javier Alonso-Mora, Supervisor
	Assistant Professor Laura Ferranti
	Assistant Professor Luca Laurenti
	Lasse Peters, Daily supervisor

This thesis is confidential and cannot be made public until September 30, 2024.

Cover: Xinjie Liu, Lasse Peters, and Javier Alonso-Mora. Learning to Play Trajectory Games Against Opponents with Unknown Objectives. *IEEE Robotics and Automation Letters (RA-L)*, 2023.

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

On Game-Theoretic Planning with Unknown Opponents' Objectives

Xinjie Liu

Abstract

Many autonomous navigation tasks require mobile robots to operate in dynamic environments involving interactions between agents. Developing interaction-aware motion planning algorithms that enable safe and intelligent interactions remains challenging. Dynamic game theory renders a powerful mathematical framework to model these interactions rigorously as coupled optimization problems. By solving the resultant coupled optimization problems to equilibrium solutions, the game-theoretic models explicitly account for the interdependence of agents' decisions and achieve simultaneous prediction and planning. Coupled constraints between players, such as collision avoidance, can also be handled explicitly. However, most existing game-theoretic motion planning approaches rely on known objective models of all agents. This assumption presents a key obstacle to real-world *ego-centric* planning applications of these methods, where only local information is available. This thesis investigates solution approaches to relax this assumption and explicitly account for the ego agent's uncertainty about other agents' objectives while adaptively conducting game-theoretic motion planning.

The main contribution of this work is an online adaptive model-predictive game-play (MPGP) framework that jointly infers other players' objectives and computes corresponding generalized Nash equilibrium (GNE) strategies. These strategies are then used as predictions for other players and control strategies for the ego agent. The adaptivity of the proposed approach is enabled by differentiating through a trajectory game solver whose gradient signal is used for maximum likelihood estimation (MLE) of opponents' objectives. Compared with existing objective inference solutions in dynamic games, the proposed approach handles general inequality constraints in games and further supports direct integration with other differentiable modules, such as neural networks (NNs). Two simulation experiments indicate that the proposed approach performs closely to solving games with known objectives and outperforms the game-theoretic and model-predictive control (MPC) baselines. Two hardware experiments further demonstrate the real-time planning capability of the planner and its real-world applicability.

In addition to this main contribution, the second contribution of this work is a variational autoencoder (VAE) pipeline built upon the proposed differentiable game solver. This contribution aims at going beyond the point estimation in the first contribution and inferring potentially multi-modal *beliefs* about players' objectives based on observations. The main idea is to employ variational inference (VI) to approximate Bayesian inference of players' objectives. The variational autoencoder (VAE) framework is utilized for amortization to avoid per-sample optimization. Initial results on a single-player example show that after training, the proposed pipeline can: (i) generate a game objective distribution that resembles the underlying training data distribution and (ii) accurately predict a narrow, uni-modal posterior objective distribution when the observation is unambiguous based on seen data in the past and (iii) generate a multi-modal belief distribution of player's objective to capture mostly likely modes in case of high uncertainty.

Acknowledgements

Studying at TU Delft has been two incredible years in my life. I want to thank my supervisors, Lasse Peters and Dr. Javier Alonso-Mora. This thesis would not have been possible without their support and guidance. I learned amazing things from them about being a good researcher and advisor. Experience in AMR has been a highlight of my journey at TUD. I want to thank my lab mates for all the discussions and joy. I also want to thank the other members of my MSc defense committee: Dr. Luca Laurenti and Dr. Laura Ferranti, for participating in this journey and discussing my work with me at my big moment.

I am thankful to Dr. Wendelin Böhmer, Dr. Wei Pan, and Dr. Sergio Grammatico (and also Dr. Javier Alonso-Mora). I enjoyed interacting with them in classes so much, and thanks for being my academic referees. I also owe great gratitude to many people I met here, Dr. Jens Kober, Dr. Julian Kooij, Dr. Luka Peternel, Prof. Martijn Wisse, Prof. David Abbink, Prof. Joost de Winter, Karin van Tongeren, Astrid van der Niet, Dr. Jihong Zhu, Dr. Bruno Brito, Cilia Claij, Yujie Tang, Giovanni Franzese, Rodrigo Pérez Dattari, Alvaro Serra, Zimin Xia and many more. They have always been so helpful and enthusiastic, making this MSc program very enjoyable!

I am especially grateful to my friends for being together on this journey, Qing Zhang, Liangchen Sui, Yuezhe Zhang, Weijia Yi, Mingjia He, Tim Verburg, Maria de Fonseca, Ruben Martin Rodriguez, Paul Féry, Vassil Atanassov, Amin Berjaoui Tahmaz, Sara Boby, Shaohang Han, Jiarong Wei, Siyuan Wu, Moji Shi, Xinyu Wang, Xianzhong Liu, Ranbao Deng, Sahánd Wagemakers, Mariano Ramírez Montero, Stan Zwinkels, Jeroen Zwanepol, Victor van der Drift, Heqi Wang, Mayank Prashar and more. Studying abroad would have been much lonelier without them. I will always miss the beautiful time we spent here; the mornings we had great discussions and the evenings we struggled with deadlines together.

Last but not least, none of this would come true without my family. I thank my parents for always supporting and believing in me, no matter what I want to pursue. They have made every dream I had and have possible and made me believe in myself. I also miss days with my cousin, Nanxiang Wang, in the Netherlands, where we reviewed our beautiful childhood memories.

I owe my biggest gratitude to my grandparents for raising me and for so much love and joy they brought to my life, which is the origin of everything. You have made me good. This is for you.

Xinjie Liu
Delft, June 2023

Contents

List of Figures

List of Tables

List of Acronyms

BIRL Bayesian inverse reinforcement learning.

ELBO evidence lower bound.

GNE generalized Nash equilibrium.
GNEP generalized Nash equilibrium problem.

IFT implicit function theorem.
iLQR iterative linear-quadratic regulator.
IOC inverse optimal control.

KKT Karush–Kuhn–Tucker.
KL Kullback–Leibler.

LICQ linear independence constraint qualification.
LQ linear-quadratic.
LQG linear-quadratic-Gaussian.
LQR linear-quadratic regulator.

MCP mixed complementarity problem.
MLE maximum likelihood estimation.
MPC model-predictive control.
MPEC mathematical program with equilibrium constraints.
MPGP model-predictive game-play.

NEP Nash equilibrium problem.
NN neural network.

OOD out-of-distribution.

POMDP partially observable Markov decision process.
POSG partially observable stochastic game.

RNN recurrent neural network.
ROS Robot Operating System.

SEM standard error of the mean.
SGD stochastic gradient descent.
SQP sequential quadratic programming.
STL signal temporal logic.

UKF unscented Kalman filter.

VAE variational autoencoder.
VI variational inference.

1 Introduction

1.1. Background

Autonomous mobile robots, such as the self-driving car in Figure 1.1, are increasingly deployed in our society and can potentially change human mobility and benefit humanity drastically. For instance, well-designed autonomous vehicles are expected to mitigate around 41% of traffic accidents caused by drivers' inattention, distractions, and inadequate surveillance [1]. Economically, just a small percentage of fully automated vehicles are projected to reduce up to 40% of the fuel consumption caused by traffic congestion, worth 166 billion dollars in 2017 in the U.S. alone [2, 3]. A 60% decrease in greenhouse gas emissions can also be expected from a transition to automated vehicles [4]. Moreover, fully automated vehicles can save time in traffic and parking and improve mobility for seniors and individuals with disabilities.

Figure 1.1: Self-driving car by Waymo in Burlingame, California. Figure source: [5].

However, reliable autonomy for such applications remains an open challenge. Completing tasks in dynamic environments like crowded urban scenarios involving multiple potentially *noncooperative* agents requires an autonomous robot to be able to intelligently and safely interact with other agents. Hence, effective solution approaches to this multi-agent decision-making problem [6] are important.

The classic optimal control view addresses this problem by modeling an ego robot's decision-making as optimizing a specified objective function, which encodes tasks to be completed. The ego agent's actions are often computed based on *fixed* predictions of other agents given by assumed behavior models. This "predict-then-plan" simplification *decouples* the ego agent's optimization from other agents' decision-making processes and reduces the problem to a single-agent optimal control problem [7]. However, this view neglects the fact that other agents in

the scene are responsive to the ego agent's actions and cannot reason about other agents' reactions for decision-making. In many multi-agent interactive settings, agents' plans and task success are naturally interdependent. For instance, in a driving scenario, one agent's acceleration and lane change may force another agent behind to slow down and avoid collision with them. Consequently, the decoupling simplification can result in inefficient or unsafe behaviors [8]. To this end, various ego-conditioned behavior models have been developed to alleviate this issue, either hand-crafted [9] or learned from data [10].

By contrast, a multi-agent perspective employing dynamic game theory is to model other agents also as rational players optimizing their individual objective functions. The resulting tightly *coupled* optimization problems between players capture all agents' decision-making simultaneously. By solving this *dynamic game* [11] to an equilibrium solution, e.g., Nash equilibrium in noncooperative games, highly interactive behavior models that directly account for interdependence between agents' decisions can be obtained [12, 13]. General constraints coupled between players, such as collision avoidance, can also be handled explicitly. All these features render game-theoretic reasoning a principled and attractive approach to interactive multi-agent decision-making.

This thesis focuses on this dynamic game-theoretic perspective to solve multi-agent motion planning problems. Section 1.2 introduces an example to contextualize the research problem and highlights a key challenge that hinders current game-theoretic planning approaches from real-world applications and motivates the contributions of this work.

1.2. Motivating Example

Take a common traffic scenario in Figure 1.2 as an example. The multi-agent motion planning problem is modeled as a dynamic game, where each driver seeks to optimize their driving efficiency while staying safe. They might have their individual preferred driving state, e.g., desired traveling speed and target lane. These aspects characterize players' objective functions in this noncooperative game. No player wishes to reduce efficiency by deviating from their desired driving state. As a consequence, agents need to negotiate without direct communication and find an underlying equilibrium strategy while exploiting strategic interdependence for their decision-making. For instance, agent number one may accelerate to merge from the ramp and suggest agent number two to slow down and give way to them. Game-theoretic planning approaches [12–15] solve dynamic games repetitively in a receding-horizon [16] fashion to compute strategic motion plans online, which we refer to as model-predictive game-play (MPGP).

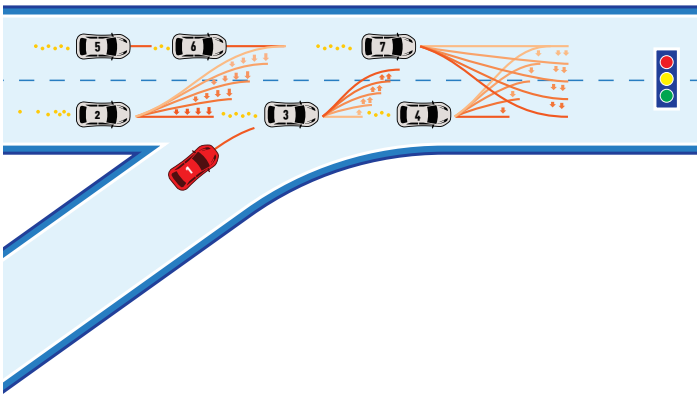


Figure 1.2: A ramping merging scenario populated by seven agents. Each driver seeks to interact with other agents to optimize their driving efficiency while staying safe.

An equilibrium strategy contains tightly coupled plans for all players in the game. An ego agent may use opponent plans as interaction-aware predictions and execute the ego plan. In order to apply game-theoretic methods for interactive motion planning from this *ego-centric* rather than *omniscient* perspective, such methods must be capable of operating only based on local information available to the ego agent. For instance, in driving scenarios as shown in Fig. 1.2, the red ego vehicle may only have partial-state observations of the surrounding vehicles and incomplete knowledge of their objectives due to unknown preferences for travel velocity, target lane, or driving style. A human driver may carefully interact with opponent drivers and try to infer those preferences on the fly. However, vanilla game-theoretic methods require known objectives of *all* players [12, 13]. This requirement constitutes a key obstacle in applying such techniques for real-world strategic decision-making. This thesis aims at addressing this challenge to relax this assumption and make game-theoretic methods more applicable to real-world applications. The following section introduces the contributions of this work.

1.3. Main Contributions

The main contribution of this work is an adaptive model-predictive game solver, which adapts to initially unknown opponents' objectives and solves for generalized Nash equilibrium (GNE) strategies. The adaptivity of this approach is enabled by differentiating through a trajectory game solver whose gradient signal is used for maximum likelihood estimation (MLE) of opponents' objectives. The proposed adaptive solver is thoroughly evaluated in simulation, comparing against game-theoretic and non-game-theoretic baselines. The solver is also tested on a real hardware platform to demonstrate its real-world applicability.

In addition to this main contribution, the second contribution of this thesis is a variational autoencoder (VAE) [17, 18] pipeline built upon the proposed differentiable game solver. This contribution aims at going beyond the point estimation of MLE in the first contribution and inferring potentially multi-modal *beliefs* about players' objectives based on observed game interactions. Initial results on the proposed variational-inference prototype are provided as a stepping-stone into future directions.

1.4. Outline

The subsequent chapters proceed as follows:

- Chapter 2 provides background information on dynamic game theory and surveys four main bodies of works related to this thesis. Chapter 3 offers formal mathematical formulations of problems being solved in this work.
- Chapter 4 introduces the main contribution of this work—the adaptive model-predictive game-play (MPGP) framework by deriving sensitivity information of a trajectory game solver and using online gradient descent for MLE of opponents' objectives. This chapter also shows extension possibilities of combining the solver with other differentiable modules. Chapter 5 evaluates the proposed framework in two simulated traffic scenarios with different settings and two hardware experiments. The experimental results show superior performance of the proposed approach over baselines.
- Chapter 6 takes a first step toward future work in inferring distributional uncertainty of unknown game objectives beyond the point-estimation approach proposed in Chapter 4. The extension is enabled by encoding the proposed differentiable solver into the decoder of a VAE. Initial results indicate a promising path in this direction.
- Chapter 7 summarizes the main results and provides an outlook toward future work.

The first contribution of this thesis (the online adaptive MPGP solver) has led to a journal article publication in *IEEE Robotics and Automation Letters (RA-L)* entitled *"Learning to Play Trajectory Games Against Opponents with Unknown Objectives,"* see Appendix A. This thesis contains part of the materials from the journal article. Beyond the results initially published in the journal article, this thesis includes the following major contributions:

- Chapter 2 provides background information on dynamic games and a more detailed survey of related works. Section 3.3 compares feedback and open-loop Nash games.
- Chapter 5 contains more qualitative results and a more quantitative experiment evaluating the approaches' performance under partial-state versus full-state observations.
- Chapter 6 presents a new pipeline for inference of distributional objective uncertainty as well as initial results.

2
Related Work

This chapter first provides background information on dynamic game theory and discusses solution concepts. Then, four main bodies of related work are discussed. Section 2.2 discusses works on *forward* games, which assume access to the objectives of all players in the scene, and the task is to compute players' strategic plans. Section 2.3 introduces works on *inverse* dynamic games that infer unknown objectives from data. Section 2.4 also relates this work to non-game-theoretic interaction-aware planning techniques. Finally, Section 2.5 surveys recent advances in differentiable optimization, which provide the underpinning for the proposed differentiable game solver of this thesis.

2.1. Dynamic Game Theory and Solution Concepts

Game theory focuses on the study of strategic interactions among multiple rational decision-making agents [19]. There are multiple ways of classifying games. In a static game, players make decisions only once and simultaneously, while in dynamic games, decision-making is *sequential* or *repeated*. In games that we consider, players are typically *non-cooperative* and may have partially conflicting goals. They are either adversarial, as in zero-sum games, or partially competing, as in general-sum games. In motion planning problems, games are primarily general-sum and dynamic [12]. Başar and Olsder [11] provide more thorough details about the computational aspects of non-cooperative dynamic games.

2.1.1. Equilibrium Concepts: Stackelberg and Nash Equilibrium

Different equilibrium concepts exist in dynamic games. In robotics literature, *Stackelberg* and *Nash* equilibrium are the most commonly studied ones. The distinction is the order in which players make their decisions. In a Stackelberg game, players make decisions sequentially, with the "leader" player(s) making their decision first and announcing it to the "follower" player(s), who then compute their best-response strategy. The solutions to this type of game are referred to as Stackelberg equilibria. In contrast, Nash games do not prescribe a hierarchical structure and all players make their decisions simultaneously, with the solutions referred to as Nash equilibria. In motion planning problems we study, such a hierarchical structure mostly does not exist, so this thesis focuses on Nash equilibrium solutions.

2.1.2. Information Pattern: Open-loop and Feedback Games

Within the scope of Nash games, there are two main types of game solutions that can be computed based on the information pattern: *open-loop* and *feedback* Nash equilibria. Feedback

Nash equilibria take into account the repetitive nature of dynamic games and yield feedback control policies, while open-loop Nash equilibria require players to choose their entire trajectories at once. Feedback Nash equilibria are sub-game perfect [20, Chapter 3] and more robust to the perturbation, actuation noise, and imperfect behavior of the agents, though they are strictly more challenging to compute. Feedback Nash equilibria can be computed via dynamic programming, while computing open-loop Nash equilibria follows Pontryagin's minimum principle [11]. This thesis focuses on open-loop solutions for tractability. Section 3.3 provides mathematical formulations for both types of problems and explains in detail this work's design considerations.

2.1.3. Feasible Sets Dependence: Standard and Generalized Nash Equilibrium Problems

Finally, in a standard Nash equilibrium problem (NEP), players are only coupled through their objectives, while in a generalized Nash equilibrium problem (GNEP), one player's feasible set may also depend on the decisions of other players (i.e., coupled constraints exist among players) [21]. In multi-agent motion planning problems, collision avoidance constraints are coupled between players, so this thesis solves GNEPs.

2.2. Forward Games

This section provides a glance at the literature on game-theoretic planning, where methods on feedback and open-loop NEPs are briefly surveyed.

2.2.1. Feedback Nash Games

Over recent years, effective computation methods for feedback games have been developed for robotic applications. A line of work, based on approximate dynamic programming [22], proposes to locally approximate nonlinear games with linear-quadratic (LQ) games. Although nonlinear games, as they appear in most robotic applications, are challenging to solve, LQ games have well-studied closed-form solutions.

Fridovich-Keil et al. [12], following the insight of Newton-type methods [23, Chapter 3] and the iterative linear-quadratic regulator (iLQR) [24, 25], propose an efficient iterative LQ approximation method for solving general nonlinear differential games. As is shown in Figure 2.1, at each iteration, the algorithm generates a nominal trajectory by forward integrating the system dynamics given current control strategies and system state. An LQ game is yielded as a local approximation by linearizing the system dynamics and quadraticizing the costs around the nominal trajectory. Solving the resultant LQ game gives new strategies. The control strategies are updated by taking steps toward the new strategies. Iterating this procedure until convergence computes an approximate local Nash equilibrium of the original game.

Laine et al. [14] generalize this iterative LQ method to handle general inequality constraints. Different from solving coupled Riccati equation at each iteration [12], Laine et al. [14] propose to use an active set method [23, Chapter 16] and solve inequality-constrained LQ games at each major iteration to handle general constraints. Moreover, at each iteration, a line-search scheme [23, Chapter 3] over the Karush–Kuhn–Tucker (KKT) residual is added for better convergence.

Mehr et al. [26] further explore this iterative LQ approximation idea in games involving noisily and boundedly rational agents. The authors show that Nash equilibria that solve these games are exactly those that solve the corresponding maximum entropy games. At each iteration, the proposed method solves a maximum entropy linear-quadratic-Gaussian (LQG) game as a local approximation.

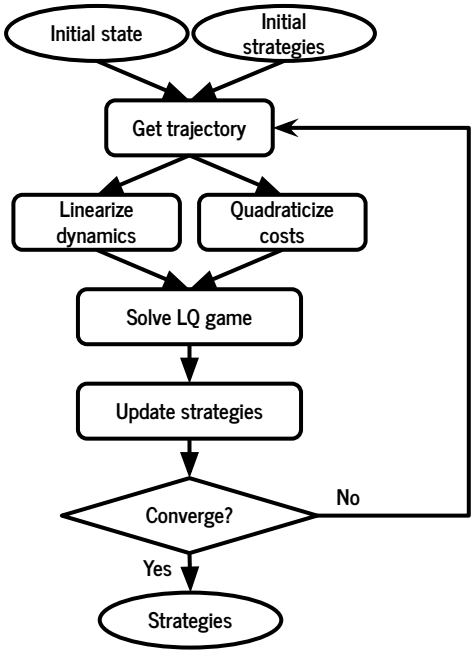


Figure 2.1: The iLQGames [12] approach iteratively approximates a nonlinear game with linear-quadratic (LQ) games and updates strategies.

2.2.2. Open-Loop Nash Games

Although solutions to feedback games have evolved over recent years, they are still highly computationally heavy to solve and are not easy to generalize to many-player cases. In contrast, open-loop solutions are easier to compute while being capable of modeling many interactions in traffic scenarios [13], although they are less expressive [27].

Iterated Best Response

The iterated best response algorithm is a classic paradigm that solves for (local) Nash equilibria [28–32]. At each iteration, the strategy of one player is computed while the strategies of the other players are fixed, reducing the problem to a single-player optimal control problem. The process is repeated iteratively until convergence, yielding a local equilibrium. Wang, Spica, and Schwager [29], Wang et al. [30], Spica et al. [31], and Schwarting et al. [32] have shown this method in autonomous drone racing, car racing, and self-driving applications. Although the iterated best response algorithm is simple, they have no convergence or stability guarantees for noncooperative games [16, Chapter 6].

Finite Games over Motion Primitives

A more efficient computation scheme is to discretize the input space and generate motion primitives, whose costs constitute a finite game [11, Chapter 3]. For a two-player racing game, Liniger and Lygeros [33] solve the resultant bimatrix games efficiently via sequential maximization approach. As a result, highly aggressive maneuvers and interactive driving behaviors are generated in a two-player racing game on a real model racing car platform with a planning frequency higher than $200\,\mathrm{Hz}$.

Rather than sampling motion primitives randomly, Peters et al. [34] use neural network (NN) to learn to propose trajectory candidates, refined by a differentiable optimization solver. Via differentiating through the entire pipeline, trajectory candidates are also decision variables and are proposed in a more principled way.

Gradient-Based Solvers

Gradient-based solvers have also been developed to compute open-loop GNE. Cleac'h, Schwager, Manchester, et al. [13] propose an augmented Lagrangian approach that casts minimization of the augmented Lagrangian function as a root-finding problem of the Lagrangian gradient. The descent direction is given by taking Newton steps on the augmented Lagrangian function. As a result, they show convergence of their method to local open-loop Nash equilibria while handling general coupled constraints. Zhu and Borrelli [15] propose a sequential quadratic programming (SQP) approach to compute the same type of solutions. At each iteration, instead of taking full steps given by the SQP iterations, a backtracking line search strategy over a merit function based on KKT residuals and constraint violation is proposed for better convergence. The authors demonstrate that their method [15] shows a higher success rate than the approach by Cleac'h, Schwager, Manchester, et al. [13] in a few traffic scenarios.

This thesis also employs a gradient-based solver to solve open-loop games. As will be discussed in Section 4.1, this work does so by casting an open-loop Nash game as an equivalent mixed complementarity problem (MCP) and solving them efficiently.

2.3. Inverse Games

Works in Section 2.2 assume known objective models of every agent in the games. This section surveys three types of methods in *inverse game* literature, where the task is to identify players' objectives given observed interactions. Figure 2.2 schematically describes forward and inverse game problems. Beyond existing methods, this work proposes a gradient-descent method via differentiable programming to solve inverse games. In Chapter 4, forward game and inverse game solutions will be combined in the proposed pipeline, where strategies are computed via solving forward games with objectives estimated from inverse game solutions.

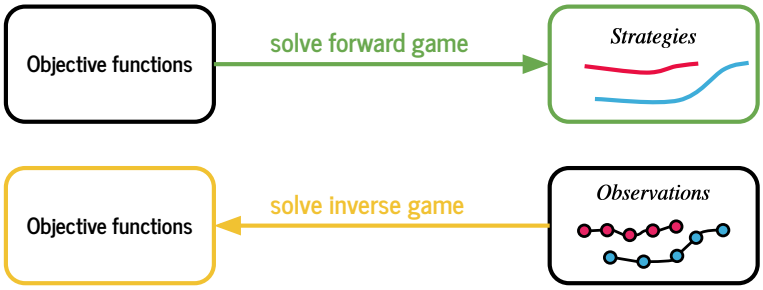


Figure 2.2: Forward games versus inverse games.

2.3.1. Bayesian Filtering

The inverse game problem can be framed as an inference task, making the Bayesian filtering technique [35] a natural choice. Peters [36] uses a particle filter to estimate human objective parameters, improving prediction accuracy in game-theoretic interactions compared with methods without objective inference. However, the particle filtering scheme has limited scalability to higher dimensional latent parameter spaces.

Le Cleac'h, Schwager, and Manchester [37] address the objective estimation problem using an online unscented Kalman filter (UKF), which drastically reduces the required sampling complexity. The belief over unknown objective parameters θ is modeled as a Gaussian distribution and sigma points for the UKF are sampled from it. At each time step, a full dynamic game is solved for each sigma point, and sigma points inducing more likely trajectories are assigned with higher weights through a standard UKF update. Controls for the ego robot are

generated by solving a dynamic game against the mean of the estimated parameters. This method has three limitations. First, it requires solving $2n + 1$ full dynamic games at each time step, with n being the dimension of unknown parameters. This computation quickly becomes complex, e.g., when the number of players increases. Second, objectives may not be unimodal in reality, leading to inaccuracies in Gaussian representations. Third, when sampling sigma points, it is important to carefully control the spread, as different sigma points may result in different local Nash equilibria.

2.3.2. Minimizing Karush–Kuhn–Tucker (KKT) Residuals

Another line of work takes inspiration from inverse optimal control (IOC) literature [38–41], where the residuals of the first-order optimality conditions of an optimal control problem are minimized. The key advantage of this formulation is that since the demonstrations are assumed to be optimal and satisfy the constraints of the forward problem, the inverse problem can be formulated as an *unconstrained* optimization problem. Also, if the objective function is convex in the parameter space at the demonstrations, the resulting inverse problem is also convex [42]. However, a major limitation of this type of formulation is that it requires full state-action demonstrations to evaluate the KKT residuals. Furthermore, Menner and Zeilinger [41] show that KKT residual methods are inferior to MLE methods that will be discussed in Section 2.3.3 for noise-corrupted data, since they assume constraint satisfaction at the demonstrations.

In the context of multi-agent games, Rothfuß et al. [43] assume exactly one player's objective function is unknown. The resultant inverse problem is a linear-quadratic regulator (LQR) problem. As a result, the residual minimization problem can be effectively solved by finding the Riccati solution. Awasthi and Lamperski [44] further generalize this idea to games with inequality constraints, where all players' objective functions and Nash equilibrium strategies are unknown. Despite the fact that the "forward" problem is an equilibrium problem between players, the resulting minimization of KKT residuals becomes N independent convex optimal control problems. However, similar to their counterparts in IOC, full state-action demonstrations are required to evaluate the residuals. Additionally, handling inequality constraints in forward games requires extra caution, as directly encoding the complementarity conditions in the KKT system as constraints in inverse games may violate the constraint qualification [45].

2.3.3. Maximum Likelihood Estimation (MLE)

While minimizing KKT residuals requires noise-free, full state-action demonstrations, another line of work formulates a maximum likelihood estimation (MLE) problem and naturally accounts for noise-corrupted, partial-state observations.

Peters et al. [46] match trajectories induced by the inferred parameters with observational data and illustrate that the probability of observing these trajectories can be maximized through a constrained optimization problem. Optimality conditions of the corresponding forward games are encoded as constraints in the inverse games. This probabilistic framework obviates the need for complete state-action demonstrations and inherently accounts for observational noise. It has been demonstrated that, in scenarios with noise-free and full state-action observations, MLE methods [46] and KKT residual methods [44] yield equivalent results. In contrast, in settings characterized by noise-corrupted and partial observations, the MLE formulation outperforms by far the KKT-residual method. However, this improvement comes at the cost of increased computational complexity, as the resulting inverse problems tend to be non-convex, even when the original forward problems are LQ games. Moreover, when inequality constraints exist in forward games, the resultant inverse games have complementarity constraints, and only a few tools are available to solve them.

While existing MLE-based approaches [46] do not support inequality constraints in forward

games, this work handles them by encoding them in a forward game solver and casting a constrained MLE problem as an unconstrained one, as will be presented in Chapter 4.

2.4. Non-Game-Theoretic Interaction Models

Besides game-theoretic methods, two categories of interaction-aware decision-making techniques have been studied extensively in the context of collision avoidance and autonomous driving: (i) approaches that learn a navigation policy for the ego agent directly without explicitly modeling the responses of others [47–50], and (ii) techniques that explicitly predict the opponents' actions to inform the ego agent's decisions [10, 51–54]. This latter category may be further split by the granularity of coupling between the ego agent's decision-making process and the predictions of others. In the simplest case, prediction depends only upon the current physical state of other agents [55]. More advanced interaction models condition the behavior prediction on additional information such as the interaction history [51], the ego agent's goal [52, 53], or even the ego agent's future trajectory [10, 54].

This work is most closely related to the latter body of work: by solving a trajectory game, this thesis' approach captures the interdependence of future decisions of all agents; and by additionally inferring the objectives of others, predictions are conditioned on the interaction history. However, a key difference in this work is that it explicitly models others as rational agents unilaterally optimizing their own costs. This assumption provides additional structure and offers a level of interpretability of the inferred behavior.

2.5. Differentiable Optimization

A wide range of work, e.g., differentiable physical engine [56, 57], differentiable rendering [58], and differentiable optimization [59, 60], relies on differentiating through various procedures to get gradient information. Backpropagation of such gradient information enables to have more expressive neural architectures.

The differentiable optimization technique is especially relevant to our application. For instance, via deriving the sensitivity information of a convex optimization solution w.r.t. the problem data, Geiger and Straehle [61] propose to train a neural network to predict unknown parameters for a trajectory optimization problem. The approach proposed by this thesis is enabled by differentiating through a GNE solver. Work by Geiger and Straehle [61] focuses on optimization problems and thus only applies to special cases of games. In contrast, differentiating through a GNEP involves N *coupled* optimization problems. This challenge is addressed in section 4.2.

2.6. Summary

The take-away messages from this chapter are:

- This thesis focuses on open-loop dynamic games and solves for generalized Nash equilibrium solutions where coupling constraints between players are present, such as collision avoidance constraints.
- In forward games, players' cost functions are typically known, and the objective is to compute equilibrium strategies.
- In inverse games, players' cost functions are unknown, and the task is to estimate cost functions from observations. In optimization-based inverse game solutions, MLE methods have the advantage of handling partial-state and noise-corrupted observations over approaches that minimize KKT residuals, at the cost of higher computational complexity.

3

Preliminaries

This chapter provides mathematical formulations for two key concepts underpinning this work: forward and inverse dynamic games. Section 3.1 and Section 3.2 present how multi-agent motion planning can be formulated as a general-sum game and introduce the notation used throughout this thesis. In forward games, Section 3.3 provides formulations for both feedback and open-loop Nash games and motivates the choice of open-loop games by this thesis. Then, the inverse game formulation is presented in Section 3.4.

3.1. Multi-Agent Motion Planning as a Game

Many robot motion planning problems involving multi-agent interactions can be formulated as general-sum games. For instance, in a multi-agent collision avoidance setting in Figure 3.1, no agent wants to deviate from the direct path to their goal and lose efficiency. At the same time, they need to avoid collisions with one another. Hence, they must negotiate without direct communication and find an underlying equilibrium strategy. This planning problem can be formulated as an N-player general-sum game. Moreover, this thesis assumes the ego agent computes their plans from an *ego-centric* perspective, where they do not have perfect information about other agents' intent.

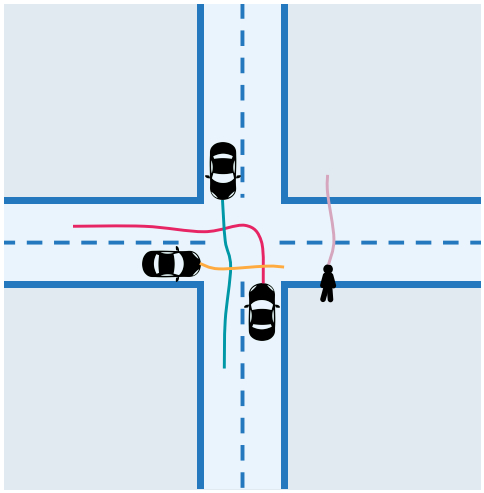


Figure 3.1: Several agents are navigating at an intersection. The multi-agent motion planning problem can be formulated as a general-sum dynamic game.

3.2. Notation

In the subsequent sections, we consider N-player games in discrete time with a horizon of T. We denote each player i's control input as $u_t^i \in \mathbb{R}^{m^i}$, which they may use to influence their state $x_t^i \in \mathbb{R}^{n^i}$ at each discrete time $t \in [T-1]$. Colons are used to denote time slicing, e.g., $x_{t:T-1}^i$ means player i's state from the time step t to $T-1$. Note that although we assume the evolution of each player's state is characterized by an individual dynamical system $x_{t+1}^i = f^i(x_t^i, u_t^i)$ for clarity, the joint system dynamics might not be separable by player in general games. Throughout the remainder of this thesis, we introduce the following short-hands for brevity: boldface and capitalization are used to indicate aggregation of variables over players and time, e.g., $\mathbf{x}_t := (x_t^1, \ldots, x_t^N)$, $U^i := (u_1^i, \ldots, u_{T-1}^i)$, $\mathbf{X} := (\mathbf{x}_1, \ldots, \mathbf{x}_T)$. With a given initial state $\hat{\mathbf{x}}_1 := (\hat{x}_1^1, \ldots, \hat{x}_1^N)$, players seek to find a control sequence U^i to minimize their own cost function $J^i(\mathbf{X}, U^i; \theta^i)$, which can take in a parameter vector θ^i.[1] Additionally, each player must consider private inequality constraints ${}^p g^i(X^i, U^i) \geq 0$ as well as shared constraints between players ${}^s g(\mathbf{X}, \mathbf{U}) \geq 0$, such as collision avoidance with shared responsibility among players.

3.3. Forward Games

This section provides formulations of open-loop and feedback Nash games.

3.3.1. Open-Loop Nash Games

First, we consider an open-loop Nash game, where the strategies are a sequence of control inputs chosen all at once. This general-sum noncooperative game can be cast as a tuple of N coupled trajectory optimization problems:

$$\forall i \in [N] \begin{cases} \min\limits_{X^i, U^i} & J^i(\mathbf{X}, \mathbf{U}; \theta^i) \\ \text{s.t.} & x_{t+1}^i = f^i(x_t^i, u_t^i), \forall t \in [T-1] \\ & x_1^i = \hat{x}_1^i \\ & {}^p g^i(X^i, U^i) \geq 0 \\ & {}^s g(\mathbf{X}, \mathbf{U}) \geq 0. \end{cases} \tag{3.1}$$

Because of the shared inequality constraints ${}^s g(\mathbf{X}, \mathbf{U}) \geq 0$, each player's feasible set in this problem may depend upon the decision variables of others, which is referred to as a GNEP rather than a standard NEP [21].

As a solution to the game in Equation (3.1), an open-loop generalized Nash equilibrium is a tuple of GNE strategies $\mathbf{U}^* := (U^{1*}, \ldots, U^{N*})$ that satisfies the inequalities $J^i(\mathbf{X}^*, U^{i*}; \theta^i) \leq J^i((X^i, \mathbf{X}^{\neg i*}), U^i; \theta^i)$ for any feasible deviations (X^i, U^i) of any player i, where we denote all but player i's states as $\mathbf{X}^{\neg i}$. We only require these conditions to hold locally for tractability. Then, no player has a unilateral incentive to deviate *locally* from a GNE in a feasible direction to get a reduced cost.

[1] The role of this parameter will be introduced in Section 3.4.

3.3.2. Feedback Nash Games

In feedback Nash games, the resultant strategies are feedback policies instead of open-loop trajectories. We define Nash equilibrium *policies* $\pi_t^i(\mathbf{x}_t)$ at each stage t as a solution to the following sub-game optimization problem starting at that stage:

$$\forall i \in [N] \begin{cases} \pi_t^i(\mathbf{x}_t) := \tilde{u}_t^i \in \underset{u_t^i}{\arg} \underset{\mathbf{x}_{t:T}, U_{t:T-1}^i, \tilde{\mathbf{U}}_{t+1:T-1}^{\neg i}}{\min} & J_t^i(\mathbf{X}_{t:T}, U_{t:T-1}^i, \tilde{\mathbf{U}}_{t:T-1}^{\neg i}; \theta^i) \\ & \text{s.t. } \tilde{\mathbf{u}}_s^{\neg i} = \boldsymbol{\pi}_s^{\neg i}(\mathbf{x}_s), \forall s \in \{t+1, \dots, T-1\} \\ & \quad \mathbf{x}_{s+1} = f(\mathbf{x}_s, u_s^i, \tilde{\mathbf{u}}_s^{\neg i}), \forall s \in \{t, \dots, T-1\} \\ & \quad x_t^i = \hat{x}_t^i \\ & \quad {}^p g^i(X_{t:T}^i, U_{t:T-1}^i) \geq 0 \\ & \quad {}^s g(\mathbf{X}_{t:T}, U_{t:T-1}^i, \tilde{\mathbf{U}}_{t:T-1}^{\neg i}) \geq 0 \end{cases} \tag{3.2}$$

where the operator $\underset{a}{\arg}\min_{a,b}$ indicates minimization over a and b, but only the value of a is returned, and a tilde $\tilde{\mathbf{U}}_{t:T-1}^{\neg i}$ indicates a Nash equilibrium strategy. We are interested in solving for a feedback GNE of the sub-game starting at $t = 1$.

There are two important benefits from this formulation. As feedback GNE are state-dependent optimal policies, they are robust against disturbances, actuation noise, and imperfect behavior of other agents, referred to as sub-game-perfect strategies [20, Chapter 3]. Furthermore, the equilibrium constraints $\tilde{\mathbf{u}}_s^{\neg i} = \boldsymbol{\pi}_s^{\neg i}(\mathbf{x}_s)$ constrain other players' controls to be from GNE feedback policies $\boldsymbol{\pi}_s^{\neg i}(\mathbf{x}_s)$ for time $s > t$, and other players' controls are also treated as decision variables in this formulation. Therefore, feedback dynamic games have higher expressiveness, allowing for optimization of cost functions and constraints that would otherwise be impossible to optimize in open-loop games [62]. However, solving the feedback sub-game starting at $t = 1$ involves solving $T - 1$ nested equilibrium problems resulting from the equilibrium constraints. Besides, other computational challenges may also present, e.g., evaluating the gradient of the equilibrium policies $\nabla_{\boldsymbol{x_s}} \boldsymbol{\pi}_s^{\neg i}(\mathbf{x}_s)$ in gradient-based methods [14]. Mainly because of the significantly heavier computational burden and challenge, this thesis focuses on open-loop dynamic games.

3.4. Inverse Games

While solving a forward game requires finding equilibrium strategies for all players given known objectives, an inverse game amounts to finding players' objectives which explain observed behavior. This section now switches context to the *inverse* dynamic game setting.

Let $\boldsymbol{\theta} := (\hat{\mathbf{x}}_1, \theta^2, ..., \theta^N)$ denote the aggregated tuple of parameters initially unknown to the ego agent with index 1. Note that we explicitly infer the initial state of a game $\hat{\mathbf{x}}_1$ to account for the potential sensing noise and partial state observations. To model the inference task over these parameters, we assume that the ego agent observes behavior originating from an unknown Nash game $\Gamma(\boldsymbol{\theta}) := (\hat{\mathbf{x}}_1, {}^s g, \{f^i, {}^p g^i, J^i(\cdot; \theta^i)\}_{i \in [N]})$, with objective functions and constraints parameterized by initially unknown values θ^i and $\hat{\mathbf{x}}_1$, respectively.

Similar to the existing method [46], we employ an MLE formulation to allow observations to be *partial* and *noise-corrupted*. In contrast to that method, however, we also allow for inequality constraints in the hidden game. That is, we propose to solve

$$\begin{aligned} \max_{\boldsymbol{\theta}, \mathbf{X}, \mathbf{U}} \quad & p(\mathbf{Y} \mid \mathbf{X}, \mathbf{U}) \\ \text{s.t.} \quad & (\mathbf{X}, \mathbf{U}) \text{ is a GNE of } \Gamma(\boldsymbol{\theta}), \end{aligned} \tag{3.3}$$

where $p(\mathbf{Y} \mid \mathbf{X}, \mathbf{U})$ denotes the likelihood of observations $\mathbf{Y} := (\mathbf{y}_1, ..., \mathbf{y}_T)$ given the estimated game trajectory $(\mathbf{X}, \mathbf{U})$ induced by parameters $\boldsymbol{\theta}$. This formulation yields a *mathematical program with equilibrium constraints (MPEC)* [63], where the outer problem is an estimation problem while the inner problem involves solving a dynamic game. As pointed out in Section 2.3.3, encoding the equilibrium constraints typically yields a non-convex problem at the outer level, even in relatively simple LQ settings. Furthermore, when a forward game contains inequality constraints, the resultant inverse problem necessarily includes *complementarity constraints* with only a few tools available. The next chapter shows how to transform Equation (3.3) into an unconstrained problem by making the inner game differentiable, which also enables combination with other differentiable components.

3.5. Summary

Finally, this section summarizes this chapter:

- Feedback Nash games are nested equilibrium problems, while open-loop Nash games are flat equilibrium problems. Although feedback games are more expressive, they are significantly more challenging to solve. Gradient-based solutions also involve extra challenges [14]. Hence, for computational reasons, this thesis solves for open-loop solutions.
- Forward open-loop Nash games can be formulated as coupled trajectory optimization problems with general coupled constraints.
- Inverse Games can be cast as MLEs with equilibrium constraints. Non-convexity of the resultant problem and dealing with complementarity constraints are the major challenges.

4

Adaptive Model-Predictive Game-Play (MPGP) Framework

This thesis aims at solving the problem of MPGP from an ego-centric perspective, i.e., without complete prior knowledge of other players' objectives. To this end, this chapter presents the main contribution of this work—an adaptive MPGP planner enabled by implicit differentiation of a forward game solver. Online, it first performs MLE of unknown objectives by solving an *inverse game* (Section 3.4); then, it solves a *forward game* using this estimate to compute strategic motion plans (Section 3.3). The procedure is repeated in a receding-horizon fashion.

The main contribution is presented in three steps. First, Section 4.1 and Section 4.2 cast a GNEP as an mixed complementarity problem (MCP) and derive the gradient of an MCP solution w.r.t. the unknown parameters. Based on this derivation, Section 4.3 then presents an algorithm that infers parameters online using the gradient information to plan safe and efficient motion in an adaptive MPGP fashion.

4.1. Forward Games as Mixed Complementarity Problems (MCPs)

This section discusses the conversion of the GNEP in Equation (3.1) to an equivalent MCP. There are three main advantages of taking this view. First, there exists a wide range of off-the-shelf solvers for this problem class [64]. Furthermore, MCP solvers directly recover strategies for all players *simultaneously* in a principled way and avoid the cycling issues commonly observed in per-player iteration schemes such as iterated best response [28]. Finally, this formulation makes it easier to reason about derivatives of the solution w.r.t. to problem data. As will be discussed in Section 4.3, this derivative information can be leveraged to solve the inverse game problem of Equation (3.3).

In order to solve the GNEP presented in Equation (3.1) we derive its first-order necessary conditions. We collect all equality constraints for player i in Equation (3.1) into a vector-valued function $h^i(X^i, U^i; \hat{x}_1^i)$, introduce Lagrange multipliers μ^i, ${}^p\lambda^i$ and ${}^s\boldsymbol{\lambda}$ for constraints $h^i(X^i, U^i; \hat{x}_1^i)$, ${}^pg^i(X^i, U^i)$, and ${}^sg(\mathbf{X}, \mathbf{U})$ and write the Lagrangian for player i as

$$\mathcal{L}^i(\mathbf{X}, \mathbf{U}, \mu^i, {}^p\lambda^i, {}^s\boldsymbol{\lambda}; \boldsymbol{\theta}) = J^i(\mathbf{X}, \mathbf{U}; \theta^i) + \mu^{i\top} h^i(X^i, U^i; \hat{x}_1^i) - {}^s\boldsymbol{\lambda}^{\top} {}^sg(\mathbf{X}, \mathbf{U}) - {}^p\lambda^{i\top} {}^pg^i(X^i, U^i).$$

Note that we share the multipliers associated with shared constraints between the players to encode equal constraint satisfaction responsibility [65]. Under mild regularity conditions, e.g., linear independence constraint qualification (LICQ), a solution of Equation (3.1) must satisfy

the following joint KKT conditions:

$$\forall i \in [N] \begin{cases} \nabla_{(X^i,U^i)} \mathcal{L}^i(\mathbf{X}, \mathbf{U}, \mu^i, {}^p\lambda^i, {}^{\boldsymbol{s}}\boldsymbol{\lambda}; \boldsymbol{\theta}) = 0 \\ 0 \leq {}^p g^i(X^i, U^i) \perp {}^p\lambda^i \geq 0 \\ h(\mathbf{X}, \mathbf{U}; \hat{\mathbf{x}}_1) = 0 \\ 0 \leq {}^s g(\mathbf{X}, \mathbf{U}) \perp {}^{\boldsymbol{s}}\boldsymbol{\lambda} \geq 0, \end{cases} \tag{4.1}$$

where, for brevity, we denote by $h(\mathbf{X}, \mathbf{U}; \hat{\mathbf{x}}_1)$ the aggregation of all equality constraints. If the second directional derivative of the Lagrangian is positive along all feasible directions at a solution of Equation (4.1)—a condition that can be checked a posteriori—this point is also a solution of the original game. This work solves trajectory games by viewing their KKT conditions through the lens of MCPs [66, Section 1.4.2].

Definition 1 *A mixed complementarity problem (MCP) is defined by the following problem data: a function $F(z) : \mathbb{R}^d \mapsto \mathbb{R}^d$, lower bounds $\ell_j \in \mathbb{R} \cup \{-\infty\}$ and upper bounds $u_j \in \mathbb{R} \cup \{\infty\}$, each for $j \in [d]$. The solution of an MCP is a vector $z^* \in \mathbb{R}^n$, such that for each element with index $j \in [d]$ one of the following equations holds:*

$$z_j^* = \ell_j, F_j(z^*) \geq 0 \tag{4.2a}$$

$$\ell_j < z_j^* < u_j, F_j(z^*) = 0 \tag{4.2b}$$

$$z_j^* = u_j, F_j(z^*) \leq 0. \tag{4.2c}$$

The parameterized KKT system of Equation (4.1) can be expressed as a *parameterized family* of MCPs with decision variables corresponding to the primal and dual variables of Equation (4.1),

$$\boldsymbol{z} = \left[\mathbf{X}^\top, \mathbf{U}^\top, \boldsymbol{\mu}^\top, {}^p\lambda^{1\top}, \ldots, {}^p\lambda^{N\top}, {}^{\boldsymbol{s}}\boldsymbol{\lambda}^\top\right]^\top,$$

and problem data

$$F(\boldsymbol{z}; \boldsymbol{\theta}) = \begin{bmatrix} \nabla_{(X^1,U^1)} \mathcal{L}^1 \\ \vdots \\ \nabla_{(X^N,U^N)} \mathcal{L}^N \\ h \\ {}^p g^1 \\ \vdots \\ {}^p g^N \\ {}^s g \end{bmatrix}, \quad \ell = \begin{bmatrix} -\infty \\ \vdots \\ -\infty \\ -\infty \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad u = \begin{bmatrix} \infty \\ \vdots \\ \infty \\ \infty \\ \infty \\ \vdots \\ \infty \\ \infty \end{bmatrix}, \tag{4.3}$$

where, by slight abuse of notation, we overload F to be parametrized by $\boldsymbol{\theta}$ via $\mathcal{L}^i$ and use ∞ to denote elements for which upper or lower bounds are dropped.

4.2. Differentiation of an MCP Solver

An MCP solver may be viewed as a function, mapping problem data to a solution vector. Taking this perspective, for a parameterized family of MCPs as in Eq. (4.3), this section wishes to compute the function's derivatives to answer the following question: How does the solution $\boldsymbol{z}^*$ respond to local changes of the problem parameters $\boldsymbol{\theta}$?

4.2.1. The Nominal Case

Let $\Psi(\boldsymbol{\theta}) := (F(\cdot;\boldsymbol{\theta}), \ell, u)$ denote an MCP parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$ and let $\boldsymbol{z}^* \in \mathbb{R}^n$ denote a solution of that MCP, which is implicitly a function of $\boldsymbol{\theta}$. For this nominal case, we consider only solutions at which *strict complementarity* holds, i.e., any instances of Eq. (4.2a) and Eq. (4.2c) occur with strict inequalities. We shall relax this assumption later. If F is smooth, i.e., $F(\cdot;\boldsymbol{\theta}), F(\boldsymbol{z}^*;\cdot) \in C^1$, we can recover the Jacobian matrix $\nabla_{\boldsymbol{\theta}} \boldsymbol{z}^* = \left(\frac{\partial z_j^*}{\partial \theta_k}\right) \in \mathbb{R}^{n \times p}$ by distinguishing two possible cases. For brevity, below, gradients are understood to be evaluated at $\boldsymbol{z}^*$ and $\boldsymbol{\theta}$.

a) Active bounds Consider first the elements z_j^* that are either at their lower or upper bound, i.e., z_j^* satisfies Eq. (4.2a) or Eq. (4.2c). Since strict complementarity holds at the solution, $F_j(\boldsymbol{z}^*;\boldsymbol{\theta})$ must be bounded away from zero with a finite margin. Hence, the smoothness of F guarantees that a local perturbation of $\boldsymbol{\theta}$ will retain the sign of $F_j(\boldsymbol{z}^*;\boldsymbol{\theta})$. As a result, z_j^* remains at its bound and, locally, the gradient is identically zero. Let $\tilde{\mathcal{I}} := \{k \in [n] \mid z_k^* = \ell_k \vee z_k^* = u_k\}$ denote the index set of all elements matching this condition and $\tilde{z}^* := [\boldsymbol{z}^*]_{\tilde{\mathcal{I}}}$ denote the solution vector reduced to that set. Trivially, then, the Jacobian of this vector vanishes, i.e., $\nabla_{\boldsymbol{\theta}} \tilde{z}^* = 0$.

b) Inactive bounds The second case comprises elements that are strictly between the bounds, i.e., z_j^* satisfying Eq. (4.2b). In this case, under mild assumptions on F, for any local perturbation of $\boldsymbol{\theta}$ there exists a perturbed solution such that F remains at its root. Therefore, the gradient $\nabla_{\boldsymbol{\theta}} z_j^*$ for these elements is generally non-zero, and we can compute it via the implicit function theorem (IFT). Let $\bar{\mathcal{I}} := \{k \in [n] \mid F_k(\boldsymbol{z}^*;\boldsymbol{\theta}) = 0, \ell_k < z_k^* < u_k\}$ be the index set of all elements satisfying case (b) and let

$$\bar{z}^* := [\boldsymbol{z}^*]_{\bar{\mathcal{I}}}, \quad \bar{F}(\boldsymbol{z}^*, \boldsymbol{\theta}) := [F(\boldsymbol{z}^*;\boldsymbol{\theta})]_{\bar{\mathcal{I}}} \tag{4.4}$$

denote the solution vector and its complement reduced to said index set. By the IFT, the relationship between parameters $\boldsymbol{\theta}$ and solution $\boldsymbol{z}^*(\boldsymbol{\theta})$ is characterized by the stationarity of $\bar{F}$:

$$0 = \nabla_{\boldsymbol{\theta}} \left[\bar{F}(\boldsymbol{z}^*(\boldsymbol{\theta}), \boldsymbol{\theta})\right] = \nabla_{\boldsymbol{\theta}} \bar{F} + (\nabla_{\bar{z}^*} \bar{F})(\nabla_{\boldsymbol{\theta}} \bar{z}^*) + (\nabla_{\tilde{z}^*} \bar{F}) \underbrace{(\nabla_{\boldsymbol{\theta}} \tilde{z}^*)}_{\equiv 0} \tag{4.5}$$

Note that, as per the discussion in case (a), the last term in this equation is identically zero. Hence, if the Jacobian $\nabla_{\bar{z}^*} \bar{F}$ is invertible, we recover the derivatives as the unique solution of the above system of equations,

$$\nabla_{\boldsymbol{\theta}} \bar{z}^* = -\left(\nabla_{\bar{z}^*} \bar{F}\right)^{-1} (\nabla_{\boldsymbol{\theta}} \bar{F}). \tag{4.6}$$

Note that Eq. (4.5) may not always have a unique solution, in which case Eq. (4.6) cannot be evaluated. One instance of this case is given by $\boldsymbol{z}^*$ being a non-isolated solution of $\Psi(\boldsymbol{\theta})$ in which case derivatives are not defined. The section below discusses practical considerations for this special case.

4.2.2. Remarks on Assumptions and Practical Realization for Special Cases

The above derivation of gradients for the nominal case involves several assumptions on the structure of the problem. This section discusses them and considerations to improve numerical robustness for practical realization of this approach below. We note that both special cases (a) and (b) discussed hereafter are rare in practice. In fact, across 100 simulations of the 2-player tracking game example in Section 5.4, neither of them occurred.

a) Weak Complementarity The nominal case discussed above assumes strict complementarity at the solution. In the context of GNEPs, this entails that constraints at the solution must either be inactive or strongly active. If this assumption does not hold, the derivative of the MCP is not defined. Nevertheless, we can still compute subderivatives at $\boldsymbol{\theta}$. Let the set of all indices for which this condition holds be denoted by $\hat{\mathcal{I}} := \{k \in [n] \mid F_k(\boldsymbol{z}^*; \boldsymbol{\theta}) = 0 \wedge z_k^* \in \{\ell_k, u_k\}\}$. Then by selecting a subset of $\hat{\mathcal{I}}$ and including it in $\bar{\mathcal{I}}$ for evaluation of Eq. (4.6), we recover a subderivative. For the experiments conducted in this work, we found good practical performance by resolving this rare ambiguity heuristically. That is, we include all of the weakly active indices $\hat{\mathcal{I}}$ in the inactive set $\bar{\mathcal{I}}$. Intuitively, this heuristic can be understood as being "optimistic" about degrees of freedom.

b) Invertibility The evaluation Eq. (4.6) requires invertibility of $\nabla_{\bar{z}^*} \bar{F}$. To this end, we compute the least-squares solution of Eq. (4.5) rather than explicitly inverting $\nabla_{\bar{z}} \bar{F}$.

c) Smoothness A key assumption underpinning the above derivation is sufficient smoothness of F in both arguments. In the context of MCPs, this amounts to assuming that Lagrangians are twice-differentiable in the trajectory of each player and once-differentiable in the parameters $\hat{\mathbf{x}}_1$ and θ^i. For the primal decision variables, this level of smoothness is common to any second-order method and is also required by gradient-based MCP solvers that we use in this work [67]. Therefore, the only added design constraint is smoothness in cost parameters, which is also found in related work in inverse games [46] and inverse optimal control [41].

4.3. Model-Predictive Game Play with Gradient Descent

Finally, this section presents the pipeline for adaptive game play against opponents with unknown objectives. The adaptive MPGP scheme is summarized in Algorithm 1. At each time step, it first updates the estimate of the parameters by approximating the inverse game in Eq. (3.3) via gradient descent. To obtain an unconstrained optimization problem, we substitute the constraints in Eq. (3.3) with the proposed differentiable game solver. Following the discussion of Eq. (4.3), we denote by $\boldsymbol{z}^*(\boldsymbol{\theta})$ the solution of the MCP formulation of the game parameterized by $\boldsymbol{\theta}$. Furthermore, by slight abuse of notation, we overload $\mathbf{X}(\boldsymbol{z}^*), \mathbf{U}(\boldsymbol{z}^*)$ to denote functions that extract the state and input vectors from $\boldsymbol{z}^*$. Then, the inverse game of Eq. (3.3) can be written as unconstrained optimization,

$$\max_{\boldsymbol{\theta}} \quad p(\mathbf{Y} \mid \mathbf{X}(\boldsymbol{z}^*(\boldsymbol{\theta})), \mathbf{U}(\boldsymbol{z}^*(\boldsymbol{\theta}))). \tag{4.7}$$

Online, we approximate solutions to this problem by taking gradient descent steps on the negative logarithm of this objective, with gradients computed by chain rule,

$$\nabla_{\boldsymbol{\theta}} \left[p(\mathbf{Y} \mid \mathbf{X}(\boldsymbol{z}^*(\boldsymbol{\theta})), \mathbf{U}(\boldsymbol{z}^*(\boldsymbol{\theta}))\right] = (\nabla_{\mathbf{X}} p)(\nabla_{\boldsymbol{z}^*} \mathbf{X})(\nabla_{\boldsymbol{\theta}} \boldsymbol{z}^*) + (\nabla_{\mathbf{U}} p)(\nabla_{\boldsymbol{z}^*} \mathbf{U})(\nabla_{\boldsymbol{\theta}} \boldsymbol{z}^*). \tag{4.8}$$

Here, the only non-trivial term is $\nabla_{\boldsymbol{\theta}} \boldsymbol{z}^*$, whose computation was discussed in Section 4.2. To reduce the computational cost, we warm-start using the estimate of the previous time step and terminate early if a maximum number of steps is reached. Then, we solve a forward game parametrized by the estimated $\tilde{\boldsymbol{\theta}}$ to compute control commands. We execute the first control input for the ego agent and repeat the procedure.

Algorithm 1: Adaptive MPGP

Hyper-parameters: stopping tolerance: stop_tol, learning rate: lr
Input: initial $\tilde{\boldsymbol{\theta}}$, current observation buffer $\mathbf{Y}$, new observation $\mathbf{y}$
$\mathbf{Y} \leftarrow$ updateBuffer($\mathbf{Y}, \mathbf{y}$)
`/* inverse game approximation */`
while not stop_tol and not max_steps_reached **do**
  $(\boldsymbol{z}^*, \nabla_{\boldsymbol{\theta}} \boldsymbol{z}^*) \leftarrow$ solveDiffMCP($\tilde{\boldsymbol{\theta}}$) ▷ sec. 4.2
  $\nabla_{\boldsymbol{\theta}} p \leftarrow$ composeGradient($\boldsymbol{z}^*, \nabla_{\boldsymbol{\theta}} \boldsymbol{z}^*, \mathbf{Y}$) ▷ eq. (4.8)
  $\tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}} p \cdot$ lr
end
$\boldsymbol{z}^* \leftarrow$ solveMCP($\tilde{\boldsymbol{\theta}}$) ▷ forward game, eq. (4.3)
applyFirstEgoInput($\boldsymbol{z}^*$)
return $\tilde{\boldsymbol{\theta}}, \mathbf{Y}$

4.4. Model-Predictive Game Play: Extension

A computation graph of the MPGP pipeline is shown in Figure 4.1. An essential benefit of the proposed framework is that the entire computation graph is differentiable. The trajectory game solver can be effectively viewed as an NN layer encoding a dynamic game. Hence, the pipeline supports direct integration with other differentiable elements like NNs, as shown in Figure 4.2. Section 5.5.2 shows a proof-of-concept experiment using this idea to accelerate the online computation of the proposed method. Moreover, a combination of the proposed solver with a generative model is presented in Chapter 6.

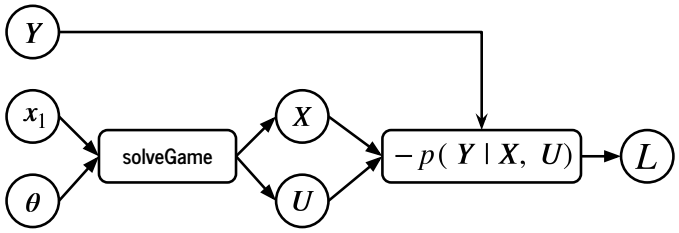


Figure 4.1: Computation graph of the model-predictive game-play pipeline.

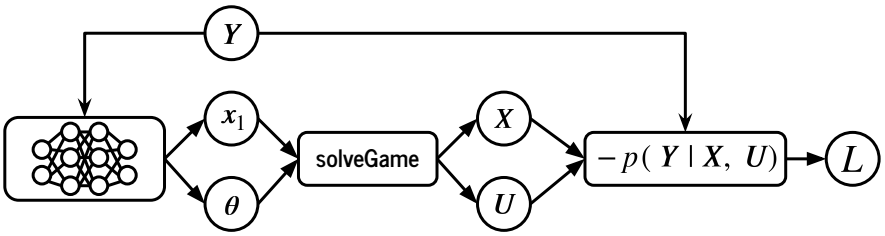


Figure 4.2: Extension of the model-predictive game-play pipeline: a neural network is employed to propose an initial guess for the game solver. The whole pipeline is end-to-end differentiable.

4.5. Summary

To summarize, the main points from this chapter are:

- The proposed MPGP pipeline solves forward games by casting them as equivalent MCPs, which has efficient solution approaches. This work uses a Newton-type method [67] to solve them.
- Inverse game solves of the proposed framework are enabled by differentiation through a forward game solver. By encoding the equilibrium constraints into the proposed differentiable game solver, inverse games can be cast as unconstrained optimization problems and be solved via online gradient descent.
- Online, the MPGP framework solves inverse games to estimate unknown players' objectives and computes strategic motion plans by solving forward games induced by the estimated objectives.
- Same as the existing MLE inverse game solutions, the proposed method also handles partial-state observations and noise-corrupted data. Beyond that, this work further handles general inequality constraints in underlying forward games and supports end-to-end combinations with neural networks.

5 Evaluation

This chapter evaluates the proposed adaptive game-theoretic planner in two simulation traffic scenarios to demonstrate the performance of the approach compared with existing game-theoretic and non-game-theoretic model-predictive control (MPC) baselines. Furthermore, the chapter showcases the real-time planning capability of the proposed planner deployed on a real hardware platform. The experiments below are designed to support the key claims that the proposed method (i) outperforms both game-theoretic and non-game-theoretic baselines in highly interactive scenarios, (ii) can be combined with other differentiable components such as NNs, and (iii) is sufficiently fast for real-time planning on a hardware platform.

5.1. Implementation & Supplementary Video

The proposed adaptive solver is implemented in the Julia Programming Language [68]. MCPs are solved using a Newton-style method presented in [67]. Besides the problem data in Equation (4.3), the Jacobian of the vector-valued function $F(z)$ is required. This information is acquired from a symbolic implementation using `Symbolics.jl` [69].

The libraries `ForwardDiff.jl` [70] and `ChainRulesCore.jl` are employed for implicit differentiation through the solver, and `Zygote.jl` [71] is used for auto-differentiation of the pipeline. For the combination with NNs, this work uses the NN implementation from the `Flux.jl` [72, 73] library. The hardware experiments employ Robot Operating System (ROS) [74] to communicate with and control the Clearpath Jackal robots. A supplementary video is available at https://xinjie-liu.github.io/projects/game/.

5.2. Baselines

The evaluation features the following game-theoretic and non-game-theoretic baselines:

KKT-Constrained Solver As a game-theoretic baseline, the evaluation considers the inverse game solver by Peters et al. [46]. Compared to this work, the solver by Peters et al. [46] has no support for either private or shared inequality constraints. Consequently, this baseline can be viewed as solving a simplified version of the problem in Equation (3.3), where the inequality constraints associated with the inner-level GNEP are dropped. Nonetheless, the evaluation still uses a cubic penalty term, as in Equation (5.2) and Equation (5.7), to encode soft collision avoidance. Furthermore, for a fair comparison, the evaluation only uses the baseline to *estimate* the objectives but computes control commands from a GNEP considering all constraints.

Heuristic Estimation MPGP To study the effect of online intent inference itself, an ablation study for the ramp merging scenario is conducted. The evaluation compares against a game-theoretic baseline that assumes a fixed intent for all opponents. This fixed intent is recovered by taking each agent's initial lane and velocity as a heuristic preference estimate. This baseline conducts game-theoretic planning while opponents' objectives are rather inaccurate.

MPC with Constant-Velocity Predictions For comparison against non-game-theoretic approaches, this evaluation employs an MPC baseline. This baseline assumes that opponents move with constant velocity as observed at the latest time step. While this prediction model is rather simple, it is a popular approach that is known to provide good prediction performance both in pedestrian prediction [55] and highway driving [75] scenarios. In highway settings like Section 5.5 will discuss below, it has been demonstrated to be competitive even against much more elaborate learning-based predictions [75]. This work uses this baseline as a representative method for predictive planning approaches that do not explicitly model interaction. Although selecting a task-specific prediction model could change the prediction performance, task-specific models would also lead to arbitrary choices. Given the popularity and simplicity of constant-velocity MPC, the author therefore believes that it provides a valuable reference performance to many readers.

5.3. Parameters

Parameter values used in the evaluation are shown in Table 5.1. To ensure a fair comparison, the evaluation uses the same MCP backend [67] to solve all GNEPs and optimization problems with a default convergence tolerance of 1×10^{-6}. Furthermore, all planners utilize the same planning horizon and history buffer size of 10 time steps with a time-discretization of $0.1\,\mathrm{s}$. For the iterative MLE solve procedure in the 2-player tracking example and the ramp merging scenario below, the evaluation employs a learning rate of 2×10^{-2} for objective parameters and 1×10^{-3} for initial states. The maximum likelihood estimation iteration is terminated when the norm of the parameter update step is smaller than 1×10^{-4}, or after a maximum of 30 steps. Finally, opponent behavior is generated by solving a separate ground-truth game whose parameters are hidden from the ego agent.

Table 5.1: Parameter values used in the experiments.

Parameter	Value
MCP convergence tolerance	1×10^{-6}
Planning horizon	$1\,\mathrm{s}$
Learning rate: objectives	2×10^{-2}
Learning rate: initial states	1×10^{-3}
MLE convergence tolerance	1×10^{-4}
MLE maximal steps	Max. 30

5.4. Two-Player Tracking Game: Inferring the Opponent's Goal Position

First, a toy example between 2 players is introduced. This example is designed to concretize the concepts and to test the inference accuracy and convergence of the proposed method in an intuitive setting. An ego robot is tasked to track a target robot while the target robot wishes to drive efficiently to their goal position, which is initially unknown to the ego. Both players

share collision avoidance responsibility. Therefore, in order to effectively and safely stay as close as possible to the target robot, the ego robot infers the opponent's goal positions on the fly and tracks them.

5.4.1. Game Formulation

Let each agent's dynamics be characterized by a planar double-integrator in Equation (5.1):

$$x_{t+1}^i = \begin{bmatrix} 1.0 & 0.0 & \Delta t & 0.0 \\ 0.0 & 1.0 & 0.0 & \Delta t \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} x_t^i + \begin{bmatrix} 0.5\Delta t^2 & 0.0 \\ 0.0 & 0.5\Delta t^2 \\ \Delta t & 0.0 \\ 0.0 & \Delta t \end{bmatrix} u_t^i, \tag{5.1}$$

where states $x_t^i = (p_{x,t}^i, p_{y,t}^i, v_{x,t}^i, v_{y,t}^i)$ are position and velocity, and control inputs $u_t^i = (a_{x,t}^i, a_{y,t}^i)$ are acceleration in horizontal and vertical axes in a Cartesian frame. We define the game's state as the concatenation of the two players' individual states $\mathbf{x}_t := (x_t^1, x_t^2)$.

Each player's objective is characterized by an individual cost

$$J^i = \sum_{t=1}^{T-1} \|p_{t+1}^i - p_{\mathsf{goal}}^i\|_2^2 + 0.1\|u_t^i\|_2^2 + 50\,\mathsf{max}(0, d_{\mathsf{min}} - \|p_{t+1}^i - p_{t+1}^{\neg i}\|_2)^3, \tag{5.2}$$

where we set $p_{\mathsf{goal}}^1 = p_t^2$ so that player 1, the tracking robot, is tasked to track player 2, the target robot. Player 2 has a fixed goal point p_{goal}^2. Both agents wish to get to their goal position efficiently while avoiding proximity beyond a minimal distance d_{min}. Players also have shared collision avoidance constraints ${}^s g_{t+1}(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) = \|p_{t+1}^1 - p_{t+1}^2\|_2 - d_{\mathsf{min}} \geq 0, \forall t \in [T-1]$ and private bounds on state and controls ${}^p g^i(X^i, U^i)$. Agents need to negotiate and find an underlying equilibrium strategy in this noncooperative game, as no one wants to deviate from the direct path to their goal.

We further assign the tracker (player 1) to be the ego agent and parameterize the game with the goal position of the target robot $\theta^2 = p_{\mathsf{goal}}^2$. That is, the tracker does not know the target agent's goal and tries to infer this parameter from position observations. To ensure that Equation (3.3) remains tractable, the ego agent maintains only a fixed-length buffer of observed opponent's positions. Note that solving the inverse game requires solving games rather than optimal control problems at the inner level to account for the noncooperative nature of observed interactions, which is different from inverse optimal control even in the 2-player case.

We employ an isotropic Gaussian observation model $\mathcal{N}(\mathbf{Y}, \sigma^2 I)$, with its mean being the observation data. We further represent it with an equivalent negative log-likelihood objective $-\log(\prod_{j=1}^{P} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_j(\mathbf{X},\mathbf{U})-Y_j)^2}{2\sigma^2}})$, where $r(\mathbf{X}, \mathbf{U})$ maps $(\mathbf{X}, \mathbf{U})$ to the corresponding sequence of expected positions, j denotes the jth element of the vectors, and P is the total element number of $\mathbf{Y}$ and $r(\mathbf{X}, \mathbf{U})$. Hence, minimizing the observation likelihood loss can be further reduced to minimization of the squared distance between the inferred trajectory and the observation:

$$\arg\min_{\mathbf{X},\mathbf{U}} -\log(\prod_{j=1}^{P} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_j(\mathbf{X},\mathbf{U})-Y_j)^2}{2\sigma^2}}) = \arg\min_{\mathbf{X},\mathbf{U}} -\sum_{j=1}^{P} \log(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_j(\mathbf{X},\mathbf{U})-Y_j)^2}{2\sigma^2}}) \tag{5.3}$$

$$= \arg\min_{\mathbf{X},\mathbf{U}} -P\log(\frac{1}{\sigma\sqrt{2\pi}}) + \sum_{j=1}^{P} \frac{(r_j(\mathbf{X},\mathbf{U}) - Y_j)^2}{2\sigma^2} \tag{5.4}$$

$$= \arg\min_{\mathbf{X},\mathbf{U}} \|r(\mathbf{X},\mathbf{U}) - \mathbf{Y}\|_2^2. \tag{5.5}$$

5.4.2. Monte Carlo Study

This section presents a Monte Carlo study to evaluate the proposed approach in the 2-player tracking example. We sample the opponent's intent—i.e., their unknown goal position in Equation (5.2)— uniformly from the environment. Each approach is evaluated with 100 trials, with each trial consisting of 70 time steps, which is sufficient for the ground truth behavior to converge to an equilibrium state—both players stop close to the target robot's goal position. Partial observations comprise the position of each agent.

Qualitative Results

This section shows one trial of the tracking experiment using the proposed approach versus the MPC baseline. Figure 5.1 shows the resultant behavior using the proposed planner. The tracking robot actively reasons about the opponent's goal position and approaches the inferred goal. The target robot thinks that the tracking robot understands the interaction between them. So the target robot gives way to their opponent, and the tracking robot successfully figures out the hidden goal and stops at that position to smartly keep close to the target robot. The two robots come to a stable state at the end of the episode. In contrast, as Figure 5.2 shows, the MPC baseline does not reason about the interaction and assumes the opponent will keep driving at the observed velocity. So they keep chasing the target robot without stopping.

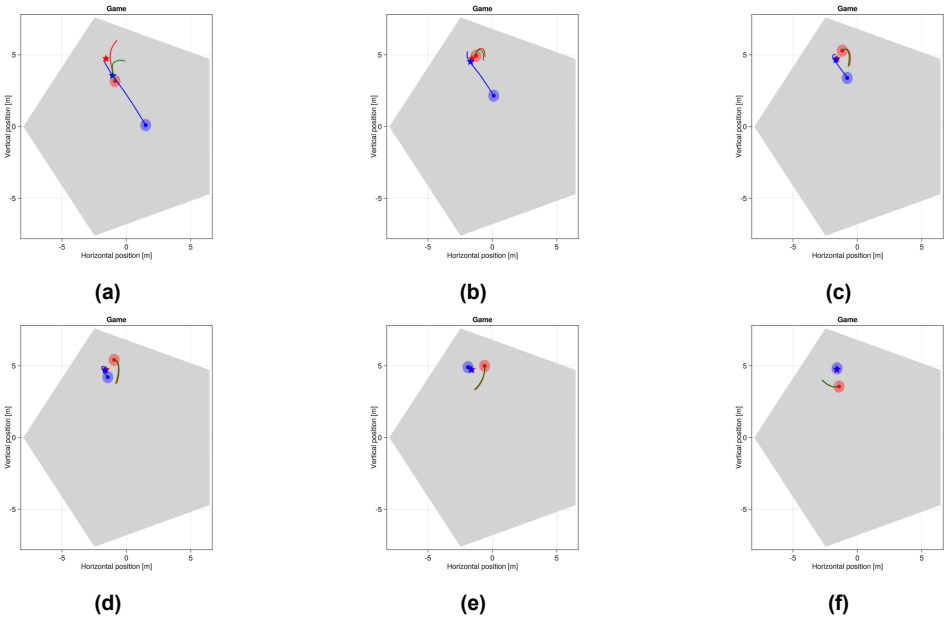


Figure 5.1: A tracking game between two robots with the ego agent (blue disc) using the proposed method. The target robot (red disc) drives to their goal position (red star), which is hidden from the tracking robot (blue disc). The tracking robot infers the target robot's goal from observed position sequences and tries to stay as close as possible to the target robot. The inferred goal is shown as a blue star, and the predicted motion plan by the tracking robot is shown in green. The actual plans of the two robots are shown in blue and red. With objective inference and strategic reasoning, the tracking robot quickly figures out the target robot's goal position and stops at that point to smartly keep close to the opponent. The two robots come to a stable state.

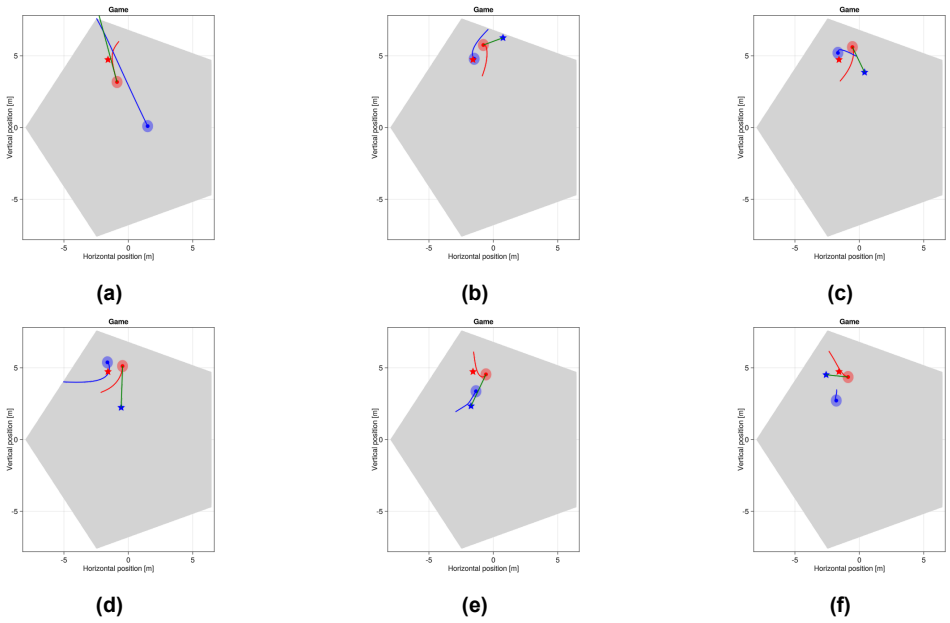


Figure 5.2: The same tracking game with the ego robot using constant-velocity model-predictive control. Without understanding the interaction and objective inference, the tracking robot assumes the target robot will drive at the current speed and keeps chasing them. The two robots do not come to a stable state.

Quantitative Results

Figure 5.4 summarizes the quantitative results for the 2-player running example. For this evaluation, we filter out any runs for which a solver resulted in a collision. For the proposed solver, the KKT-constrained baseline, and the MPC baseline, this amounts to $2, 2$ and 13 out of 100 episodes, respectively (depicted in Figure 5.3).

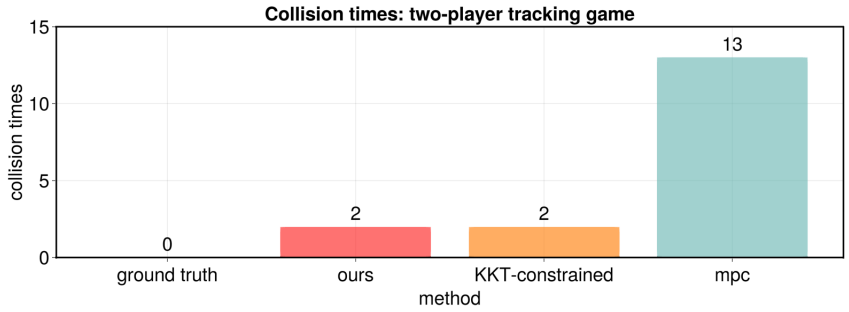


Figure 5.3: Collision times of each approach in the 100-trial Monte Carlo study of the 2-player tracking game.

Figures 5.4(a-b) show the prediction error of the goal position and opponent's trajectory, each of which is measured by ℓ^2-norm. Since the MPC baseline does not explicitly reason about costs of others, parameter inference error is not reported for it in Figure 5.4a. As evident from this visualization, both game-theoretic methods give relatively accurate parameter estimates and trajectory predictions. Among these methods, the proposed solver converges more quickly and consistently yields a lower error. By contrast, MPC gives inferior prediction performance with reduced errors only in trivial cases, when the target robot is already at the goal. Figure 5.4c shows the distribution of costs incurred by the ego agent for the same set of experiments. Again, game-theoretic methods yield better performance and the proposed method outperforms the baselines with more consistent and robust behaviors, indicated by fewer outliers and lower variance in performance.

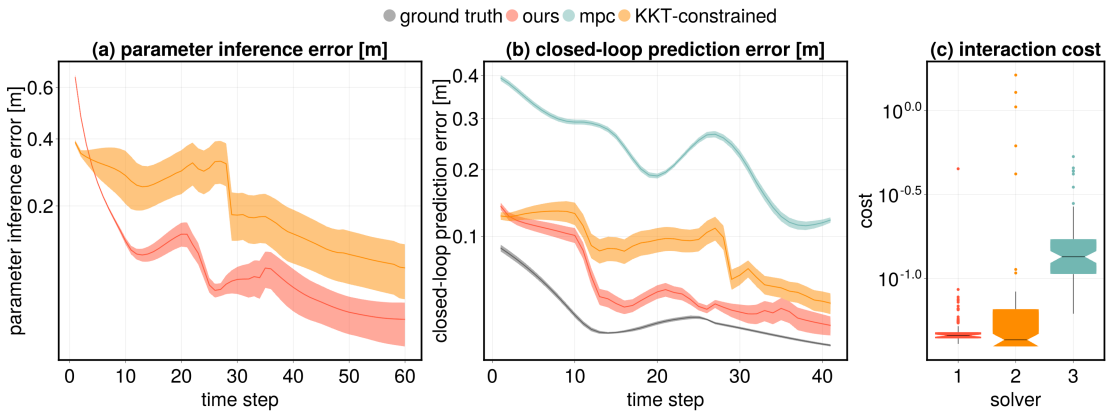


Figure 5.4: Monte Carlo study of the 2-player tracking game for 100 trials. Solid lines and ribbons in (a) and (b) indicate the mean and standard error of the mean. Cost distributions in (c) are normalized by subtracting ground truth costs.

5.5. Ramp-Merging Game: Inferring the Opponents' Desired Driving States

To study the scalability of the proposed approach to more complex scenarios and to support the claim that the solver outperforms the baselines in highly interactive settings, this section tests the proposed method in a ramp-merging scenario with varying numbers of players. This experiment is inspired by the setup used by Cleac'h, Schwager, Manchester, et al. [13] and is schematically visualized in Figure 5.5. An ego agent (red) wishes to merge onto a busy road populated by other agents (gray). To enable safe interactions, the proposed adaptive planner infers other drivers' desired driving speed and lateral position on the fly and conducts game-theoretic planning based on the estimates. In order to enforce more interactions, all agents have to stop in front of a traffic light at the end of the road.

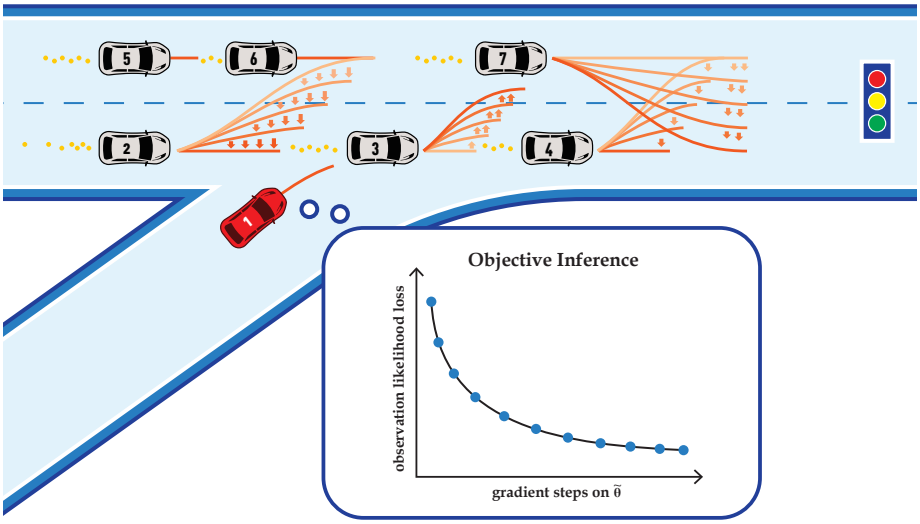


Figure 5.5: An ego agent (red) merging onto a busy road populated by six surrounding vehicles whose preferences for travel velocity and lane are initially unknown. The proposed approach adapts the ego agent's strategy by inferring opponents' intention parameters $\tilde{\theta}$ from partial-state observations.

5.5.1. Game Formulation

To test the proposed approach with nonlinear system dynamics, each player's dynamics are modeled by a discrete-time kinematic bicycle in Equation (5.6):

$$x_{t+1}^i = \begin{cases} p_{x,t}^i + \Delta t\, v_t^i \cos\psi_t^i \\ p_{y,t}^i + \Delta t\, v_t^i \sin\psi_t^i \\ v_t^i + \Delta t\, a_t^i \\ \psi_t^i + \Delta t\, \dfrac{v_t^i}{l} \tan\phi_t^i, \end{cases} \tag{5.6}$$

with the state comprising position, velocity, and orientation, i.e., $x_t^i = (p_{x,t}^i, p_{y,t}^i, v_t^i, \psi_t^i)$, controls comprising acceleration and steering angle, i.e., $u_t^i = (a_t^i, \phi_t^i)$, and l denoting the bicycle length.

Agents' individual behavior is captured by a cost function:

$$J^i = \sum_{t=1}^{T-1} \|p^i_{y,t+1} - p^i_{\text{y,lane}}\|_2^2 + \|v^i_{t+1} \cos \psi^i_{t+1} - v^i_{\text{ref}}\|_2^2 + 0.1\|u^i_t\|_2^2 + \sum_{j \in \{1,\dots,N\} \setminus \{i\}} 500 \max(0, d_{\min} - \|p^i_{t+1} - p^j_{t+1}\|_2)^3, \quad (5.7)$$

which penalizes deviation from a reference longitudinal travel velocity and target lane; i.e., $\theta^i = (v^i_{\text{ref}}, p^i_{\text{y,lane}})$. The term $500 \max(0, d_{\min} - \|p^i_{t+1} - p^j_{t+1}\|_2)^3$ denotes collision avoidance cost between player i and an opponent player j. Moreover, hard constraints are added for lane boundaries, for limits on speed, steering, and acceleration, for the traffic light, and for collision avoidance.

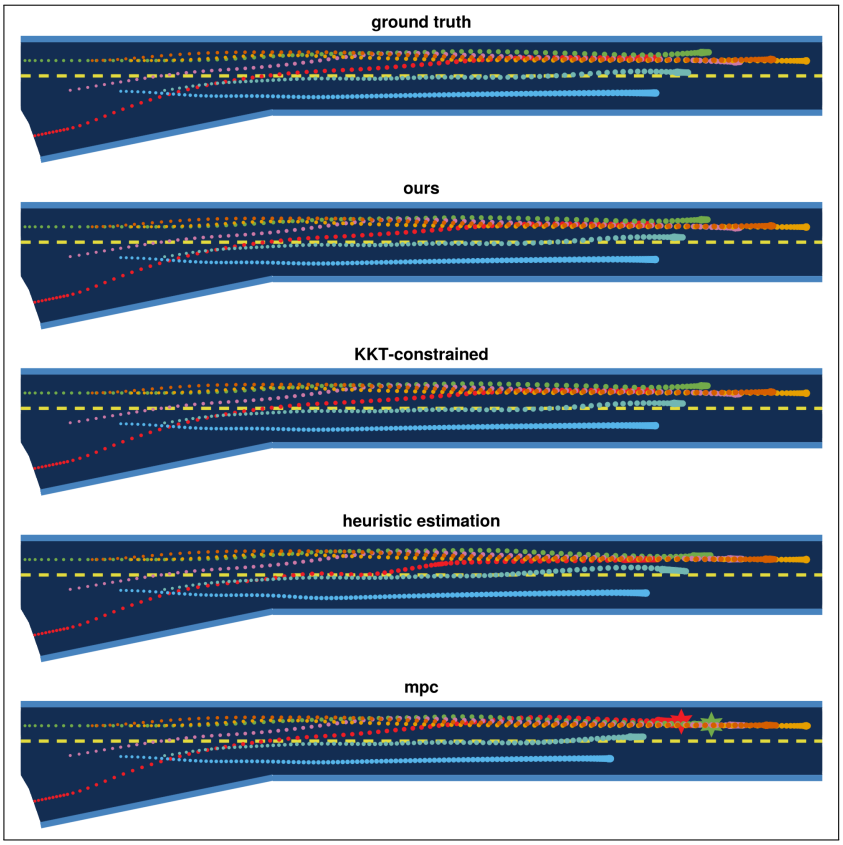
5.5.2. Monte Carlo Study

A Monte Carlo study is conducted for the ramp-merging scenario with 3, 5, and 7 players to test the performance of the proposed approach scaling from simple to complex settings. Besides the KKT-constrained baseline and the constant-velocity MPC, this section additionally compares the proposed approach against a heuristic estimation baseline as an ablation study to test the importance of online intent inference.

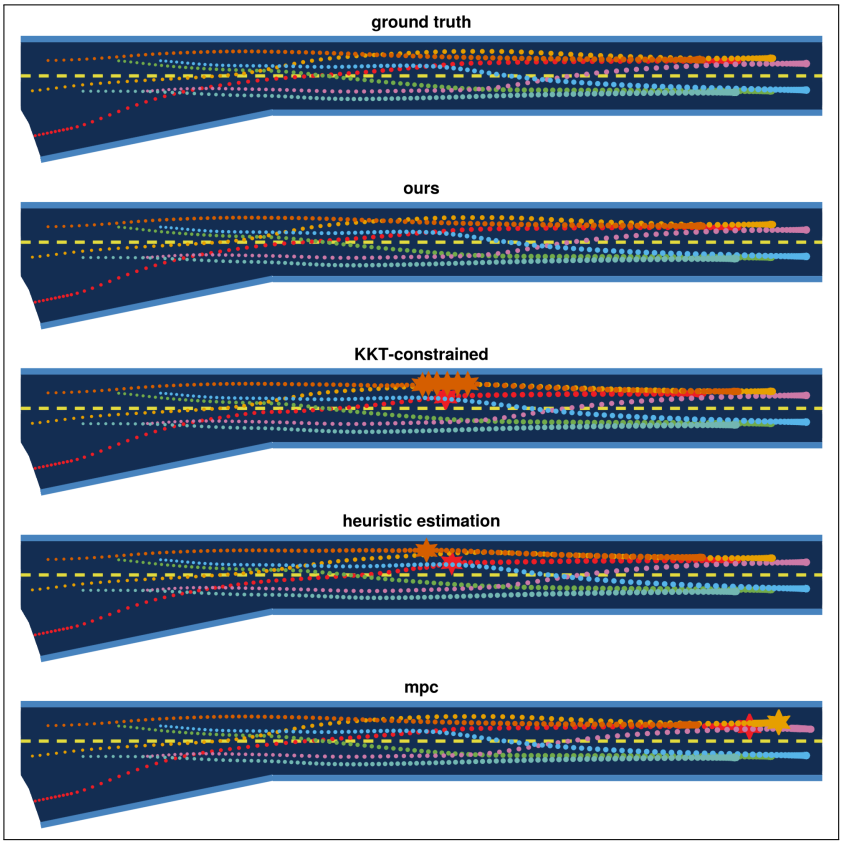
To quantify the task performance, the evaluation reports costs as an indicator of interaction efficiency, the number of collisions as a measure of safety, the number of infeasible solves as an indicator of robustness, and trajectory and parameter error as a measure of inference accuracy. To encourage rich interaction in simulation, each agent's initial state is sampled by sampling their speed and longitudinal positions uniformly at random from the intervals from zero to maximum velocity $v_{\max}$ and four times the vehicle length l_{car}, respectively. The ego agent always starts on the ramp and wishes to drive at the highest longitudinal speed $v_{\max}$. Each opponent's intent is sampled from the uniform distribution over the two lane centers and the target speed interval $[0.4v_{\max}, v_{\max}]$. Finally, all agents are initially aligned with their current lane. Partial observations comprise the position and orientation of each agent.

Qualitative Results

This section shows the qualitative results of two episodes from the ramp-merging experiment using different approaches with 7 players, as depicted in Figure 5.6. The two episodes differ regarding players' initial states and opponents' objectives. In this dense traffic scenario, the proposed approach matches the ground truth performance the most closely. In many runs, as Figure 5.6a shows, the game-theoretic approaches all provide safe interaction behaviors. Among them, the two game-theoretic methods with objective inference are more comparable. The heuristic-estimation baseline leads to a less efficient trajectory for the ego agent (red) with a sudden deceleration in the middle of the road. Furthermore, in some settings, as Figure 5.6b shows, only the proposed approach remains safe while all the baselines lead to collisions. The MPC baseline collides significantly more often than other game-theoretic approaches in the densest traffic setting. The following sections present a detailed analysis of the methods' performances and quantitative comparisons.



(a)



(b)

Figure 5.6: Two episodes of the 7-player ramp merging scenario with different players' initial states and opponents' objectives. The ego agent's trajectory is shown in red, marker size increases with time steps, and star markers indicate collisions. In this dense traffic scenario, the proposed approach matches the ground truth performance the most closely. A video containing more trials of this experiment is included in the supplementary material, available at: https://xinjie-liu.github.io/projects/game/.

Varying Number of Players

First, the proposed approach is evaluated in the ramp-merging scenario with a varying number of players. A kinematic bicycle is used to model the system dynamics, and the objective estimation is conducted based on partial-state observation comprising the position and orientation of each agent.

Results are shown in Table 5.2. On a high level, we observe that the game-theoretic methods generally outperform the MPC baseline, especially for the settings with higher traffic density. While MPC achieves high efficiency (ego cost) in the 3-player case, it collides significantly more often than the other methods across all settings. Among the game-theoretic approaches, we observe that online inference of opponent intents—as performed by the proposed method and the KKT-constrained baseline—yields better performance than a game that uses a heuristic estimate of the intents, which renders the importance of goal estimation in these traffic games.

Within the inference-based game solvers, a Manning-Whitney U-test reveals that, in denser scenarios with 5 and 7 players, both methods achieve an ego-cost that is significantly lower than all other baselines but not significantly higher than solving the game with ground truth opponent intents. Despite this tie in terms of interaction *efficiency*, the results show a statistically significant improvement of the proposed method over the KKT-constrained baseline in terms of *safety*: in the highly interactive 7-player case, the KKT-constrained baseline collides seven times more often than the proposed method. This advantage is mostly enabled by the proposed method's ability to model inequality constraints within the inverse game. As evidenced by another experiment in Table 5.2, eliminating collision avoidance inequality constraints from inverse games in the proposed framework leads to five times more collisions than the original full framework. After elimination, the approach leads to statistically significantly higher collision times than solving games with ground truth objectives. The full framework proposed in this work is the only approach that matches the ground truth performance statistically regarding both efficiency and safety in the densest setting. The results of the Manning-Whitney U-test are presented in Appendix B.

To summarize, the proposed adaptive planner shows the best performance among all the tested approaches, and the performance advantage is more pronounced in denser scenarios.

Table 5.2: Monte Carlo studies of the ramp merging scenario depicted in Figure 5.5 for settings with 3, 5, and 7 players, each with 100 trials. Except for collision and infeasible solve times, all metrics are reported by mean and standard error of the mean. Interaction costs are normalized by subtracting the ground truth costs. Results of corresponding Manning-Whitney U-tests are shown in Appendix B.

Set.	Method	Ego cost	Opp. cost	Collision	Inf.	Traj. err. [m]	Param. err.
3 players	Ours	0.64 $\pm$ 0.36	0.06 $\pm$ 0.03	0	0	1.29 $\pm$ 0.05	0.41 $\pm$ 0.03
	KKT-con	1.85 $\pm$ 1.21	0.05 $\pm$ 0.02	0	1	1.32 $\pm$ 0.06	2.39 $\pm$ 0.11
	Heuristic	6.73 $\pm$ 2.40	0.09 $\pm$ 0.07	0	11	7.89 $\pm$ 0.26	3.96 $\pm$ 0.13
	MPC	1.50 $\pm$ 0.45	0.33 $\pm$ 0.07	28	218	2.40 $\pm$ 0.11	n/a
5 players	Ours	0.56 $\pm$ 0.43	0.16 $\pm$ 0.06	0	2	1.66 $\pm$ 0.07	0.47 $\pm$ 0.03
	KKT-con	0.07 $\pm$ 0.32	0.06 $\pm$ 0.02	1	4	1.70 $\pm$ 0.06	2.15 $\pm$ 0.06
	Heuristic	2.06 $\pm$ 0.44	0.35 $\pm$ 0.10	5	25	8.05 $\pm$ 0.19	2.91 $\pm$ 0.07
	MPC	5.73 $\pm$ 2.91	0.42 $\pm$ 0.13	44	552	2.87 $\pm$ 0.13	n/a
7 players	Ours	1.60 $\pm$ 1.19	0.06 $\pm$ 0.02	1	1	1.89 $\pm$ 0.05	0.46 $\pm$ 0.02
	Ours (without inequalities)	2.10 $\pm$ 1.33	0.14 $\pm$ 0.08	5	1	1.95 $\pm$ 0.06	0.47 $\pm$ 0.02
	KKT-con	3.11 $\pm$ 1.72	0.09 $\pm$ 0.04	7	22	2.01 $\pm$ 0.06	1.93 $\pm$ 0.03
	Heuristic	6.60 $\pm$ 1.67	0.27 $\pm$ 0.06	8	8	8.18 $\pm$ 0.15	2.44 $\pm$ 0.05
	MPC	8.41 $\pm$ 1.45	0.59 $\pm$ 0.09	43	848	3.07 $\pm$ 0.08	n/a

Partial & Full Observation

An advantage of MLE methods over KKT residual approaches is that they only require partial-state observation. This section studies how the inference performance is affected by the observation type. In particular, this section compares the parameter inference and trajectory prediction performance of the proposed adaptive solver and the KKT-constrained baseline using full-state versus partial-state observation. The partial-state observation is the same as in Section 5.5.2, comprised of the position and orientation of each agent.

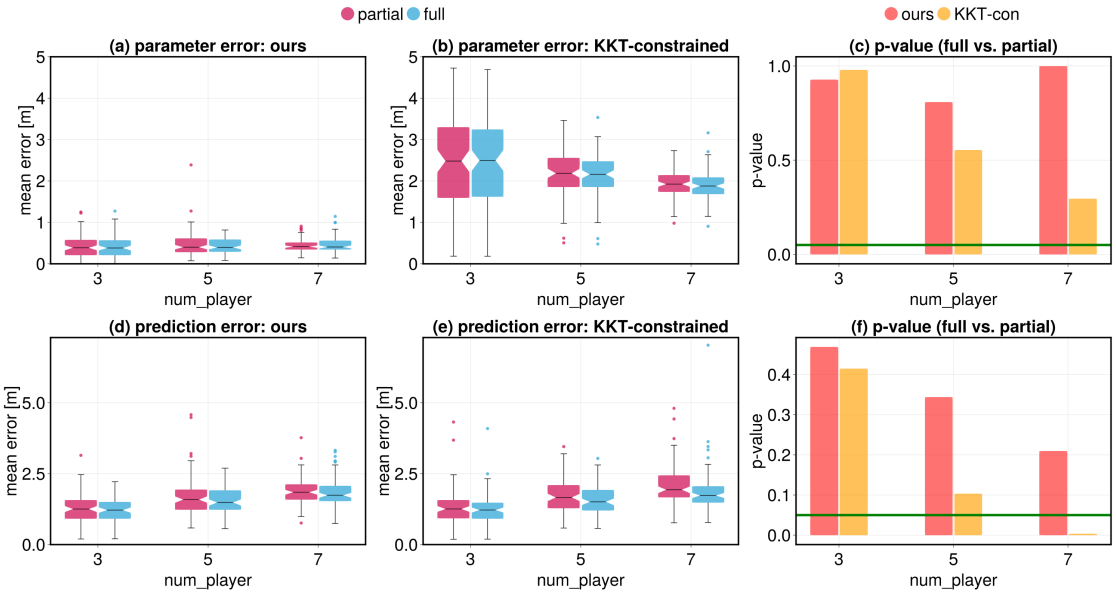


Figure 5.7: Performance of the proposed method and the KKT-constrained baseline using partial-state versus full observation. The first two columns: performance gaps between partial and full observation data with the two approaches. The right-most column: the corresponding P values of a Manning-Whitney U-test. P values lower than the green lines (0.05) indicate statistical significance.

The results are shown in Figure 5.7. The first two columns show the performance of the methods in each setting using partial-state versus full-state observations, while the right-most column shows the P values from a Manning-Whitney U-test comparing the approaches using partial-state versus full-state observations in those settings. Regarding the parameter inference error, neither approach shows a statistically significant performance drop using partial-state observation compared with full observation, as indicated by P values in Figure 5.7(c) consistently higher than the significance threshold indicated by the green line across the varying number of players. However, the proposed approach achieves lower inference errors, as indicated by Figure 5.7(a-b), and shows higher robustness against partial-state observations in denser scenarios, as indicated by the higher P values in Figure 5.7(c). Noticeably, in Figure 5.7(c), the P-value gap between the two approaches becomes more pronounced in scenarios with 5 and 7 players, indicating that the KKT-constrained baseline is less robust against partial-state observation in dense scenarios.

The trend mentioned above is more significant regarding the trajectory prediction error. In particular, in the 7-player ramp-merging case, the KKT-constrained baseline suffers a statistically significant performance drop using partial-state observation compared to the full-state observation, as shown in Figure 5.7(f). By contrast, the proposed approach does not show any statistically significant performance deterioration, with the P values higher than the threshold by a large margin, amounting to around 0.5, 0.35, and 0.22 for the 3, 5, and 7-player cases.

Computation Time

The computation time of each approach in Section 5.5.2 is measured, as shown in Table 5.3. The inference-based game solvers generally have a higher run time than the remaining methods due to the added complexity. Within the inference methods, the proposed method is only marginally slower than the KKT-constrained baseline, despite solving a more complex problem that includes inequality constraints. The average number of MLE updates for the proposed method was 11.0, 19.2, and 22.7 for the 3, 5, and 7-player settings, respectively. While the current implementation of the proposed framework achieves real-time planning rates only for up to three players, we note that additional optimizations may further reduce the run time of the approach. Among such optimizations are low-level changes such as sharing memory between MLE updates as well as algorithmic changes to perform intent inference asynchronously at an update rate lower than the control rate. The following section briefly explores another algorithmic optimization.

Table 5.3: The computation time of each setting and method in Table 5.2.

Setting	Method	Time [s]	Setting	Method	Time [s]	Setting	Method	Time [s]
3 players	Ours	0.081 ± 0.002	5 players	Ours	0.29 ± 0.02	7 players	Ours	0.68 ± 0.02
	KKT-con	0.060 ± 0.002		KKT-con	0.28 ± 0.02		KKT-con	0.63 ± 0.06
	Heuristic	0.008 ± 0.001		Heuristic	0.015 ± 0.001		Heuristic	0.031 ± 0.002
	MPC	0.009 ± 0.002		MPC	0.014 ± 0.002		MPC	0.0274 ± 0.004

Combination with Neural Networks

To accelerate the online computation and to support the claim that the proposed method can be combined with other differentiable modules, this section demonstrates the integration with an NN. Figure 5.8 shows the computation graph for this extension. For this proof of concept, a two-layer feed-forward NN is employed, which takes the buffer of recent partial state observations as input and predicts other players' objectives. Training of this module is enabled by propagating the gradient of the observation likelihood loss of Equation (4.7) through the differentiable game solver to the parameters of the NN. Online, the planner uses the network's prediction as an initial guess to reduce the number of gradient steps. As summarized in Figure 5.9, this combination reduces the computation time by more than 60% while incurring only a marginal loss in performance.

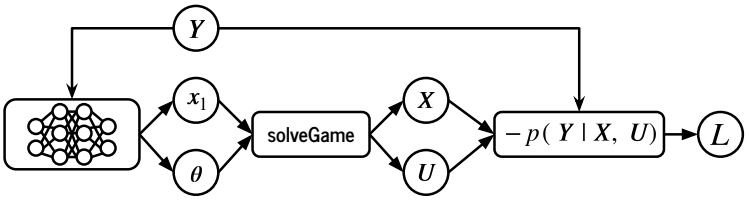
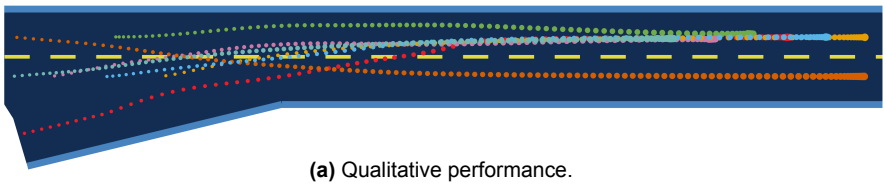


Figure 5.8: The proposed differentiable adaptive game-theoretic planner, in combination with a neural network.



(a) Qualitative performance.

Ego cost	Opp. cost	Coll.	Inf.	Traj. err. [m]	Param. err.	Time [s]
2.19 $\pm$ 1.21	0.17 $\pm$ 0.07	3	5	2.34 $\pm$ 0.08	0.91 $\pm$ 0.08	0.274 $\pm$ 0.01

(b) Quantitative performance.

Figure 5.9: Performance of the proposed solver combined with a neural network for 100 trials of the 7-player ramp merging scenario.

5.6. Hardware Demonstration: Two-Player Tracking Game

To test the real-world applicability of the proposed method and to support the claim that the method is sufficiently fast for hardware deployment, this section demonstrates the 2-player tracking game in Section 5.4 on Clearpath Jackal ground robots. Plans are computed online on a mobile i7 CPU. The experiments generate plans using the point mass dynamics in Equation (5.1) with a velocity constraint of $0.8\,\mathrm{m\,s^{-1}}$ and realize low-level control via a feedback controller [76]. A video of these hardware demonstrations is included in the supplementary material at https://xinjie-liu.github.io/projects/game/. The average computation time in both experiments was $0.035\,\mathrm{s}$.

5.6.1. Two Jackal Robots

First, the algorithm is implemented on a pair of Jackal robots, as in Figure 5.10, where one robot is tasked to track another. We observe that the proposed adaptive MPGP planner enables the tracking robot to infer the unknown goal position to track the target while avoiding collisions. In each episode, the unknown goal position is robustly recovered via online gradient descent. It is worth mentioning that with game-theoretic reasoning, the tracking robot behaves “socially compliant” and actively gives way to the target robot if needed.

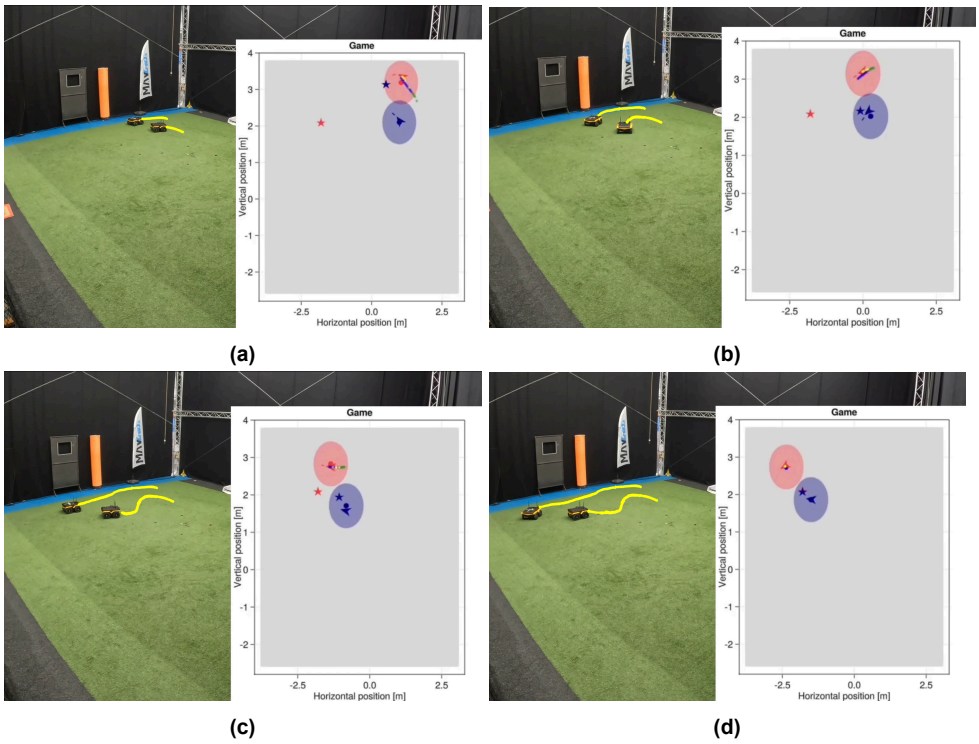


Figure 5.10: A tracking game between two Jackal ground robots. The figures show the target robot (red disc) with its hidden goal position (red star) and the tracker (dark blue disc) with its estimated goal position (dark blue star). Historic positions are shown in yellow, estimated states are shown in green, and corresponding predictions are visualized in indigo. A video of this experiment is included in the supplementary material, available at: https://xinjie-liu.github.io/projects/game/.

5.6.2. Human-Robot Interaction

In order to test the robustness of the proposed method against noise in agents' behaviors [77], this section demonstrates a human-robot interaction experiment, where the target robot in the tracking game is replaced with a human. A time-lapse of this experiment is shown in Figure 5.11. In the experiment, we observe that the adaptive planner is capable of recovering a good goal estimation and interacting with the human under noisy observation. The robot effectively tracks the human while actively giving way to the human to avoid collisions.

Figure 5.11: Time-lapse of the tracking game in which a Jackal robot tracks a human. A video of this experiment is included in the supplementary material, available at: https://xinjie-liu.github.io/projects/game/.

5.7. Discussion and Conclusion

This chapter has conducted three sets of qualitative and quantitative experiments to validate the effectiveness of the proposed MPGP framework.

In a simulated 2-player tracking game in Section 5.4, qualitative results suggest that reasoning about interactions and the opponent's objectives can be essential in this scenario. With the proposed method, the tracking robot successfully figures out the hidden goal position and stays at that position to smartly keep close to the target robot, leading to effective interactions for both agents. However, the MPC baseline does not understand the interactions and simply keeps chasing the target robot.

Quantitatively, a Monte Carlo study of the tracking game compares three approaches. Two game-theoretic approaches outperform the MPC baseline that does not reason about the opponent's decision-making with more accurate trajectory prediction, safer (lower collisions), and more efficient (lower interaction costs) interactions. Within the two game-theoretic methods, the proposed adaptive planner yields more accurate parameter estimations, trajectory predictions, and lower interaction cost variance than the KKT-constrained baseline.

The proposed method's scalability to more complex scenarios is tested in a ramp-merging game in Section 5.5. Monte Carlo studies comparing four approaches with varying traffic densities are conducted. Again, game-theoretic approaches with goal inference generally outperform the MPC baseline. The MPC only shows competitive prediction accuracy and ego costs in settings with fewer agents but generally results in much more collisions, rendering the importance of strategic reasoning in these traffic scenarios. Within the game-theoretic approaches, an ablation study compares methods with versus without objective inference and demonstrates the necessity of understanding the opponents' intentions for increased prediction accuracy, efficiency and safety.

The two game-theoretic approaches with objective inference are comparable. Regarding interaction *efficiency*, the proposed adaptive planner and the KKT-constrained baseline both

provide an ego cost performance that is statistically significantly better than all other baselines but not statistically significantly worse than solving games with ground truth opponent intents in denser settings with 5 and 7 agents. This result is as expected, as the two approaches solve the same MLE problem when game inequality constraints are inactive, e.g., when players are not close enough to one another. However, statistical significance is indeed observed in interaction *safety*. In the densest setting, the KKT-constrained baseline collides statistically significantly more often—7 times more than the proposed approach that handles collision avoidance constraints in inverse games. As a piece of evidence, eliminating collision avoidance inequality constraints from inverse game solutions of the proposed framework also leads to 5 times more collisions than the original full framework.

Another Monte Carlo study compares the two inference-based game solvers' inference and prediction performance using partial versus full observations. The results indicate that the proposed method's trajectory prediction capability is more robust against partial-state observation in dense scenarios. The KKT-constrained baseline shows a statistically significant performance drop using partial-state than full-state observations in the 7-player case.

In run time performance, the proposed adaptive planner is marginally slower than the KKT-constrained baseline, despite solving a more complex problem with inequality constraints. Although the current implementation of the proposed planner demonstrates a real-time planning rate for only up to three players, the differentiability of the proposed method further supports integration with NNs. A combination of the adaptive planner with an NN shows computational acceleration by 60% with a marginal performance loss in the 7-player case. Other run-time optimization schemes are also possible and to be explored, as will be discussed in Section 7.2.

Finally, the third set of experiments on hardware show that the proposed method is capable of running on a real hardware platform featuring two agents and accounting for real-world noise, e.g., interacting with a human agent whose behavior is noisier than a computer-controlled robot.

To summarize, the proposed adaptive MPGP planner gives promising results in the simulation and hardware experiments. By estimating opponents' objective models online and conducting game-theoretic planning adaptively, it performs closely to solving games with ground truth opponent objectives. The adaptive planner outperforms the MPC baseline and the game-theoretic baseline without goal inference. Compared with the state-of-the-art inverse game solver, the proposed approach matches its efficiency performance, outperforms it in safety, and further provides a differentiability feature. Chapter 6 explores using this feature to go beyond point estimations of objectives in this chapter to characterize distributional *beliefs* of players' objectives. The performance advantage of the proposed approach over the baselines is more pronounced in denser interaction scenarios.

Despite these encouraging results, a few assumptions made in this work still require further investigation to solidify the conclusions from here. These assumptions and future work are discussed in Section 7.2.

6

Beyond Point Estimation: Towards Distributional Uncertainty

The preceding chapters have presented the main contribution of this thesis—an adaptive game-theoretic planner. The planner is enabled by differentiating through a trajectory game solver and utilizing the gradient for MLE of opponents' objectives. This chapter focuses on going beyond MLE and inferring a belief *distribution* instead of a *point estimation* of players' objectives. The approach is realized by combining a variational autoencoder (VAE) [17, 18] with the proposed differentiable solver, which learns from data to encode observations to a latent distribution and recover players' objective distribution from the latent space.

Note that this chapter is meant to take a first step toward future work and conduct an initial test to verify the viability of the proposed VAE framework. Therefore, only initial results are provided as proof of concept. A more thorough evaluation, such as on large-scale real-world driving data [78], is left for future work and is beyond the scope of this thesis.

6.1. Motivation

Beyond point estimation from MLE, the main benefit of quantifying the intention uncertainty is added information for decision-making, which may enable safer plans. For instance, when observations are less informative, the uncertainty of inferred opponents' intention should be higher, and the ego robot should be more careful in making their decisions.

Sometimes, opponents may commit to distinct sequences of actions that will lead to different interactions, and their true intent is uncertain to the robot. In such cases, it is necessary for the robot to maintain a multi-modal belief over opponents' intent. Simply using MLE and ignoring other likely modes can cause danger. Take robot-pedestrian collision avoidance in Figure 6.1 as an example. A robot encounters a pedestrian whose intent might be crossing to the left or right and is ambiguous from current observation. An MLE solution might be overly confident that the pedestrian wants to go to either side and plan unsafe trajectories. By contrast, maintaining a multi-modal belief over the opponent's intent and carefully planning trajectories conditioned on all possible intents presents a safer solution [79]. Many existing interaction-aware motion planning frameworks, such as contingency planning and partially observable Markov decision process (POMDP) planning, require a belief over opponents' intent to plan against the intention uncertainty. The approach proposed by this chapter exactly seeks to provide this information for these types of frameworks [62, 79–83].

In robotics literature, Bayesian inference has been widely used to infer unknown parameter distribution for decision-making, such as in POMDP planning [81–83] and robust control [84–

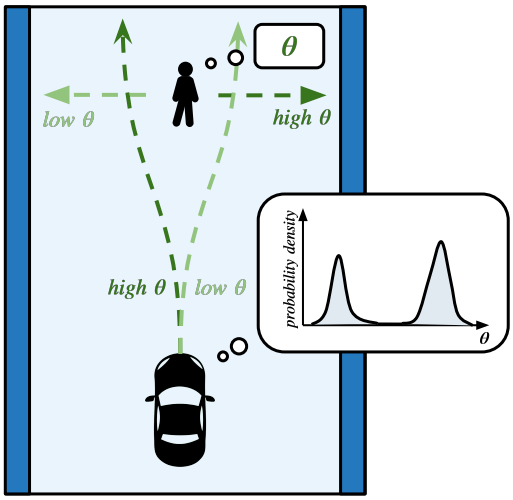


Figure 6.1: A robot encounters a pedestrian whose intent might be crossing to the left or right and is ambiguous from current observation. The robot maintains a multi-modal belief of the pedestrian's intent.

86]. However, the intractability of exact Bayesian inference has limited these methods to simple, discretized, low-dimensional distributions. Similarly, in machine learning literature, Bayesian inverse reinforcement learning (BIRL) presents an effective solution to reward distribution learning [87, 88] but is limited by its scalability. To this end, Chan and Schaar [89] propose to use variational inference (VI) [90, 91] to pose an inference problem as an optimization problem and approximate the posterior distribution of a reward function, which improves the scalability of BIRL methods. This chapter employs a similar idea in noncooperative games and approximates a posterior objective parameter distribution given observations $p(\boldsymbol{\theta} \mid \mathbf{Y})$ via VI. Unknown parameter $\boldsymbol{\theta}$ values then determine possible intents. To solve the VI problem, this work employs a VAE [17, 18] combined with a game solver via differentiable programming for added structure and inductive bias for training. By incorporating an offline training phase, prior knowledge from existing data can be employed.

6.2. Problem Statement

Ideally, we wish to solve a Bayesian inference problem to get players' objective distribution, given observation data:

$$p(\boldsymbol{\theta} \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(\mathbf{Y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \tag{6.1}$$

However, evaluating the posterior distribution $p(\boldsymbol{\theta} \mid \mathbf{Y})$ is mostly intractable due to intractability of the integration $\int p(\mathbf{Y} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$. Sampling-based approaches have been demonstrated to provide good performance in both games and single-agent optimal control settings [92, 93]. Hence, we wish to at least sample from an approximation to the posterior $p(\boldsymbol{\theta} \mid \mathbf{Y})$ to characterize it.

In order to approximate the posterior objective distribution $p(\boldsymbol{\theta} \mid \mathbf{Y})$, we assume it can be captured by a latent variable model. The sampling process of objectives $\boldsymbol{\theta}$ is assumed to consist of two steps: (i) A latent variable $\boldsymbol{\alpha}$ is sampled from a prior distribution $p_{\psi^*}(\boldsymbol{\alpha})$. (ii) A game objective $\boldsymbol{\theta}$ is then generated from some conditional distribution $p_{\psi^*}(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$. Both $p_{\psi^*}(\boldsymbol{\alpha})$ and $p_{\psi^*}(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$ come from parameterized families of distributions $p_{\psi}(\boldsymbol{\alpha})$ and $p_{\psi}(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$ with their PDFs being differentiable w.r.t. both ψ and $\boldsymbol{\alpha}$. Dynamic game play then maps objectives $\boldsymbol{\theta}$ to observations $\mathbf{Y}$.

Assume we have the latent variable model $p_{\psi^*}(\boldsymbol{\alpha})$, $p_{\psi^*}(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$ and the posterior distribution $p_{\psi^*}(\boldsymbol{\alpha} \mid \mathbf{Y})$ available. Given an observation $\mathbf{Y}$, we can effectively sample game objectives $\boldsymbol{\theta}$ by sampling latent variables $\boldsymbol{\alpha}$ from the posterior $p_{\psi^*}(\boldsymbol{\alpha} \mid \mathbf{Y})$ and sampling $\boldsymbol{\theta}$ from distributions conditioned on the latent samples $p_{\psi^*}(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$. This procedure is equivalent to approximating the integration $p_{\psi^*}(\boldsymbol{\theta} \mid \mathbf{Y}) = \int p_{\psi^*}(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) p_{\psi^*}(\boldsymbol{\alpha} \mid \mathbf{Y}) d\boldsymbol{\alpha}$ by sampling $\boldsymbol{\alpha}$ in the latent space.

Unfortunately, given an observation $\mathbf{Y}$, most of the process above is hidden: the true parameter ψ, the values of the latent variable $\boldsymbol{\alpha}$, and even the form of the distributions $p_\psi(\boldsymbol{\alpha})$, $p_\psi(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$, and $p_\psi(\boldsymbol{\alpha} \mid \mathbf{Y})$. Hence, by assuming the form of the distributions, we wish to approximate the posterior distribution $p_{\psi^*}(\boldsymbol{\alpha} \mid \mathbf{Y})$ with a model $q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y})$ by minimizing the Kullback–Leibler (KL) divergence between them:

$$\phi^* \in \arg\min_{\phi} D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha} \mid \mathbf{Y})) \text{ (posterior approximation)}, \tag{6.2}$$

while estimating the latent variable model parameter ψ^* via maximum likelihood estimation (MLE):

$$\psi^* \in \arg\max_{\psi} \log p(\mathbf{Y} \mid \psi) = \arg\max_{\psi} \log[\int p(\mathbf{Y} \mid \boldsymbol{\alpha}, \psi) p(\boldsymbol{\alpha} \mid \psi) d\boldsymbol{\alpha}] \text{ (latent model estimation)}. \tag{6.3}$$

6.3. Approach

We approximate Bayesian inference in Equation (6.1) with an optimization problem with a VI objective introduced in [94, Section 10.1.1]. This section first introduces the optimization objective and then presents the proposed framework.

Posterior Approximation First, assuming ψ in the problem in Equation (6.2) is fixed, we seek an approximating posterior distribution $q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y})$ by minimizing the KL divergence:

$$\begin{aligned}
\phi^* &\in \arg\min_{\phi} D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha} \mid \mathbf{Y})) && (6.4)\\
&= \arg\min_{\phi} \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) - \log(\frac{p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha}) p_\psi(\boldsymbol{\alpha})}{p_\psi(\mathbf{Y})})] && (6.5)\\
&= \arg\min_{\phi} \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) - \log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha}) - \log p_\psi(\boldsymbol{\alpha})] + \log p_\psi(\mathbf{Y}). && (6.6)
\end{aligned}$$

Hence, we have:

$$D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha} \mid \mathbf{Y})) \overbrace{-\mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) - \log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha}) - \log p_\psi(\boldsymbol{\alpha})]}^{\mathcal{L}(\psi, \phi \mid \mathbf{Y})} = \log p_\psi(\mathbf{Y}), \tag{6.7}$$

where the term $\mathcal{L}(\psi, \phi \mid \mathbf{Y})$ is called evidence lower bound (ELBO). Because the KL divergence $D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha} \mid \mathbf{Y})) \geq 0$, $\mathcal{L}(\psi, \phi \mid \mathbf{Y})$ provides a lower bound for the "evidence" $\log p_\psi(\mathbf{Y})$. Since the term $\log p_\psi(\mathbf{Y})$ is a constant w.r.t. the decision variable ϕ, maximizing $\mathcal{L}(\psi, \phi \mid \mathbf{Y})$ also minimizes the KL divergence $D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha} \mid \mathbf{Y}))$. Therefore,

the problem becomes:

$$\phi^* \in \arg\max_{\phi} \mathcal{L}(\psi, \phi \mid \mathbf{Y}) \tag{6.8}$$

$$= \arg\max_{\phi} -\mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) - \log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha}) - \log p_\psi(\boldsymbol{\alpha})] \tag{6.9}$$

$$= \arg\max_{\phi} \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})] - \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log \frac{q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y})}{p_\psi(\boldsymbol{\alpha})}] \tag{6.10}$$

$$= \arg\max_{\phi} \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})] - D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha})). \tag{6.11}$$

Latent Model Estimation Now we move on to estimate the parameter ψ for the latent variable model. For the problem in Equation (6.3), due to the intractability of the integration $\int p(\mathbf{Y} \mid \boldsymbol{\alpha}, \psi) p(\boldsymbol{\alpha} \mid \psi) d\boldsymbol{\alpha}$, the MLE is intractable to compute exactly. Fortunately, the ELBO is a lower bound on it as well $\mathcal{L}(\psi, \phi \mid \mathbf{Y}) \leq \log p(\mathbf{Y} \mid \psi) = \log p_\psi(\mathbf{Y})$. Hence, we can estimate ψ by also maximizing the ELBO:

$$\psi^* \in \arg\max_{\psi} \mathcal{L}(\psi, \phi \mid \mathbf{Y}) = \arg\max_{\psi} \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})] - D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha})). \tag{6.12}$$

Complete Objective Then, we can solve the two problems in Equation (6.2) and Equation (6.3) by jointly maximizing the ELBO w.r.t. both ψ and ϕ:

$$\psi^*, \phi^* \in \arg\max_{\psi, \phi} \mathcal{L}(\psi, \phi \mid \mathbf{Y}) \tag{6.13}$$

$$= \arg\max_{\psi, \phi} \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})] - D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha})). \tag{6.14}$$

Framework To avoid optimizing the local variational parameters ϕ for each observation $\mathbf{Y}$, we employ an amortized VI [94, Section 10.1.5] and use NNs as function approximators, which allows ϕ to be shared across all data points. More specifically, we use an NN as a recognition model (a probabilistic *encoder*) $q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y})$, which encodes observation data $\mathbf{Y}$ into a latent distribution over possible values of $\boldsymbol{\alpha}$. We utilize another NN as a probabilistic *decoder* $p_\psi(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$, which recovers a game objective distribution from a latent variable $\boldsymbol{\alpha}$. Thereafter, the proposed differentiable game solver is added as another unparameterized layer and reconstructs observations $\hat{\mathbf{Y}}$ given sampled objectives $\boldsymbol{\theta}$. In this way, the decoder network and the differentiable game solver act as the model $p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})$ together. The pipeline is depicted in Figure 6.2.

Training The NN models are trained on a data set $\mathcal{D} = \{\mathbf{Y}^{(j)}\}_{j=1}^{M}$ consisting of M i.i.d. and potentially partial-state game trajectory observations. We employ a mini-batch stochastic gradient descent (SGD) to optimize the objective in Equation (6.14) and fit the models. We employ an unbiased estimator to compute the objective based on mini-batches:

$$\tilde{\mathcal{L}}^B(\psi, \phi \mid \mathbf{Y}) = \frac{M}{B} \sum_{j=1}^{B} \frac{1}{L} \sum_{l=1}^{L} \log p_\psi(\mathbf{Y}^{(j)} \mid \boldsymbol{\alpha}^{(j,l)}) - D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha}^{(j)} \mid \mathbf{Y}^{(j)}) \| p_\psi(\boldsymbol{\alpha}^{(j)})), \tag{6.15}$$

where $\tilde{\mathcal{L}}^B(\psi, \phi \mid \mathbf{Y})$ denotes ELBO estimated from a mini-batch $\mathcal{D}^B$ consisting of B samples from the full data set $\mathcal{D}$ with M data points. L denotes the number of samples used to approximate $\mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})]$ per step. In each training epoch, the training set $\mathcal{D}$ is divided into mini-batches and handed in in order for training. Samples are shuffled in between epochs.

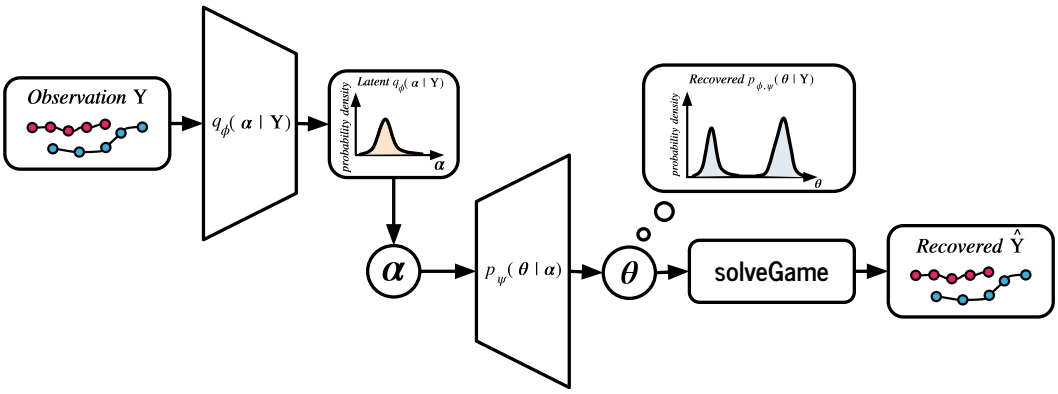


Figure 6.2: The proposed variational autoencoder pipeline. Given an observation $\mathbf{Y}$, objective belief distribution $p(\boldsymbol{\theta} \mid \mathbf{Y})$ can be recovered by sampling from the posterior latent distribution $q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y})$ and mapping the latent samples $\boldsymbol{\alpha}$ through the decoder network $p_\psi(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$. Even if the latent distribution is limited to uni-modal, the recovered belief can be multi-modal.

Computation of the gradient $\nabla_\phi \mathbb{E}_{q_\phi(\boldsymbol{\alpha}|\mathbf{Y})}[\log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})]$ would involve differentiating through a sampling operation, which is not differentiable. So we employ a reparameterization trick [94, Section 10.2.1] to work around this issue. For instance, for a diagonal Gaussian distribution, instead of sampling $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{Y}), \text{diag}(\boldsymbol{\sigma}_\phi(\mathbf{Y})))$ directly, we sample a noise vector $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ from a standard Gaussian. It follows that $\boldsymbol{\alpha} = \boldsymbol{\mu}_\phi(\mathbf{Y}) + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_\phi(\mathbf{Y})$, where $\odot$ denotes elementwise multiplication. In this case, taking gradient steps w.r.t. ϕ does not require a re-sampling of latent variables.

For simplification, we assume the prior distribution $p_\psi(\boldsymbol{\alpha})$ is a standard Gaussian $\mathcal{N}(0, I)$. Then, we can compute the KL divergence term analytically [94, Section 10.2.1.1]:

$$D_{\mathcal{KL}}(q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y}) \| p_\psi(\boldsymbol{\alpha})) = \frac{1}{2} \sum_{j=1}^{J} ((\mu_{\phi,j}(\mathbf{Y}))^2 + (\sigma_{\phi,j}(\mathbf{Y}))^2 - 1 - \log((\sigma_{\phi,j}(\mathbf{Y}))^2)), \tag{6.16}$$

where $\mu_{\phi,j}(\mathbf{Y})$ and $\sigma_{\phi,j}(\mathbf{Y})$ denote the jth elements of the mean and variance vectors output by the recognition model $q_\phi(\boldsymbol{\alpha} \mid \mathbf{Y})$, both with a total element number of J.

Furthermore, if we assume the marginal distribution $p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})$ is an isotropic Gaussian $\mathcal{N}(\boldsymbol{\mu}_\psi(\boldsymbol{\alpha}), rI)$, with r denoting a positive real number, the term $\log p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})$ can be further reduced to a negative squared distance $-\|\mathbf{Y} - \boldsymbol{\mu}_\psi(\boldsymbol{\alpha})\|^2$ as an approximation [95, Section 20.3.5]. It is important to note that this assumption only constrains that the distribution $p_\psi(\mathbf{Y} \mid \boldsymbol{\alpha})$ for a *fixed* $\boldsymbol{\alpha}$ is unimodal. The recovered objective distribution $p(\boldsymbol{\theta} \mid \mathbf{Y})$ can still be multi-modal, as will be shown in Section 6.4.2.

6.4. A Highway Driving Example

This section presents initial results on inference in a highway driving scenario for the proposed VAE-based framework. We use a simplified one-dimensional version of the setting in Section 5.5, which is shown in Figure 6.3. A driver is fixed laterally at a lane center and only controls the longitudinal speed of the vehicle. Given observed driving velocity sequences of a player, the proposed pipeline is trained to predict the posterior distribution of the player's desired driving velocity. Note that this toy example uses a special case of games—a single-player optimal control problem as initial tests. But the approach is general and can be readily applied to multi-agent settings. However, thorough tests in those settings are deferred to future work.

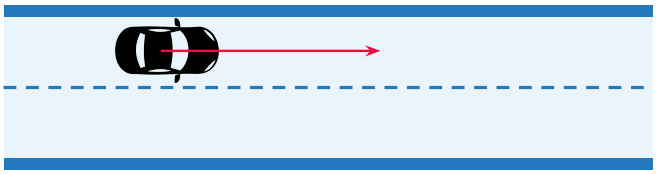


Figure 6.3: The proposed variational autoencoder pipeline is trained to predict the posterior distribution of a driver's desired driving speed in a one-dimensional highway driving example.

6.4.1. Experiment Setup

For simplification of initial tests, vehicles are modeled as double integrators with states $x_t^i = (p_{x,t}^i, v_{x,t}^i)$ consisting of longitudinal position and speed, and controls $u_t^i = a_{x,t}^i$ being longitudinal acceleration at time step t. In this single-player example, player index $i = 1$. A player seeks to optimize their objective function:

$$J^i = \sum_{t=1}^{T-1} \|v_{x,t+1}^i - v_{x,\text{goal}}^i\|_2^2 + 0.1\|u_t^i\|_2^2, \tag{6.17}$$

which penalizes them from deviating from the desired driving speed $v_{x,\text{goal}}^i$ and applying control efforts. Here, the objective parameter θ is $v_{x,\text{goal}}^1$. In the presentation hereafter, we ignore the player's index in the single-player example for conciseness.

For this inference task, this section uses simple fully-connected feed-forward neural networks with tanh activation and one hidden layer consisting of 128 neurons as the encoder and decoder models. The latent variable α's dimension is 1, the same as the dimension of the objective parameter $v_{x,\text{goal}}$. The encoder takes an observed driving speed sequence as input and predicts the mean and variance for $q_\phi(\alpha \mid Y)$. We assume $p_\psi(Y \mid \alpha)$ is an isotropic Gaussian $\mathcal{N}(\mu, I)$, and we take the output of the decoder network directly as θ. The models are trained on a data set consisting of 20000 driving plans with a horizon of 10 obtained by solving the optimal control problem characterized by Equation (6.17) for one player. An observation Y comprises the driving velocities $v_{x,t}$ of the player at each time step. Hence, given an observation Y, the pipeline is trained to predict objective samples θ from the approximated posterior distribution $p(\theta \mid Y)$. The ground truth objective velocities are randomly sampled from a Gaussian mixture distribution depicted in Figure 6.4, which has two clusters with their means at 20% and 70% of the maximal velocity $v_{\text{max}} = 20\,\text{m}\,\text{s}^{-1}$ and unit variance. Initial velocities are set to $0.9\,v_{x,\text{goal}}$ to ensure the open-loop plans always reach the desired velocities within the planning horizon, i.e., the player's intent is observable. We employ an Adam optimizer [96] with a learning rate of 0.002 and a batch size of 128 for a training of 2400 epochs.

6.4.2. Results

First, the proposed framework's capacity as a generative model is tested. As shown in Figure 6.4, an objective distribution that qualitatively resembles the ground-truth training objective distribution $p_{\psi^*}(\theta)$ can be recovered by extensive sampling from the *unconditioned* latent distribution $p_\psi(\alpha)$ and mapping the samples through the decoder network $p_\psi(\theta \mid \alpha)$. That is, the NN captures the underlying training distribution which the player's desired velocities are sampled from. Quantitatively, KL divergence between the two distributions is approximately 0.18, estimated on two data sets with a sample size of 20000.

Furthermore, the trained pipeline's capacity to predict a belief objective distribution conditioned on observation is tested on several data sets. Results on observations from the training set and an unseen test set are given in Figure 6.5 and Figure 6.6. The distributions are char-

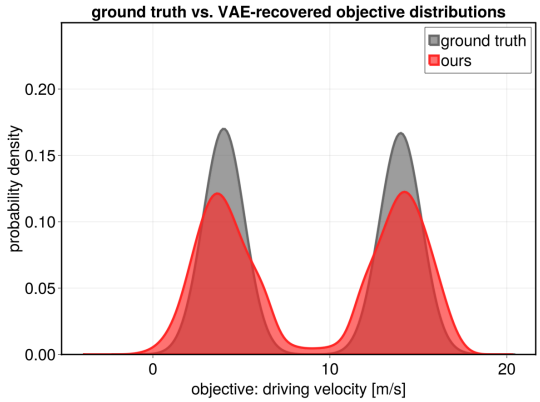


Figure 6.4: The ground truth $p_{\psi^*}(\theta)$ and a recovered unconditioned objective distribution by the proposed variational autoencoder pipeline. The objective distribution is recovered by sampling extensively from the unconditioned latent distribution $p_\psi(\alpha)$ and mapping the samples through the decoder network $p_\psi(\theta \mid \alpha)$.

acterized by mapping latent variables α sampled from a *conditioned* latent space $q_\phi(\alpha \mid Y)$ through the decoder network $p_\psi(\theta \mid \alpha)$.

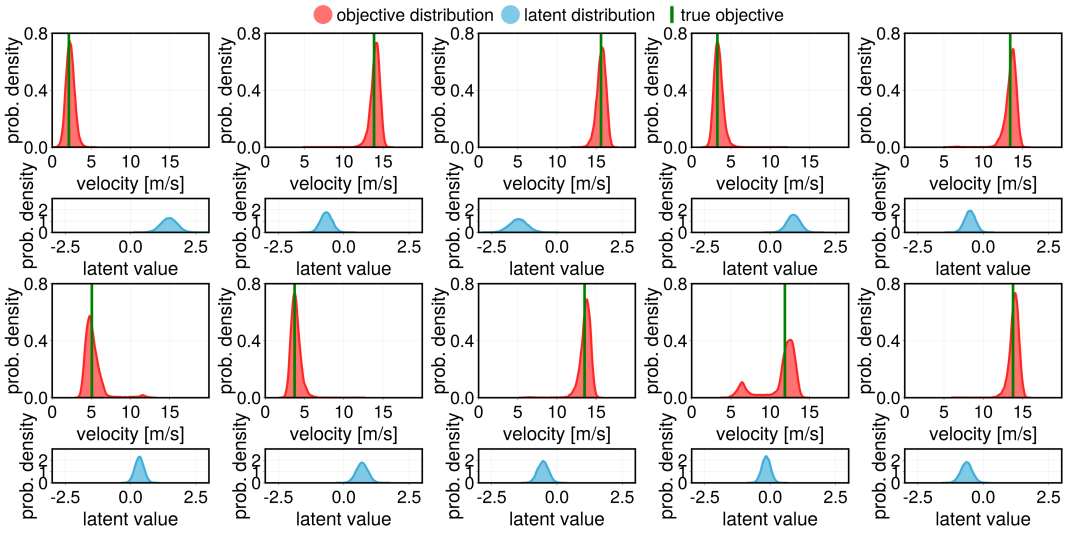


Figure 6.5: Inferred objective beliefs on observations from the training set. The beliefs are characterized by mapping latent variables α sampled from a *conditioned* latent space $q_\phi(\alpha \mid Y)$ through the decoder network $p_\psi(\theta \mid \alpha)$. When the uncertainty about the observation is high, the pipeline generates a multi-modal belief distribution of game objectives to capture the most likely modes.

Results on the two data sets are highly similar—the NN is able to give accurate objective predictions and generalize to unseen test data from the same distribution as the training set. As shown in Figure 6.5 and Figure 6.6, when the true objective velocity is around either of the two peaks of the training distribution, i.e., 4 or $14\,\mathrm{m\,s^{-1}}$ of $p_{\psi^*}(\theta)$ in Figure 6.4, the network gives a narrow posterior distribution with little uncertainty, which matches the ground truth well. Moreover, when the objective velocity is between the two peaks, the network is able to give a multi-modal prediction to capture both likely modes. The latent distribution is shifted and shaped differently w.r.t. the observation.

As pointed out before and evidenced by the figures, even if the conditional distributions $q_\phi(\alpha \mid Y)$ and $p_\psi(Y \mid \alpha)$ are assumed to be Gaussian, the predicted objective distribution $p(\theta \mid Y)$ can still be multi-modal due to the nonlinear mapping between the latent and objective spaces.

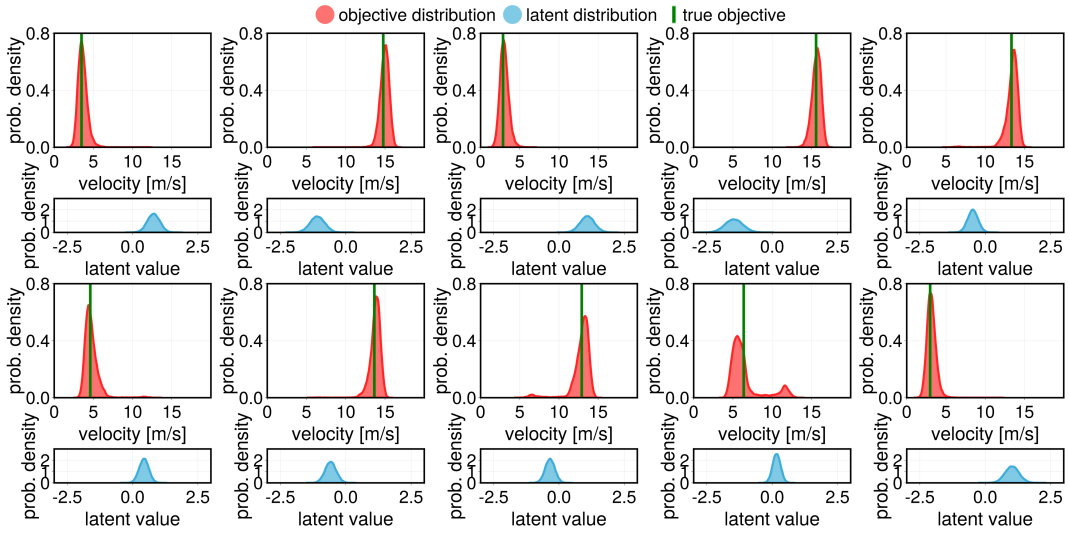


Figure 6.6: Inferred objective beliefs on observations from an unseen test set. The trained variational autoencoder pipeline successfully generalizes to the unseen test set and provides similar behaviors as on the training set.

Additionally, the trained pipeline is tested on a data set consisting of artificially generated constant-velocity observations, as depicted in Figure 6.7. The inference problem might seem trivial to a human, as we can simply conclude that a vehicle wants to drive at the current speed if they never changes their velocity. However, results from this simple setting demonstrate a clear spectrum of predicted objective distributions when the "artificial objective" is shifted throughout the training distribution.

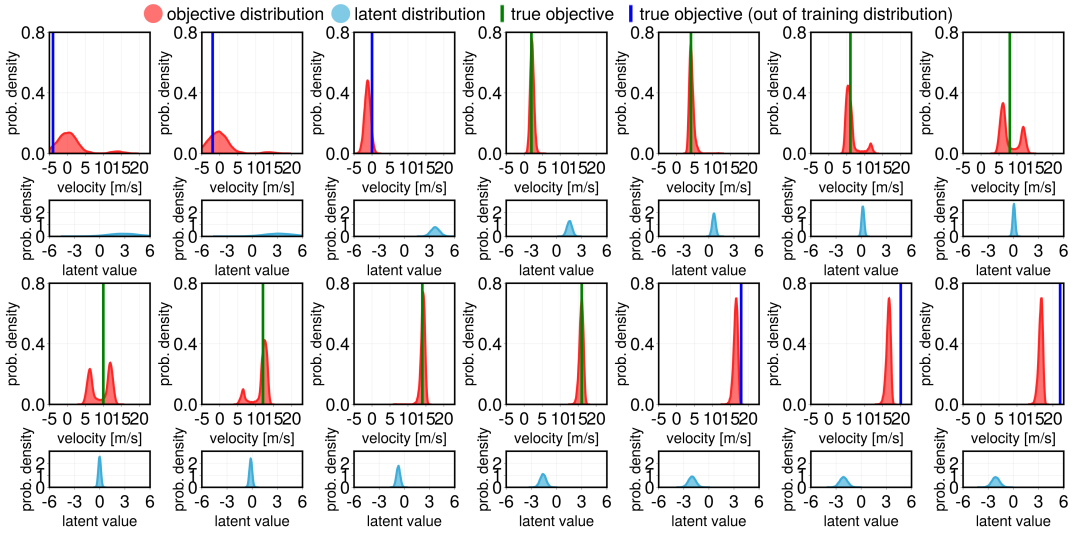


Figure 6.7: Inferred objective beliefs on some artificially generated constant-velocity observations. The pipeline shows a spectrum of predicted beliefs as the observed velocities shift within the training distribution. When observing out-of-distribution data, the pipeline can give wrong but certain predictions.

The minimal and maximal velocities seen by the network during training are 0.199 and $17.866\,\mathrm{m\,s^{-1}}$. In tests depicted in Figure 6.7, an artificial objective velocity is increasingly shifted by an interval of $2\,\mathrm{m\,s^{-1}}$ with the first and last three velocities out of the training distribution. With these out-of-distribution observations, the network gives relatively inaccurate predictions. Below the minimum of seen velocities, the network predicts a wide distribution with high uncertainty. However, it tends to give a certain but wrong prediction while seeing

velocities above the maximum. This clearly demonstrates the trustworthy issue of data-driven methods. Within the training distribution, the network shows a clear spectrum of predicted distributions while the artificial objective moves between the two modes.

6.5. Discussion and Conclusion

This chapter has proposed a variational autoencoder pipeline built upon the proposed differentiable game solver in Chapter 4 to infer potentially multi-modal beliefs about players' intents. Going beyond point estimation of players' objectives using MLE in Chapter 4, this chapter uses variational inference to approximately solve Bayesian inference of players' objectives given observations. To avoid per sample optimization, the VAE framework is employed for amortization. Initial tests on a toy example have validated the feasibility of the proposed pipeline. In a driving example, the trained VAE framework can generate a game objective distribution that resembles the underlying training data distribution, demonstrating its capability as a generative model. Furthermore, the trained pipeline shows to predict an accurate, narrow, uni-modal posterior objective distribution if the observation is unambiguous based on seen data. When the uncertainty about the observation is high, the pipeline is able to infer a multi-modal belief of players' objectives to capture the most likely modes. However, as a drawback, the performance of the pipeline on an artificially generated data set demonstrates that the VAE can give a wrong but certain prediction while observing unseen out-of-distribution data. This trustworthy issue is commonly observed in learning-based approaches [97] and is especially concerning for safety-critical applications like autonomous driving and requires further investigation.

Various promising future research directions have been indicated after conducting initial tests of the proposed VAE framework. These future directions, as well as the issue mentioned above, will be discussed in Section 7.2.

7

Summary and Future Work

In pursuit of reliable autonomy for mobile robots like self-driving cars in dynamic environments, developing interaction-aware motion planning algorithms that enable safe and intelligent interactions between agents remains challenging. Dynamic game theory renders a powerful mathematical framework to model these interactions rigorously as coupled optimization problems. By solving the resultant coupled optimization problems to equilibrium solutions, the game-theoretic models explicitly account for the interdependence of agents' decisions and achieve simultaneous prediction and planning while handling general coupled constraints between agents. However, most existing game-theoretic motion planning approaches [12–15] rely on perfectly known objective models of all agents. This assumption presents a key obstacle to real-world ego-centric planning applications of these methods, where only local information is available. This thesis investigated solution approaches to relax this assumption and explicitly account for the ego agent's uncertainty about other agents' objectives while adaptively conducting game-theoretic motion planning.

7.1. Summary

This section summarizes the contributions and the results. Thereafter, Section 7.2 reflects on the results and provides an outlook on future work.

7.1.1. Adaptive Game Solver

The first contribution (Chapter 4) of this thesis is an online adaptive model-predictive gameplay (MPGP) framework that jointly infers other players' objectives and computes corresponding generalized Nash equilibrium (GNE) strategies. These strategies are then used as predictions for other players and control strategies for the ego agent. The adaptivity of the proposed approach is enabled by differentiating through a trajectory game solver whose gradient signal is used for maximum likelihood estimation (MLE) of opponents' objectives. By replacing the equilibrium constraints of a constrained MLE problem with the proposed differentiable solver, the objective inference problem is cast as an unconstrained optimization problem and effectively solved via online gradient descent. By further computing GNE strategies based on objective estimations, the proposed planner actively adapts to estimated opponents' intentions. Compared with existing inverse game solutions, the proposed approach handles general inequality constraints in forward games, such as collision avoidance constraints. Furthermore, the differentiability feature of the solver enables a combination of this optimization-based framework with learning-based modules, as shown in Section 5.5.2 and Chapter 6.

Chapter 5 evaluates the proposed approach in two simulated scenarios and two hardware experiments and shows promising results. In a two-player tracking game in Section 5.4, an ego robot is tasked to predict the other agent's goal position to stay close to them while avoiding collisions. A Monte Carlo study indicates that the proposed method outperforms the model-predictive control (MPC) baseline that does not reason about the opponent's decision-making with more accurate trajectory prediction, safer (lower collisions), and more efficient (lower interaction costs) interactions. The performances of the two game-theoretic approaches that infer the opponent's objectives are closer to each other. Compared with the Karush–Kuhn–Tucker (KKT)-constrained baseline [46], the proposed adaptive planner shows more accurate inference and prediction performance and lower interaction cost variance.

The scalability of the proposed approach is tested in a more complex ramp-merging scenario in Section 5.5. An ego robot is tasked to merge into a busy traffic flow. Compared with the proposed approach, the MPC baseline only shows competitive prediction accuracy and ego costs in settings with fewer agents but generally results in much more collisions, rendering the importance of strategic reasoning in dense traffic scenarios. Among the game-theoretic approaches, an ablation study comparing planning with versus without online objective inference further demonstrates the necessity of reasoning about opponents' objectives in such settings for increased safety and efficiency. The two game-theoretic approaches with objective inference are comparable in this experiment. A Manning-Whitney U-test reveals that, in denser scenarios with 5 and 7 players, both methods achieve an ego cost (*efficiency*) performance that is significantly better than all other baselines but not significantly worse than solving games with ground truth opponent intents. This result is as expected, as the two approaches solve the same MLE problem when inequality constraints in games are inactive, e.g., agents are not close to one another. Nevertheless, the handling of inequality constraints by the proposed approach provides better *safety* in highly interactive settings. In the densest setting with 7 agents, the KKT-constrained baseline results in statistically significantly higher collision times—7 times more than the proposed approach. Moreover, a Manning-Whitney U-test in Section 5.5.2 suggests that in dense scenarios, the proposed approach's trajectory prediction capability is more robust against partial-state observations than the KKT-constrained approach.

Regarding the run time results, the two approaches with objective inference result in a higher run time than other methods due to added complexity. The proposed adaptive planner achieves a marginally slower computation time than the KKT-constrained baseline, despite solving a more complex problem with inequality constraints in games. While the proposed pipeline demonstrates a real-time computation for settings with up to 3 agents, Section 5.5.2 explores a combination of the differentiable solver with an NN, which is trained to propose an initial guess online and reduces the run time by 60% while only resulting in a marginal performance loss. This exploration shows a promising future direction to accelerate the online computation of the proposed framework, as will be discussed in Section 7.2.

Finally, two real-world hardware experiments in Section 5.6 demonstrate the tracking game between a robot and (i) another robot and (ii) a human. Both experiments show that the adaptive planner reliably recovers the opponent's goal positions and enables effective tracking behaviors while actively avoiding collisions, despite real-world noise.

All the experiment results together indicate that the proposed MPGP planner performs closely to solving games with ground truth objectives and surpasses the baselines. It outperforms the game-theoretic baseline without goal inference and the MPC baseline. Compared with the KKT-constrained baseline, it matches its efficiency performance, outperforms it in safety, and further supports a differentiability feature that leads to the next contribution. The performance advantage of the proposed approach over the baselines is more pronounced in denser interaction scenarios.

7.1.2. Variational Inference of Objective Distribution

In addition to the main contribution in Section 7.1.1, the second (Chapter 6) contribution of this thesis is a variational autoencoder (VAE) pipeline built upon the proposed differentiable game solver. This contribution aims at going beyond the point estimation in the first contribution and inferring potentially multi-modal *beliefs* about players' objectives based on observed game interactions. The belief information can be employed by various planning approaches, e.g., partially observable Markov decision process (POMDP) [81–83], robust control-theoretic [84–86], and contingency [79] planning methods. The main idea is to employ variational inference (VI) to approximate Bayesian inference of players' objectives. The VAE framework is utilized for amortization to avoid per-sample optimization. The prototype is tested on a toy example in Section 6.4 to infer a vehicle's desired driving speed from observed driving data. Results show that after training, the proposed pipeline can: (i) generate a game objective distribution that resembles the underlying training data distribution and (ii) accurately predict a narrow, uni-modal posterior objective distribution when the observed driving velocity sequence is unambiguous based on seen data in the past and (iii) generate a multi-modal belief distribution of player's objective to capture mostly likely modes in case of high uncertainty.

The encouraging initial results validate the viability of the VI idea and motivate appealing next steps for future work, as will be pointed out in Section 7.2.

7.2. Future Work

The results of this thesis have indicated many promising future research directions. This section gives an outlook on these future works.

First, to solidify the conclusions of this thesis, a few assumptions made by this work need to be further investigated. The first assumption is that the players' objectives, e.g., travel speed and lane preferences, are assumed to be fixed in this work. However, in real-world driving scenarios, human drivers' intents might not necessarily be fixed over time. It requires further investigation of whether human agents' objectives change and in which interaction scenarios. If the answer is yes, it is necessary to study how will the changes influence the interactions. Although the current online inference framework can be readily applied to settings with varying objectives, how the adaptation will influence the method's performance still requires more validation. Second, throughout this work, agents' behaviors are assumed to be approximately characterized by local generalized Nash equilibria. In quantitative experiments, the modeling of opponent players is rather ideal—they are modeled as solving a ground truth game together. Section 5.6 demonstrates a robot using the proposed game-theoretic planner to interact reasonably with a human agent in a 2-player tracking game. However, it remains to be quantified how closely these game-theoretic models capture real-world agent behaviors. Moreover, it is necessary to thoroughly test the proposed planner interacting with a higher number of less ideal agents. As a first step, this could be realized by interacting with human drivers or learned models in simulations. Third, safety constraints used in the evaluation of this work are assumed to be simple. However, tremendous work has explored advanced safety constraints, e.g., signal temporal logic (STL) [98] and reachability-based constraints [99]. Incorporating these advanced safety constraints into more realistic multi-agent interaction settings, e.g., driving scenarios involving more complex traffic rules, still requires much research.

Second, Chapter 6 takes an initial step and validates the proposed VAE pipeline in a special case of games—a single-player optimal control problem. The pipeline itself is general for multi-agent problems and can be readily applied to interactive settings. Hence, a natural next step is to employ the proposed framework in multi-player game scenarios. It is to be investigated how well the proposed VI approach with added game-theoretic structure can ap-

proximate players' objectives in settings where players' decision-making is coupled. It is also interesting to research the effect of the added differentiable game solver on model training's data efficiency and final performance compared with model-free approaches. Furthermore, to apply the proposed method in more realistic settings, the training data should involve richer driving behaviors beyond short open-loop acceleration rollouts used in Section 6.4. In order to train the VAE pipeline on larger-scale temporal sequence data, further investigation into more effective NN architectures might be required, e.g., recurrent neural networks (RNNs) [100, 101]. However, extra care needs to be taken in utilizing data-driven models in safety-critical applications, such as autonomous driving. As is shown in Section 6.4, in the case of out-of-distribution (OOD) data, the trained pipeline can give a wrong but certain prediction. This trustworthy issue is commonly observed in learning-based approaches. Identifying and dealing with these OOD predictions still needs much research [97].

Beyond the VAE framework that predicts interaction-aware objective beliefs, a planning approach utilizing the resultant beliefs is especially interesting. Various approaches can employ the belief information, e.g., POMDP [81–83] or contingency planning [79]. In the context of dynamic games, an approximate solution approach to partially observable stochastic games (POSGs) that account for players' objective uncertainty is especially interesting. Previous works [82, 102] have explored solutions to POSGs with uni-modal uncertainties for tractability. Through approximating belief update steps using predicted beliefs by the VAE, a tractable approximate POSG solution that handles multi-modal uncertainty can be anticipated.

The third line of future work is end-to-end planning pipelines. The differentiability feature of the proposed solver provides a possibility of bridging the gap between optimization-based and learning-based approaches. The VAE line of work above is one example. Furthermore, the learning end can be extended to directly operate on raw sensor data, such as images, to exploit additional visual cues for intent inference, e.g., pedestrians' body language or gaze.

The fourth line of future work lies in optimizing the proposed adaptive MPGP planner for improved computation performance. The current optimization scheme to solve inverse games is simple gradient descent with a fixed step size. For a better convergence, a linesearch strategy can be incorporated to have an adaptive step size [23, Chapter 3] to minimize the likelihood loss. Furthermore, as pointed out in Section 5.5.2, various procedures can be investigated to reduce the run time of the proposed approach, such as low-level changes like sharing memory between MLE updates or algorithmic changes to perform intent inference asynchronously at an update rate lower than the control rate.

A

Journal Article

The first contribution of this thesis (the online adaptive MPGP solver) has led to a journal article published in *IEEE Robotics and Automation Letters (RA-L)* entitled "*Learning to Play Trajectory Games Against Opponents with Unknown Objectives* [103]," attached below.

Learning to Play Trajectory Games Against Opponents With Unknown Objectives

Xinjie Liu, *Graduate Student Member, IEEE*, Lasse Peters, *Graduate Student Member, IEEE*, and Javier Alonso-Mora, *Senior Member, IEEE*

***Abstract*—Many autonomous agents, such as intelligent vehicles, are inherently required to interact with one another. Game theory provides a natural mathematical tool for robot motion planning in such interactive settings. However, tractable algorithms for such problems usually rely on a strong assumption, namely that the objectives of all players in the scene are known. To make such tools applicable for ego-centric planning with only local information, we propose an adaptive model-predictive game solver, which jointly infers other players' objectives online and computes a corresponding generalized Nash equilibrium (GNE) strategy. The adaptivity of our approach is enabled by a differentiable trajectory game solver whose gradient signal is used for maximum likelihood estimation (MLE) of opponents' objectives. This differentiability of our pipeline facilitates direct integration with other differentiable elements, such as neural networks (NNs). Furthermore, in contrast to existing solvers for cost inference in games, our method handles not only partial state observations but also general inequality constraints. In two simulated traffic scenarios, we find superior performance of our approach over both existing game-theoretic methods and non-game-theoretic model-predictive control (MPC) approaches. We also demonstrate our approach's real-time planning capabilities and robustness in two-player hardware experiments.**

***Index Terms*—Trajectory games, multi-robot systems, integrated planning and learning, human-aware motion planning.**

I. Introduction

MANY robot planning problems, such as robot navigation in a crowded environment, involve rich interactions with other agents. Classic "predict-then-plan" frameworks neglect the fact that other agents in the scene are responsive to the ego-agent's actions. This simplification can result in inefficient or even unsafe behavior [1]. Dynamic game theory explicitly models the interactions as coupled trajectory optimization problems from a multi-agent perspective. A noncooperative equilibrium solution of this game-theoretic model then provides strategies for all players that account for the strategic coupling of plans. Beyond that, general constraints between players, such as collision avoidance, can also be handled explicitly. All of these features render game-theoretic reasoning an attractive approach to interactive motion planning.

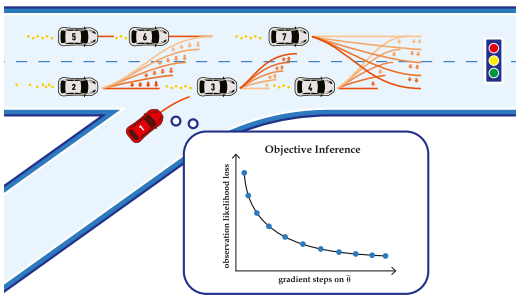


Fig. 1. An ego-agent (red) merging onto a busy road populated by six surrounding vehicles whose preferences for travel velocity and lane are initially unknown. Our approach adapts the ego agent's strategy by inferring opponents' intention parameters $\tilde{\theta}$ from partial state observations.

In order to apply game-theoretic methods for interactive motion planning from an *ego-centric* rather than *omniscient* perspective, such methods must be capable of operating only based on local information. For instance, in driving scenarios as shown in Fig. 1, the red ego-vehicle may only have partial-state observations of the surrounding vehicles and incomplete knowledge of their objectives due to unknown preferences for travel velocity, target lane, or driving style. Since vanilla game-theoretic methods require an objective model of *all* players [2], [3], this requirement constitutes a key obstacle in applying such techniques for autonomous strategic decision-making.

To address this challenge, we introduce our main contribution: a model-predictive game solver, which adapts to unknown opponents' objectives and solves for generalized Nash equilibrium (GNE) strategies. The adaptivity of our approach is enabled by a differentiable trajectory game solver whose gradient signal is used for MLE of opponents' objectives.

We perform thorough experiments in simulation and on hardware to support the following three key claims: our solver (i) outperforms both game-theoretic and non-game-theoretic baselines in highly interactive scenarios, (ii) can be combined with other differentiable components such as NNs, and (iii) is fast and robust enough for real-time planning on a hardware platform.

II. Related Work

To put our contribution into context, this section discusses four main bodies of related work. First, we discuss works on trajectory games which assume access to the objectives of all players in the scene. Then, we introduce works on inverse dynamic games that infer unknown objectives from data. Thereafter, we

Manuscript received 16 December 2022; accepted 16 April 2023. Date of publication 29 May 2023; date of current version 6 June 2023. This letter was recommended for publication by Associate Editor Yue Hu and Editor G. Venture upon evaluation of the reviewers' comments. This work was supported by European Union through ERC, INTERACT, under Grant 101041863. *(Xinjie Liu and Lasse Peters contributed equally to this work.) (Corresponding author: Xinjie Liu.)*

The authors are with the Department of Cognitive Robotics (CoR), Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: x.liu-47@student.tudelft.nl; l.peters@tudelft.nl; j.alonsomora@tudelft.nl).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3280809, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3280809

also relate our work to non-game-theoretic interaction-aware planning-techniques. Finally, we survey recent advances in differentiable optimization, which provide the underpinning for our proposed differentiable game solver.

A. N-Player General-Sum Dynamic Games

Dynamic games are well-studied in the literature [4]. In robotics, a particular focus is on multi-player general-sum games in which players may have differing yet non-adversarial objectives, and states and inputs are continuous.

Various equilibrium concepts exist in dynamic games. The Stackelberg equilibrium concept [5] assumes a "leader-follower" hierarchy, while the Nash equilibrium problem (NEP) [2], [5] does not presume such a hierarchy. Within the scope of NEP, there exist open-loop NEPs [3] and feedback NEPs [2], [6]. We refer the readers to [4] for more details about the difference between the concepts. When shared constraints exist between players, such as collision avoidance constraints, one player's feasible set may depend on other players' decisions. In that case, the problem becomes a generalized Nash equilibrium problem (GNEP) [7]. In this work, we focus on GNEPs under an open-loop information pattern which we solve by converting to an equivalent Mixed Complementarity Problem (MCP) [8].

B. Inverse Games

There are three main paradigms for solving inverse games: (i) Bayesian inference, (ii) minimization of Karush–Kuhn–Tucker (KKT) residuals, and (iii) equilibrium-constrained maximum-likelihood estimation. In type (i) methods, Le Cleac'h et al. [9] employ an Unscented Kalman Filter (UKF). This sigma-point sampling scheme drastically reduces the sampling complexity compared to vanilla particle filtering. However, a UKF is only applicable for uni-modal distributions, and extra care needs to be taken when uncertainty is multi-modal, e.g., due to multiple Nash equilibria. Type (ii) methods require *full* demonstration trajectories, i.e., including noise-free states and inputs, to cast the N-player inverse game as N independent unconstrained optimization problems [10], [11]. However, they assume full constraint satisfaction at the demonstration and have limited scalability with noisy data [12]. The type (iii) methods use KKT conditions of an open-loop Nash equilibrium (OLNE) as constraints to formulate a constrained optimization problem [12]. This type of method finds the same solution as type (ii) methods in the noise-free cases but can additionally handle partial and noisy state observations. However, encoding the equilibrium constraints is challenging, as it typically yields a non-convex problem, even in relatively simple linear-quadratic game settings. This challenge is even more pronounced when considering inequality constraints of the observed game, as this results in complementarity constraints in the inverse problem.

Our solution approach also matches the observed trajectory data in an MLE framework. In contrast to all methods above, we do so by making a GNE solver differentiable. This approach yields two important benefits over existing methods: (i) general (coupled) inequality constraints can be handled explicitly, and (ii) the entire pipeline supports direct integration with other differentiable elements, such as NNs. This latter benefit is a key motivation for our approach that is not enabled by the formulations in [9] and [12].

Note that Geiger et al. [13] explore a similar differentiable pipeline for inference of game parameters. In contrast to their work, however, our method is not limited to the special class of potential games and applies to general GNEPs.

C. Non-Game-Theoretic Interaction Models

Besides game-theoretic methods, two categories of interaction-aware decision-making techniques have been studied extensively in the context of collision avoidance and autonomous driving: (i) approaches that learn a navigation policy for the ego-agent directly without explicitly modeling the responses of others [14], [15], [16], and (ii) techniques that explicitly predict the opponents' actions to inform the ego-agent's decisions [17], [18], [19], [20], [21]. This latter category may be further split by the granularity of coupling between the ego-agent's decision-making process and the predictions of others. In the simplest case, prediction depends only upon the current physical state of other agents [22]. More advanced interaction models condition the behavior prediction on additional information such as the interaction history [17], the ego-agent's goal [19], [20], or even the ego-agent's future trajectory [18], [21].

Our approach is most closely related to this latter body of work: by solving a trajectory game, our method captures the interdependence of future decisions of all agents; and by additionally inferring the objectives of others, predictions are conditioned on the interaction history. However, a key difference of our method is that it explicitly models others as rational agents unilaterally optimizing their own cost. This assumption provides additional structure and offers a level of interpretability of the inferred behavior.

D. Differentiable Optimization

Our work is enabled by differentiating through a GNE solver. Several works have explored the idea of propagating gradient information through optimization algorithms [23], [24], [25], enabling more expressive neural architectures. However, these works focus on optimization problems and thus only apply to special cases of games, such as potential games studied by Geiger et al. [13]. By contrast, differentiating through a GNEP involves N *coupled* optimization problems. We address this challenge in Section IV-B.

III. Preliminaries

This section introduces two key concepts underpinning our work: forward and inverse dynamic games. In *forward* games, the objectives of players are known, and the task is to find players' strategies. By contrast, *inverse* games take (partial) observations of strategies as inputs to recover initially *unknown objectives*. In Section IV, we combine these two approaches into an adaptive solver that computes forward game solutions while estimating player objectives.

A. General-Sum Trajectory Games

Consider an N-player discrete-time general-sum trajectory game with horizon of T. In this setting, each player i has a control input $u_t^i \in \mathbb{R}^{m^i}$ which they may use to influence the their state $x_t^i \in \mathbb{R}^{n^i}$ at each discrete time $t \in [T]$. In this work, we assume that the evolution of each

player's state is characterized by an individual dynamical system $x^i_{t+1} = f^i(x^i_t, u^i_t)$. For brevity throughout the remainder of the letter, we shall use boldface to indicate aggregation over players and capitalization for aggregation over time, e.g., $\mathbf{x}_t := (x^1_t, \dots, x^N_t)$, $U^i := (u^i_1, \dots, u^i_T)$, $\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_T)$. With a joint trajectory starting at a given initial state $\hat{\mathbf{x}}_1 := (\hat{x}^1_1, \dots, \hat{x}^N_1)$, each player seeks to find a control sequence U^i to minimize their own cost function $J^i(\mathbf{X}, U^i; \theta^i)$, which depends upon the joint state trajectory $\mathbf{X}$ as well as the player's control input sequence U^i and, additionally, takes in a parameter vector θ^i.[1] Each player must additionally consider private inequality constraints ${}^p g^i(X^i, U^i) \geq 0$ as well as shared constraints ${}^s g(\mathbf{X}, \mathbf{U}) \geq 0$. This latter type of constraint is characterized by the fact that all players have a shared responsibility to satisfy it, with a common example being collision avoidance constraints between players. In summary, this noncooperative trajectory game can be cast as a tuple of N coupled trajectory optimization problems:

$$\forall i \in [N] \begin{cases} \min\limits_{X^i, U^i} & J^i(\mathbf{X}, U^i; \theta^i) \\ \text{s.t.} & x^i_{t+1} = f^i(x^i_t, u^i_t), \forall t \in [T-1] \\ & x^i_1 = \hat{x}^i_1 \\ & {}^p g^i(X^i, \ U^i) \geq 0 \\ & {}^s g(\mathbf{X}, \ \mathbf{U}) \geq 0. \end{cases} \tag{1}$$

Note that each player's feasible set in this problem may depend upon the decision variables of others, which makes it a GNEP rather than a standard NEP [7].

A solution of this problem is a tuple of GNE strategies $\mathbf{U}^* := (U^{1*}, \dots, U^{N*})$ that satisfies the inequalities $J^i(\mathbf{X}^*, U^{i*}; \theta^i) \leq J^i((X^i, \mathbf{X}^{\neg i*}), U^i; \theta^i)$ for any feasible deviation (X^i, U^i) of any player i, with $\mathbf{X}^{\neg i}$ denoting all but player i's states. Since identifying a global GNE is generally intractable, we require these conditions only to hold locally. At a local GNE, then, no player has a unilateral incentive to deviate *locally* in feasible directions to reduce their cost.

Running example: We introduce a simple running example[2] which we shall use throughout the presentation to concretize the key concepts. Consider a tracking game played between $N = 2$ players. Let each agent's dynamics be characterized by those of a planar double-integrator, where states $x^i_t = (p^i_{x,t}, p^i_{y,t}, v^i_{x,t}, v^i_{y,t})$ are position and velocity, and control inputs $u^i_t = (a^i_{x,t}, a^i_{y,t})$ are acceleration in horizontal and vertical axes in a Cartesian frame. We define the game's state as the concatenation of the two players' individual states $\mathbf{x}_t := (x^1_t, x^2_t)$. Each player's objective is characterized by an individual cost

$$\begin{aligned} J^i = & \sum_{t=1}^{T-1} \|p^i_{t+1} - p^i_{\text{goal}}\|_2^2 + 0.1\|u^i_t\|_2^2 \\ & + 50 \max(0, d_{\min} - \|p^i_{t+1} - p^{-i}_{t+1}\|_2)^3, \end{aligned} \tag{2}$$

where we set $p^1_{\text{goal}} = p^2_t$ so that player 1, the tracking robot, is tasked to track player 2, the target robot. Player 2 has a fixed goal point p^2_{goal}. Both agents wish to get to their goal position efficiently while avoiding proximity beyond a minimal distance $d_{\min}$. Players also have shared collision avoidance constraints ${}^s g_{t+1}(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) = \|p^1_{t+1} - p^2_{t+1}\|_2 - d_{\min} \geq 0, \forall t \in [T-1]$ and private bounds on state and controls ${}^p g^i(X^i, U^i)$. Agents need to negotiate and find an underlying equilibrium strategy in this noncooperative game, as no one wants to deviate from the direct path to their goal.

[1]The role of the parameters will become clear later in the letter when we move on to *inverse* dynamic games.

[2]Our final evaluation in Section V features denser interaction such as the 7-player ramp-merging scenario shown in Fig. 1.

B. Inverse Games

We now switch context to the *inverse* dynamic game setting. Let $\theta := (\hat{\mathbf{x}}_1, \theta^2, \dots, \theta^N)$ denote the aggregated tuple of parameters initially unknown to the ego-agent with index 1. Note that we explicitly infer the initial state of a game $\hat{\mathbf{x}}_1$ to account for the potential sensing noise and partial state observations. To model the inference task over these parameters, we assume that the ego-agent observes behavior originating from an unknown Nash game $\Gamma(\theta) := (\hat{\mathbf{x}}_1, {}^s g, \{f^i, {}^p g^i, J^i(\cdot; \theta^i)\}_{i \in [N]})$, with objective functions and constraints parameterized by initially unknown values θ^i and $\hat{\mathbf{x}}_1$, respectively.

Similar to the existing method [12], we employ an MLE formulation to allow observations to be *partial* and *noise-corrupted*. In contrast to that method, however, we also allow for inequality constraints in the hidden game. That is, we propose to solve

$$\begin{aligned} \max_{\theta, \mathbf{X}, \mathbf{U}} \quad & p(\mathbf{Y} \mid \mathbf{X}, \mathbf{U}) \\ \text{s.t.} \quad & (\mathbf{X}, \mathbf{U}) \text{ is a GNE of } \Gamma(\theta) \end{aligned} \tag{3}$$

where $p(\mathbf{Y} \mid \mathbf{X}, \mathbf{U})$ denotes the likelihood of observations $\mathbf{Y} := (\mathbf{y}_1, \dots, \mathbf{y}_T)$ given the estimated game trajectory $(\mathbf{X}, \mathbf{U})$ induced by parameters θ. This formulation yields an *mathematical program with equilibrium constraints (MPEC)* [26], where the outer problem is an estimation problem while the inner problem involves solving a dynamic game. When the observed game includes inequality constraints, the resulting inverse problem necessarily contains complementarity constraints and only few tools are available to solve the resulting problem. In the next section, we show how to transform (3) into an unconstrained problem by making the inner game differentiable, which also enables combination with other differentiable components.

Running example: We assign the tracker (player 1) to be the ego-agent and parameterize the game with the goal position of the target robot $\theta^2 = p^2_{\text{goal}}$. That is, the tracker does not know the target agent's goal and tries to infer this parameter from position observations. To ensure that (3) remains tractable, the ego-agent maintains only a fixed-length buffer of observed opponent's positions. Note that solving the inverse game requires solving games rather than optimal control problems at the inner level to account for the noncooperative nature of observed interactions, which is different from inverse optimal control (IOC) even in the 2-player case. We employ a Gaussian observation model, which we represent with an equivalent negative log-likelihood objective $\|\mathbf{Y} - r(\mathbf{X}, \mathbf{U})\|_2^2$ in (3), where $r(\mathbf{X}, \mathbf{U})$ maps $(\mathbf{X}, \mathbf{U})$ to the corresponding sequence of expected positions.

IV. Adaptive Model-Predictive Game Play

We wish to solve the problem of model-predictive game play (MPGP) from an ego-centric perspective, i.e., without prior knowledge of other players' objectives. To this end, we present an adaptive model-predictive game solver that combines the tools of Section III: first, we perform MLE of unknown objectives by solving an *inverse game* (Section III-B); then, we

solve a *forward game* using this estimate to recover a strategic motion plan (Section III-A).

A. Forward Games as MCPs

We first discuss the conversion of the GNEP in (1) to an equivalent MCP. There are three main advantages of taking this view. First, there exists a wide range of off-the-shelf solvers for this problem class [27]. Furthermore, MCP solvers directly recover strategies for all players *simultaneously*. Finally, this formulation makes it easier to reason about derivatives of the solution w.r.t. to problem data. As we shall discuss in Section IV-C, this derivative information can be leveraged to solve the inverse game problem of (3).

In order to solve the GNEP presented in (1) we derive its first-order necessary conditions. We collect all equality constraints for player i in (1) into a vector-valued function $h^i(X^i, U^i; \hat{x}_1^i)$, introduce Lagrange multipliers μ^i, ${}^p\lambda^i$ and ${}^s\lambda$ for constraints $h^i(X^i, U^i; \hat{x}_1^i)$, ${}^p g^i(X^i, U^i)$, and ${}^s g(\mathbf{X}, \mathbf{U})$ and write the Lagrangian for player i as

$$\begin{aligned}\mathcal{L}^i(\mathbf{X}, \mathbf{U}, \mu^i, {}^p\lambda^i, {}^s\lambda; \theta) &= J^i(\mathbf{X}, \mathbf{U}; \theta^i) \\ &+ \mu^{i\top} h^i(X^i, U^i; \hat{x}_1^i) - {}^s\lambda^\top {}^s g(\mathbf{X}, \mathbf{U}) - {}^p\lambda^{i\top} {}^p g^i(X^i, U^i).\end{aligned} \tag{4}$$

Note that we share the multipliers associated with shared constraints between the players to encode equal constraint satisfaction responsibility [28]. Under mild regularity conditions, e.g., linear independence constraint qualification (LICQ), a solution of (1) must satisfy the following joint KKT conditions:

$$\begin{aligned}&\forall i \in [N] \begin{cases} \nabla_{(X^i, U^i)} \mathcal{L}^i(\mathbf{X}, \mathbf{U}, \mu^i, {}^p\lambda^i, {}^s\lambda; \theta) = 0 \\ 0 \leq {}^p g^i(X^i, U^i) \perp {}^p\lambda^i \geq 0 \end{cases} \\ &h(\mathbf{X}, \mathbf{U}; \hat{\mathbf{x}}_1) = 0 \\ &0 \leq {}^s g(\mathbf{X}, \mathbf{U}) \perp {}^s\lambda \geq 0,\end{aligned} \tag{5}$$

where, for brevity, we denote by $h(\mathbf{X}, \mathbf{U}; \hat{\mathbf{x}}_1)$ the aggregation of all equality constraints. If the second directional derivative of the Lagrangian is positive along all feasible directions at a solution of (5)—a condition that can be checked a posteriori—this point is also a solution of the original game. In this work, we solve trajectory games by viewing their KKT conditions through the lens of MCPs [8, Section 1.4.2].

Definition 1: A Mixed Complementarity Problem (MCP) is defined by the following problem data: a function $F(z) : \mathbb{R}^d \mapsto \mathbb{R}^d$, lower bounds $\ell_j \in \mathbb{R} \cup \{-\infty\}$ and upper bounds $u_j \in \mathbb{R} \cup \{\infty\}$, each for $j \in [d]$. The solution of an MCP is a vector $z^* \in \mathbb{R}^n$, such that for each element with index $j \in [d]$ one of the following equations holds:

$$z_j^* = \ell_j, F_j(z^*) \geq 0 \tag{6a}$$

$$\ell_j < z_j^* < u_j, F_j(z^*) = 0 \tag{6b}$$

$$z_j^* = u_j, F_j(z^*) \leq 0. \tag{6c}$$

The parameterized KKT system of (5) can be expressed as a *parameterized family* of MCPs with decision variables corresponding to the primal and dual variables of (5),

$$z = \left[\mathbf{X}^\top, \mathbf{U}^\top, \boldsymbol{\mu}^\top, {}^p\lambda^{1\top}, \ldots, {}^p\lambda^{N\top}, {}^s\lambda^\top\right]^\top,$$

and problem data

$$F(z;\theta) = \begin{bmatrix} \nabla_{(X^1,U^1)}\mathcal{L}^i \\ \vdots \\ \nabla_{(X^N,U^N)}\mathcal{L}^N \\ h \\ {}^p g^1 \\ \vdots \\ {}^p g^N \\ {}^s g \end{bmatrix}, \quad \ell = \begin{bmatrix} -\infty \\ \vdots \\ -\infty \\ -\infty \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad u = \begin{bmatrix} \infty \\ \vdots \\ \infty \\ \infty \\ \infty \\ \vdots \\ \infty \\ \infty \end{bmatrix}, \tag{7}$$

where, by slight abuse of notation, we overload F to be parametrized by θ via $\mathcal{L}^i$ and use ∞ to denote elements for which upper or lower bounds are dropped.

B. Differentiation of an MCP Solver

An MCP solver may be viewed as a function, mapping problem data to a solution vector. Taking this perspective, for a parameterized family of MCPs as in (7), we wish to compute the function's derivatives to answer the following question: How does the solution z^* respond to local changes of the problem parameters θ?

1) The Nominal Case: Let $\Psi(\theta) := (F(\cdot;\theta), \ell, u)$ denote an MCP parameterized by $\theta \in \mathbb{R}^p$ and let $z^* \in \mathbb{R}^n$ denote a solution of that MCP, which is implicitly a function of θ. For this nominal case, we consider only solutions at which *strict complementarity* holds. We shall relax this assumption later. If F is smooth, i.e., $F(\cdot;\theta), F(z^*;\cdot) \in C^1$, we can recover the Jacobian matrix $\nabla_\theta z^* = (\frac{\partial z_j^*}{\partial \theta_k}) \in \mathbb{R}^{n \times p}$ by distinguishing two possible cases. For brevity, below, gradients are understood to be evaluated at z^* and θ.

a) Active bounds: Consider first the elements z_j^* that are either at their lower or upper bound, i.e., z_j^* satisfies (6a) or (6c). Since strict complementarity holds at the solution, $F_j(z^*;\theta)$ must be bounded away from zero with a finite margin. Hence, the smoothness of F guarantees that a local perturbation of θ will retain the sign of $F_j(z^*;\theta)$. As a result, z_j^* remains at its bound and, locally, is identically zero. Let $\tilde{\mathcal{I}} := \{k \in [n] \mid z_k^* = \ell_k \vee z_k^* = u_k\}$ denote the index set of all elements matching this condition and $\tilde{z}^* := [z^*]_{\tilde{\mathcal{I}}}$ denote the solution vector reduced to that set. Trivially, then, the Jacobian of this vector vanishes, i.e., $\nabla_\theta \tilde{z}^* = 0$.

b) Inactive bounds: The second case comprises elements that are strictly between the bounds, i.e., z_j^* satisfying (6b). In this case, under mild assumptions on F, for any local perturbation of θ there exists a perturbed solution such that F remains at its root. Therefore, the gradient $\nabla_\theta z_j^*$ for these elements is generally non-zero, and we can compute it via the implicit function theorem (IFT). Let $\bar{\mathcal{I}} := \{k \in [n] \mid F_k(z^*;\theta) = 0, \ell_k < z_k^* < u_k\}$ be the index set of all elements satisfying case (b) and let

$$\bar{z}^* := [z^*]_{\bar{\mathcal{I}}}, \quad \bar{F}(z^*, \theta) := [F(z^*;\theta)]_{\bar{\mathcal{I}}} \tag{8}$$

denote the solution vector and its complement reduced to said index set. By the IFT, the relationship between parameters θ and

solution $z^*(\theta)$ is characterized by the stationarity of $\bar{F}$:

$$\begin{aligned} 0 &= \nabla_\theta \left[\bar{F}(z^*(\theta), \theta)\right] \\ &= \nabla_\theta \bar{F} + (\nabla_{\bar{z}^*} \bar{F})(\nabla_\theta \bar{z}^*) + (\nabla_{\bar{z}^*} \bar{F}) \underbrace{(\nabla_\theta \tilde{z}^*)}_{\equiv 0} \end{aligned} \tag{9}$$

Note that, as per the discussion in case (a), the last term in this equation is identically zero. Hence, if the Jacobian $\nabla_{\bar{z}^*} \bar{F}$ is invertible, we recover the derivatives as the unique solution of the above system of equations,

$$\nabla_\theta \bar{z}^* = -\left(\nabla_{\bar{z}^*} \bar{F}\right)^{-1} (\nabla_\theta \bar{F}). \tag{10}$$

Note that (9) may not always have a unique solution, in which case (10) cannot be evaluated. We discuss practical considerations for this special case below.

2) Remarks on Special Cases and Practical Realization: The above derivation of gradients for the nominal case involves several assumptions on the structure of the problem. We discuss considerations to improve numerical robustness for practical realization of this approach below. We note that both special cases discussed hereafter are rare in practice. In fact, across 100 simulations of the running example with varying initial states and objectives, neither of them occurred.

a) Weak complementarity: The nominal case discussed above assumes strict complementarity at the solution. If this assumption does not hold, the derivative of the MCP is not defined. Nevertheless, we can still compute subderivatives at θ. Let the set of all indices for which this condition holds be denoted by $\hat{\mathcal{I}} := \{k \in [n] \mid F_k(z^*; \theta) = 0 \wedge z_k^* \in \{\ell_k, u_k\}\}$. Then by selecting a subset of $\hat{\mathcal{I}}$ and including it in $\bar{\mathcal{I}}$ for evaluation of (10), we recover a subderivative.

b) Invertibility: The evaluation (10) requires invertibility of $\nabla_{\bar{z}^*} \bar{F}$. To this end, we compute the least-squares solution of (9) rather than explicitly inverting $\nabla_{\bar{z}} \bar{F}$.

C. Model-Predictive Game Play With Gradient Descent

Finally, we present our pipeline for adaptive game-play against opponents with unknown objectives. Our adaptive MPGP scheme is summarized in Algorithm 1. At each time step, we first update our estimate of the parameters by approximating the inverse game in (3) via gradient descent. To obtain an unconstrained optimization problem, we substitute the constraints in (3) with our differentiable game solver. Following the discussion of (7), we denote by $z^*(\theta)$ the solution of the MCP formulation of the game parameterized by θ. Furthermore, by slight abuse of notation, we overload $\mathbf{X}(z^*), \mathbf{U}(z^*)$ to denote functions that extract the state and input vectors from z^*. Then, the inverse game of (3) can be written as unconstrained optimization,

$$\max_\theta \quad p(\mathbf{Y} \mid \mathbf{X}(z^*(\theta)), \mathbf{U}(z^*(\theta))). \tag{11}$$

Online, we approximate solutions of this problem by taking gradient descent steps on the negative logarithm of this objective, with gradients computed by chain rule,

$$\begin{aligned} &\nabla_\theta \left[p(\mathbf{Y} \mid \mathbf{X}(z^*(\theta)), \mathbf{U}(z^*(\theta))\right] \\ &\quad = (\nabla_{\mathbf{X}} p)(\nabla_{z^*} \mathbf{X})(\nabla_\theta z^*) + (\nabla_{\mathbf{U}} p)(\nabla_{z^*} \mathbf{U})(\nabla_\theta z^*). \end{aligned} \tag{12}$$

Here, the only non-trivial term is $\nabla_\theta z^*$, whose computation we discussed in Section IV-B. To reduce the computational cost, we warm-start using the estimate of the previous time step and terminate early if a maximum number of steps is reached. Then, we solve a forward game parametrized by the estimated $\tilde{\theta}$ to compute control commands. We execute the first control input for the ego agent and repeat the procedure.

Algorithm 1: Adaptive MPGP.

Hyper-parameters: stopping tolerance: stop_tol, learning rate: lr
Input: initial $\tilde{\theta}$, current observation buffer $\mathbf{Y}$, new observation $\mathbf{y}$
$\mathbf{Y} \leftarrow$ updateBuffer$(\mathbf{Y}, \mathbf{y})$
`/* inverse game approximation */`
while not stop_tol and not max_steps_reached **do**
  $(z^*, \nabla_\theta z^*) \leftarrow$ solveDiffMCP$(\tilde{\theta})$ ▷ sec. IV-B
  $\nabla_\theta p \leftarrow$ composeGradient$(z^*, \nabla_\theta z^*, \mathbf{Y})$ ▷ eq. (12)
  $\tilde{\theta} \leftarrow \tilde{\theta} - \nabla_\theta p \cdot$ lr
end
$z^* \leftarrow$ solveMCP$(\tilde{\theta})$ ▷ forward game, eq. (7)
applyFirstEgoInput(z^*)
return $\tilde{\theta}, \mathbf{Y}$

V. Experiments

To evaluate our method, we compare against two baselines in Monte Carlo studies of simulated interaction. Beyond these quantitative results, we showcase our method deployed on Jackal ground robots in two hardware experiments.

The experiments below are designed to support the key claims that our method (i) outperforms both game-theoretic and non-game-theoretic baselines in highly interactive scenarios, (ii) can be combined with other differentiable components such as NNs, and (iii) is sufficiently fast and robust for real-time planning on a hardware platform. A supplementary video of qualitative results can be found at https://xinjie-liu.github.io/projects/gamehttps://xinjie-liu.github.io/projects/game. Upon publication of this manuscript, the code for our method and experiments will be available at the same link.

A. Experiment Setup

1) Scenarios: We evaluate our method in two scenarios.

a) 2-player running example: To test the inference accuracy and convergence of our method in an intuitive setting, we first consider the 2-player running example. For evaluation in simulation, we sample the opponent's intent—i.e., their unknown goal position in (2)— uniformly from the environment. Partial observations comprise the position of each agent.

b) Ramp merging: To demonstrate the scalability of our approach and support the claim that our solver outperforms the baselines in highly interactive settings, we also test our method on a ramp merging scenario with varying numbers of players. This experiment is inspired by the setup used in [3] and is schematically visualized in Fig. 1. We model each player's dynamics by a discrete-time kinematic bicycle with the state comprising position, velocity and orientation, i.e., $x_t^i = (p_{x,t}^i, p_{y,t}^i, v_t^i, \psi_t^i)$, and controls comprising acceleration and steering angle, i.e., $u_t^i = (a_t^i, \phi)$. We capture their individual behavior by a cost function that penalizes deviation from a reference travel velocity and target lane; i.e., $\theta^i = (v_{\text{ref}}^i, p_{\text{y,lane}}^i)$. We add constraints for lane boundaries, for limits on speed, steering,

and acceleration, for the traffic light, and for collision avoidance. To encourage rich interaction in simulation, we sample each agent's initial state by sampling their speed and longitudinal positions uniformly at random from the intervals from zero to maximum velocity $v_{\max}$ and four times the vehicle length l_{car}, respectively. The ego-agent always starts on the ramp and all agents are initially aligned with their current lane. Finally, we sample each opponent's intent from the uniform distribution over the two lane centers and the target speed interval $[0.4v_{\max}, v_{\max}]$. Partial observations comprise the position and orientation of each agent.

2) Baselines: We consider the following three baselines.

a) KKT-constrained solver: In contrast to our method, the solver by Peters et al. [12] has no support for either private or shared inequality constraints. Consequently, this baseline can be viewed as solving a simplified version of the problem in (3) where the inequality constraints associated with the inner-level GNEP are dropped. Nonetheless, we still use a cubic penalty term as in (2) to encode soft collision avoidance. Furthermore, for fair comparison, we only use the baseline to *estimate* the objectives but compute control commands from a GNEP considering all constraints.

b) MPC with constant-velocity predictions: This baseline assumes that opponents move with constant velocity as observed at the latest time step. We use this baseline as a representative method for predictive planning approaches that do not explicitly model interaction.

c) Heuristic estimation MPGP: To highlight the importance of online intent inference, for the ramp merging evaluation, we also compare against a game-theoretic baseline that assumes a fixed intent for all opponents. This fixed intent is recovered by taking each agent's initial lane and velocity as a heuristic preference estimate.

To ensure a fair comparison, we use the same MCP backend [29] to solve all GNEPs and optimization problems with a default convergence tolerance of $1e^{-6}$. Furthermore, all planners utilize the same planning horizon and history buffer size of 10 time steps with a time-discretization of 0.1s. For the iterative MLE solve procedure in the 2-player running example and the ramp merging scenario, we employ a learning rate of $2e^{-2}$ for objective parameters and $1e^{-3}$ for initial states. We terminate maximum likelihood estimation iteration when the norm of the parameter update step is smaller than $1e^{-4}$, or after a maximum of 30 steps. Finally, opponent behavior is generated by solving a separate ground-truth game whose parameters are hidden from the ego-agent.

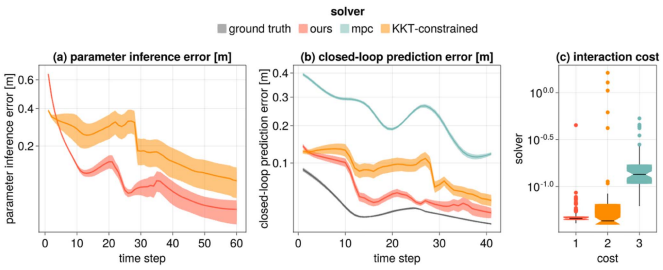


Fig. 2. Monte Carlo study for the 2-player tracking game for 100 trials. Solid lines and ribbons in (a) and (b) indicate the mean and standard error of the mean. Cost distributions in (c) are normalized by subtracting ground truth costs.

TABLE I
MONTE CARLO STUDY FOR THE RAMP MERGING SCENARIO DEPICTED IN FIG. 1 WITH 100 TRIALS FOR SETTINGS WITH 3, 5, AND 7 PLAYERS. EXCEPT FOR COLLISION AND INFEASIBLE SOLVE TIMES, ALL METRICS ARE REPORTED BY MEAN AND STANDARD ERROR OF THE MEAN

Set.	Method	Ego cost	Opp. cost	Coll.	Inf.	Traj. err. [m]	Param. err.	Time [s]
3 player	Ours	0.64 ± 0.36	0.06 ± 0.03	0	0	1.29 ± 0.05	0.41 ± 0.03	0.081 ± 0.002
	KKT-con	1.85 ± 1.21	0.05 ± 0.02	0	1	1.32 ± 0.06	2.39 ± 0.11	0.060 ± 0.002
	Heuristic	6.73 ± 2.40	0.09 ± 0.07	0	11	7.89 ± 0.26	3.96 ± 0.13	0.008 ± 0.001
	MPC	1.50 ± 0.45	0.33 ± 0.07	28	218	2.40 ± 0.11	n/a	0.009 ± 0.002
5 player	Ours	0.56 ± 0.43	0.16 ± 0.06	0	2	1.66 ± 0.07	0.47 ± 0.03	0.29 ± 0.02
	KKT-con	0.07 ± 0.32	0.06 ± 0.02	1	4	1.70 ± 0.06	2.15 ± 0.06	0.28 ± 0.02
	Heuristic	2.06 ± 0.44	0.35 ± 0.10	5	25	8.05 ± 0.19	2.91 ± 0.07	0.015 ± 0.001
	MPC	5.73 ± 2.91	0.42 ± 0.13	44	552	2.87 ± 0.13	n/a	0.014 ± 0.002
7 player	Ours	1.60 ± 1.19	0.06 ± 0.02	1	1	1.89 ± 0.05	0.46 ± 0.02	0.68 ± 0.02
	KKT-con	3.11 ± 1.72	0.09 ± 0.04	7	22	2.01 ± 0.06	1.93 ± 0.03	0.63 ± 0.06
	Heuristic	6.60 ± 1.67	0.27 ± 0.06	8	8	8.18 ± 0.15	2.44 ± 0.05	0.031 ± 0.002
	MPC	8.41 ± 1.45	0.59 ± 0.09	43	848	3.07 ± 0.08	n/a	0.0274 ± 0.004

B. Simulation Results

To compare the performance of our method to the baselines described in Section V-A2, we conduct a Monte Carlo study for the two scenarios described in Section V-A1.

1) 2-Player Running Example: Fig. 2 summarizes the results for the 2-player running example. For this evaluation, we filter out any runs for which a solver resulted in a collision. For our solver, the KKT-constrained baseline, and the MPC baseline this amounts to 2, 2 and 13 out of 100 episodes, respectively.

Figs. 2(a)–(b) show the prediction error of the goal position and opponent's trajectory, each of which is measured by ℓ^2-norm. Since the MPC baseline does not explicitly reason about costs of others, we do not report parameter inference error for it in Fig. 2(a). As evident from this visualization, both game-theoretic methods give relatively accurate parameter estimates and trajectory predictions. Among these methods, our solver converges more quickly and consistently yields a lower error. By contrast, MPC gives inferior prediction performance with reduced errors only in trivial cases, when the target robot is already at the goal. Fig. 2(c) shows the distribution of costs incurred by the ego-agent for the same set of experiments. Again, game-theoretic methods yield better performance and our method outperforms the baselines with more consistent and robust behaviors, indicated by fewer outliers and lower variance in performance.

2) Ramp Merging: Table I summarizes the results of for the simulated ramp-merging scenario for 3, 5, and 7 players.

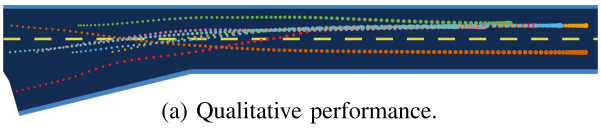
a) Task performance: To quantify the task performance, we report costs as an indicator for interaction efficiency, the number of collisions as a measure of safety, number of infeasible solves as an indicator of robustness, and trajectory and parameter error as a measure of inference accuracy. On a high level, we observe that the game-theoretic methods generally outperform the other baselines; especially for the settings with higher traffic density. While MPC achieves high efficiency (ego-cost) in the 3-player case, it collides significantly more often than the other methods across all settings. Among the game-theoretic approaches, we observe that online inference of opponent intents—as performed by our method and the KKT-constrained baseline—yields better performance than a game that uses a heuristic estimate of the intents. Within the inference-based game solvers, a Manning-Whitney U-test reveals that, across all settings, both methods achieve an ego-cost that is significantly lower than all other baselines but not significantly higher than solving the game with ground truth opponent intents. Despite this tie in terms of interaction *efficiency*, we observe a statistically significant improvement of our method over the KKT-constrained baseline in terms of *safety:* in the highly interactive 7-player case, the KKT-constrained baseline collides seven times more often than our method. This advantage is enabled by our method's ability to model inequality constraints within the inverse game.

b) Computation time: We also measure the computation time of each approach. The inference-based game solvers have generally a higher runtime than the remaining methods due to the added complexity. Within the inference methods, our method is only marginally slower than the KKT-constrained baseline, despite solving a more complex problem that includes inequality constraints. The average number of MLE updates for our method was 11.0, 19.2, and 22.7 for the 3, 5, and 7-player setting, respectively. While our current implementation achieves real-time planning rates only for up to three players, we note that additional optimizations may further reduce the runtime of our approach. Among such optimizations are low-level changes such as sharing memory between MLE updates as well as algorithmic changes to perform intent inference asynchronously at an update rate lower than the control rate. We briefly explore another algorithmic optimization in the next section.

3) Combination With an NN: To support the claim that our method can be combined with other differentiable modules, we demonstrate the integration with an NN. For this proof of concept, we use a two-layer feed-forward NN, which takes the buffer of recent partial state observations as input and predicts other players' objectives. Training of this module is enabled by propagating the gradient of the observation likelihood loss of (11) through the differentiable game solver to the parameters of the NN. Online, we use the network's prediction as an initial guess to reduce the number gradient steps. As summarized in Fig. 3 , this combination reduces the computation time by more than 60% while incurring only a marginal loss in performance.

C. Hardware Experiments

To support the claim that our method is sufficiently fast and robust for hardware deployment, we demonstrate the tracking game in the running example in Section III-A with a Jackal ground robot tracking (i) another Jackal robot (Fig. 4(a)) and (ii) a human player (Fig. 4(b)), each with initially unknown goals. Plans are computed online on a mobile i7 CPU. We generate plans using the point mass dynamics with a velocity constraint of $0.8\ \mathrm{ms}^{-1}$ and realize low-level control via the feedback controller of [30]. A video of these hardware demonstrations is included in the supplementary material. In both experiments, we observe that our adaptive MPGP planner enables the robot to infer the unknown goal position to track the target while avoiding collisions. The average computation time in both experiments was 0.035 s.

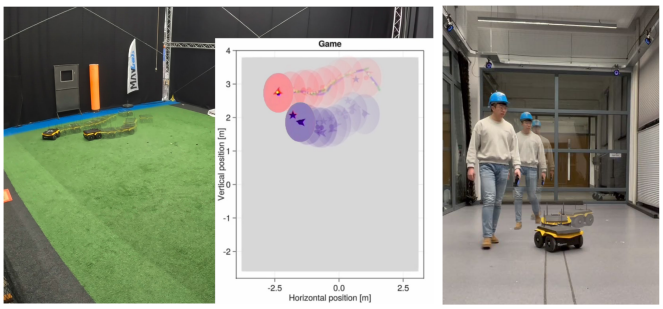


(a) Qualitative performance.

Ego cost	Opp. cost	Coll.	Inf.	Traj. err. [m]	Param. err.	Time [s]
2.19 ± 1.21	0.17 ± 0.07	3	5	2.34 ± 0.08	0.91 ± 0.08	0.274 ± 0.01

(b) Quantitative performance.

Fig. 3. Performance of our solver in combination with an NN for 100 trials of the 7-player ramp merging scenario.



(a) (b)

Fig. 4. Time lapse of the running-example in which a Jackal tracks (a) another Jackal and (b) a human. Overlaid in (a) are the position of target robot (red) its true goal (red star), the tracker (blue), and its goal estimate (blue star).

VI. Conclusion

In this letter, we presented a model-predictive game solver that adapts strategic motion plans to initially unknown opponents' objectives. The adaptivity of our approach is enabled by a differentiable trajectory game solver whose gradient signal is used for MLE of unknown game parameters. As a result, our adaptive MPGP planner allows for safe and efficient interaction with other strategic agents without assuming prior knowledge of their objectives or observations of full states. We evaluated our method in two simulated interaction scenarios and demonstrated superior performance over a state-of-the-art game-theoretic planner and a non-interactive MPC baseline. Beyond that, we demonstrated the real-time planning capability and robustness of our approach in two hardware experiments.

In this work, we have limited inference to parameters that appear in the objectives of other players. Since the derivation of the gradient in Section IV-B can also handle other parameterizations of F—so long as they are smooth—future work may extend this framework to infer additional parameters of

constraints or aspects of the observation model. Furthermore, encouraged by the improved scalability when combining our method with learning modules such as NNs, we seek to extend this learning pipeline in the future. One such extension would be to operate directly on raw sensor data, such as images, to exploit additional visual cues for intent inference. Another extension is to move beyond MLE-based point estimates to inference of potentially multi-modal distributions over opponent intents, which may be achieved by embedding our differentiable method within a variational autoencoder. Finally, our framework could be tested on large-scale datasets of real autonomous-driving behavior.

ACKNOWLEDGMENT

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 797–803.

[2] D. Fridovich-Keil, E. Ratner, L. Peters, A. D. Dragan, and C. J. Tomlin, "Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1475–1481.

[3] L. Cleac'h, M. Schwager, and Z. Manchester, "ALGAMES: A fast augmented Lagrangian solver for constrained dynamic games," *Auton. Robots*, vol. 46, no. 1, pp. 201–215, 2022.

[4] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA, USA: SIAM, 1999.

[5] A. Liniger and J. Lygeros, "A noncooperative game approach to autonomous racing," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 884–897, May 2020.

[6] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, "The computation of approximate generalized feedback Nash equilibria," *SIAM J. Optim.*, vol. 33, no. 1, pp. 294–318, 2023.

[7] F. Facchinei and C. Kanzow, "Generalized Nash equilibrium problems," *Ann. Operations Res.*, vol. 175, no. 1, pp. 177–211, 2010.

[8] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Berlin, Germany: Springer, 2003.

[9] L. Cleac'h, M. Schwager, and Z. Manchester, "LUCIDGames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5485–5492, Jul. 2021.

[10] C. Awasthi and A. Lamperski, "Inverse differential games with mixed inequality constraints," in *Proc. IEEE Amer. Control Conf.*, 2020, pp. 2182–2187.

[11] S. Rothfuß, J. Inga, F. Köpf, M. Flad, and S. Hohmann, "Inverse optimal control for identification in non-cooperative differential games," *IFAC-LettersOnLine*, vol. 50, no. 1, pp. 14909–14915, 2017.

[12] L. Peters, D. Fridovich-Keil, V. R. Royo, C. J. Tomlin, and C. Stachniss, "Inferring objectives in continuous dynamic games from noise-corrupted partial state observations," in *Proc. Robot.: Sci. Syst.*, 2021. [Online]. Available: https://www.roboticsproceedings.org/rss18/p051.html

[13] P. Geiger and C.-N. Straehle, "Learning game-theoretic models of multiagent trajectories using implicit layers," in *Proc. Conf. Advancements Artif. Intell.*, 2021, pp. 4950–4958.

[14] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots Syst.*, 2018, pp. 3052–3059.

[15] B. Brito, M. Everett, J. P. How, and J. Alonso-Mora, "Where to go next: Learning a subgoal recommendation policy for navigation in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4616–4623, Jul. 2021.

[16] V. Tolani, S. Bansal, A. Faust, and C. Tomlin, "Visual navigation among humans with optimal control as a supervisor," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2288–2295, Apr. 2021.

[17] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, 2016.

[18] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1–9.

[19] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: Prediction conditioned on goals in visual multi-agent settings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2821–2830.

[20] J. Roh, C. Mavrogiannis, R. Madan, D. Fox, and S. Srinivasa, "Multimodal trajectory prediction via topological invariance for navigation at uncontrolled intersections," in *Proc. Conf. Robot Learn.*, 2021, pp. 2216–2227.

[21] M. Sun, F. Baldini, P. Trautman, and T. Murphey, "Move beyond trajectories: Distribution space coupling for crowd navigation," in *Proc. Robot.: Sci. Syst.*, 2021. [Online]. Available: https://www.roboticsproceedings.org/rss17/p053.html

[22] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1696–1703, Apr. 2020.

[23] D. Ralph and S. Dempe, "Directional derivatives of the solution of a parametric nonlinear program," *Math. Program.*, vol. 70, no. 1, pp. 159–172, 1995.

[24] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 136–145.

[25] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9558–9570.

[26] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs With Equilibrium Constraints*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[27] S. C. Billups, S. P. Dirkse, and M. C. Ferris, "A comparison of large scale mixed complementarity problem solvers," *Comput. Optim. Appl.*, vol. 7, no. 1, pp. 3–25, 1997.

[28] A. A. Kulkarni and U. V. Shanbhag, "On the variational equilibrium as a refinement of the generalized Nash equilibrium," *Automatica*, vol. 48, no. 1, pp. 45–55, 2012.

[29] S. P. Dirkse and M. C. Ferris, "The PATH solver: A nommonotone stabilization scheme for mixed complementarity problems," *Optim. Methods Softw.*, vol. 5, no. 2, pp. 123–156, 1995.

[30] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp. 384–389.

B

Manning-Whitney U-Tests

Table B.1 and Table B.2 show two-sided P values of Manning-Whitney U-Tests in Section 5.5.2 with a statistical significance threshold of 0.05. Table B.1 compares the baselines against the proposed solver, and Table B.2 compares all the approaches against solving games with ground truth objectives. Interaction costs used in Table B.1 are normalized with the subtraction of ground truth costs, and costs in Table B.2 are raw costs. On a high level, the gap between the ground truth and all the approaches and the gap between the proposed solver and all the baselines are more pronounced in dense scenarios. In the densest setting with 7 players, only the proposed approach matches the ground truth performance regarding both driving efficiency (ego cost) and safety (collision times).

Table B.1: Two-sided P values of Manning-Whitney U-Tests in Section 5.5.2 compare the proposed approach versus baselines. Bold numbers indicate metrics that the proposed approach is statistically significantly better.

Set.	Method	Ego cost	Opp. cost	Collision	Inf.	Traj. err. [m]	Param. err.
3 players	KKT-con	0.73	0.88	1	0.32	0.95	**6.56×10^{-29}**
	Heuristic	**1.19×10^{-8}**	**7.94×10^{-3}**	1	**0.01**	**7.76×10^{-34}**	**4.14×10^{-34}**
	MPC	0.86	0.10	**1.27×10^{-8}**	**3.29×10^{-5}**	**1.71×10^{-17}**	n/a
5 players	KKT-con	0.77	0.78	0.32	0.57	0.38	**9.86×10^{-33}**
	Heuristic	**2.33×10^{-10}**	0.17	**0.02**	0.14	**3.90×10^{-34}**	**4.96×10^{-34}**
	MPC	**9.47×10^{-12}**	**4.05×10^{-6}**	**6.88×10^{-14}**	**1.36×10^{-11}**	**8.04×10^{-20}**	n/a
7 players	Ours (without inequalities)	0.81	0.95	0.09	1	0.86	0.81
	KKT-con	0.71	0.99	**0.03**	**0.05**	0.06	**2.56×10^{-34}**
	Heuristic	**5.61×10^{-14}**	**0.05**	**0.02**	0.56	**2.56×10^{-34}**	**2.56×10^{-34}**
	MPC	**2.98×10^{-16}**	**5.92×10^{-13}**	**8.70×10^{-13}**	**2.65×10^{-18}**	**1.45×10^{-27}**	n/a

Table B.2: Two-sided P values of Manning-Whitney U-Tests in Section 5.5.2 compare the approaches versus solving games with ground truth objectives. Bold numbers indicate metrics that the ground truth is statistically significantly better.

Set.	Method	Ego cost	Opp. cost	Collision	Inf.	Traj. err. [m]	Param. err.
3 players	Ours	0.89	0.87	1	1	0.12	**5.64** $\times$**10**$^{-39}$
	KKT-con	0.84	0.88	1	0.32	0.12	**5.64** $\times$**10**$^{-39}$
	Heuristic	0.14	0.97	1	**0.01**	**3.90** $\times$**10**$^{-34}$	**5.64** $\times$**10**$^{-39}$
	MPC	0.50	0.25	**1.27** $\times$**10**$^{-8}$	**3.29** $\times$**10**$^{-5}$	**5.73** $\times$**10**$^{-20}$	n/a
5 players	Ours	0.76	0.62	1	0.16	**1.97** $\times$**10**$^{-3}$	**5.64** $\times$**10**$^{-39}$
	KKT-con	0.98	0.76	0.32	0.32	**1.16** $\times$**10**$^{-4}$	**5.64** $\times$**10**$^{-39}$
	Heuristic	0.25	0.40	**0.02**	**0.01**	**2.56** $\times$**10**$^{-34}$	**5.64** $\times$**10**$^{-39}$
	MPC	0.08	0.79	**6.88** $\times$**10**$^{-14}$	**1.25** $\times$**10**$^{-12}$	**1.54** $\times$**10**$^{-26}$	n/a
7 players	Ours	0.84	0.76	0.32	0.32	**7.59** $\times$**10**$^{-9}$	**5.64** $\times$**10**$^{-39}$
	Ours (without inequalities)	0.45	0.57	**0.02**	0.32	**1.01** $\times$**10**$^{-8}$	**5.64** $\times$**10**$^{-39}$
	KKT-con	0.51	0.95	**0.01**	**0.01**	**1.29** $\times$**10**$^{-11}$	**5.64** $\times$**10**$^{-39}$
	Heuristic	**0.04**	0.50	**4.03** $\times$**10**$^{-3}$	0.16	**2.56** $\times$**10**$^{-34}$	**5.64** $\times$**10**$^{-39}$
	MPC	**2.85** $\times$**10**$^{-5}$	0.17	**1.57** $\times$**10**$^{-13}$	**9.42** $\times$**10**$^{-19}$	**2.70** $\times$**10**$^{-33}$	n/a

References

[1] Santokh Singh. *Critical reasons for crashes investigated in the national motor vehicle crash causation survey*. Tech. rep. 2015.

[2] Raphael E Stern et al. “Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments”. In: *Transportation Research Part C: Emerging Technologies* 89 (2018), pp. 205–221.

[3] Phil Lasley. *2019 Urban mobility report*. Tech. rep. 2019.

[4] Ohio University. *The future of driving*. 2021.

[5] Jusejuju. Link: `https://commons.wikimedia.org/w/index.php?curid=71233620`. No modification to the image. License: `https://creativecommons.org/licenses/by-sa/4.0/?ref=openverse`. 2018.

[6] Mykel J Kochenderfer. *Decision making under uncertainty: theory and application*. MIT press, 2015.

[7] Brian D Ziebart et al. “Planning-based prediction for pedestrians”. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2009, pp. 3931–3936.

[8] Peter Trautman and Andreas Krause. “Unfreezing the robot: Navigation in dense, interacting crowds”. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2010, pp. 797–803.

[9] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. “Congested traffic states in empirical observations and microscopic simulations”. In: *Physical review E* 62.2 (2000), p. 1805.

[10] Edward Schmerling et al. “Multimodal probabilistic model-based planning for human-robot interaction”. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. 2018.

[11] Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. 2. Society for Industrial and Applied Mathematics (SIAM), 1999.

[12] David Fridovich-Keil et al. “Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games”. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. 2020, pp. 1475–1481.

[13] Le Cleac’h, Mac Schwager, Zachary Manchester, et al. “ALGAMES: a fast augmented Lagrangian solver for constrained dynamic games”. In: *Autonomous Robots* 46.1 (2022), pp. 201–215.

[14] Forrest Laine et al. “The computation of approximate generalized feedback nash equilibria”. In: *Society for Industrial and Applied Mathematics (SIAM)* 33.1 (2023), pp. 294–318.

[15] Edward L Zhu and Francesco Borrelli. “A sequential quadratic programming approach to the solution of open-loop generalized nash equilibria”. In: *arXiv preprint arXiv:2203.16478* (2022).

[16] J. Rawlings and D. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2008.

[17] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *Proc. of the Int. Conf. on Learning Representations (ICLR)*. 2014.

[18] Carl Doersch. “Tutorial on variational autoencoders”. In: *arXiv preprint arXiv:1606.05908* (2016).

[19] Roger B Myerson. *Game theory: analysis of conflict*. Harvard university press, 1997.

[20] Alessandro Zanardi et al. *Game Theoretical Motion Planning: Tutorial ICRA 2021*.

[21] Francisco Facchinei and Christian Kanzow. “Generalized Nash equilibrium problems”. In: *Annals of Operations Research* 175.1 (2010), pp. 177–211.

[22] Dimitri Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[23] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[24] Emanuel Todorov and Weiwei Li. “A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems”. In: *Proc. of the IEEE American Control Conference (ACC)*. IEEE. 2005, pp. 300–306.

[25] Jonas Koenemann et al. “Whole-body model-predictive control applied to the HRP-2 humanoid”. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 3346–3351.

[26] Negar Mehr et al. “Maximum-Entropy Multi-Agent Dynamic Games: Forward and Inverse Solutions”. In: *IEEE Trans. on Robotics (TRO)* 39.3 (2023), pp. 1801–1815. DOI: 10.1109/TRO.2022.3232300.

[27] Jingqi Li et al. “Cost Inference for Feedback Dynamic Games from Noisy Partial State Observations and Incomplete Trajectories”. In: *arXiv preprint arXiv:2301.01398* (2023).

[28] Grady Williams et al. “Best response model predictive control for agile interactions between autonomous ground vehicles”. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE. 2018, pp. 2403–2410.

[29] Zijian Wang, Riccardo Spica, and Mac Schwager. “Game theoretic motion planning for multi-robot racing”. In: *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 225–238.

[30] Mingyu Wang et al. “Game-theoretic planning for self-driving cars in multivehicle competitive scenarios”. In: *IEEE Trans. on Robotics (TRO)* 37.4 (2021), pp. 1313–1325.

[31] Riccardo Spica et al. “A real-time game theoretic planner for autonomous two-player drone racing”. In: *IEEE Transactions on Robotics* 36.5 (2020), pp. 1389–1403.

[32] Wilko Schwarting et al. “Social behavior for autonomous vehicles”. In: *Proceedings of the National Academy of Sciences* 116.50 (2019), pp. 24972–24978.

[33] Alexander Liniger and John Lygeros. “A noncooperative game approach to autonomous racing”. In: *IEEE Trans. on Control Systems Technology (TCST)* 28.3 (2019), pp. 884–897.

[34] Lasse Peters et al. “Learning Mixed Strategies in Trajectory Games”. In: *Proc. of Robotics: Science and Systems (RSS)*. 2022. URL: https://arxiv.org/abs/2205.00291.

[35] Simo Särkkä. *Bayesian filtering and smoothing*. Cambridge university press, 2013.

[36] Lasse Peters. “Accommodating intention uncertainty in general-sum games for human-robot interaction”. Master’s thesis, Hamburg University of Technology, 2020.

[37] Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. "LUCIDGames: Online unscented inverse dynamic games for adaptive trajectory prediction and planning". In: *IEEE Robotics and Automation Letters (RA-L)* 6.3 (2021), pp. 5485–5492.

[38] Chaitanya Awasthi. "Forward and inverse methods in optimal control and dynamic game theory". Master's thesis, University of Minnesota, 2019.

[39] Peter Englert, Ngo Anh Vien, and Marc Toussaint. "Inverse KKT: Learning cost functions of manipulation tasks from demonstrations". In: *Intl. Journal of Robotics Research (IJRR)* 36.13-14 (2017), pp. 1474–1488.

[40] Sergey Levine and Vladlen Koltun. "Continuous inverse optimal control with locally optimal examples". In: *Proc. of the Int. Conf. on Machine Learning (ICML)*. 2012, pp. 475–482.

[41] Marcel Menner and Melanie N Zeilinger. "Maximum likelihood methods for inverse learning of optimal controllers". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 5266–5272.

[42] Arezou Keshavarz, Yang Wang, and Stephen Boyd. "Imputing a convex objective function". In: *Proc. of the Intl. Symp. on Intelligent Control (ISIC)*. 2011, pp. 613–619. DOI: 10.1109/ISIC.2011.6045410.

[43] Simon Rothfuß et al. "Inverse Optimal Control for Identification in Non-Cooperative Differential Games". In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 14909–14915. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.2538. URL: https://www.sciencedirect.com/science/article/pii/S2405896317334602.

[44] Chaitanya Awasthi and Andrew Lamperski. "Inverse differential games with mixed inequality constraints". In: *Proc. of the IEEE American Control Conference (ACC)*. 2020, pp. 2182–2187.

[45] Mihai Anitescu. "On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints". In: *SIAM Journal on Optimization* 15.4 (2005), pp. 1203–1236.

[46] L Peters et al. "Inferring Objectives in Continuous Dynamic Games from Noise-Corrupted Partial State Observations". In: *Proc. of Robotics: Science and Systems (RSS)*. 2021.

[47] Michael Everett, Yu Fan Chen, and Jonathan P How. "Motion planning among dynamic, decision-making agents with deep reinforcement learning". In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018.

[48] Bruno Brito et al. "Where to go next: learning a subgoal recommendation policy for navigation in dynamic environments". In: *IEEE Robotics and Automation Letters (RA-L)* 6.3 (2021), pp. 4616–4623.

[49] Bruno Brito, Achin Agarwal, and Javier Alonso-Mora. "Learning Interaction-Aware Guidance for Trajectory Optimization in Dense Traffic Scenarios". In: *IEEE Trans. on Intelligent Transportation Systems (ITS)* 23.10 (2022), pp. 18808–18821.

[50] Varun Tolani et al. "Visual navigation among humans with optimal control as a supervisor". In: *IEEE Robotics and Automation Letters (RA-L)* 6.2 (2021), pp. 2288–2295.

[51] Henrik Kretzschmar et al. "Socially compliant mobile robot navigation via inverse reinforcement learning". In: *Intl. Journal of Robotics Research (IJRR)* 35.11 (2016), pp. 1289–1307.

[52] Nicholas Rhinehart et al. "Precog: Prediction conditioned on goals in visual multi-agent settings". In: *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*. 2019.

[53] Junha Roh et al. “Multimodal trajectory prediction via topological invariance for navigation at uncontrolled intersections”. In: *Proc. of the Conf. on Robot Learning (CoRL)*. 2021.

[54] Muchen Sun et al. “Move beyond trajectories: Distribution space coupling for crowd navigation”. In: *Proc. of Robotics: Science and Systems (RSS)* (2021).

[55] Christoph Schöller et al. “What the constant velocity model can teach us about pedestrian motion prediction”. In: *IEEE Robotics and Automation Letters (RA-L)* 5.2 (2020), pp. 1696–1703.

[56] Keenon Werling et al. “Fast and feature-complete differentiable physics for articulated rigid bodies with contact”. In: *Proc. of Robotics: Science and Systems (RSS)* (2021).

[57] Taylor A Howell et al. “Dojo: A differentiable simulator for robotics”. In: *arXiv preprint arXiv:2203.00806* (2022).

[58] Hiroharu Kato et al. “Differentiable rendering: A survey”. In: *arXiv preprint arXiv:2006.12057* (2020).

[59] Brandon Amos and J Zico Kolter. “Optnet: Differentiable optimization as a layer in neural networks”. In: *Proc. of the Int. Conf. on Machine Learning (ICML)*. PMLR. 2017, pp. 136–145.

[60] Daniel Ralph and Stephan Dempe. “Directional derivatives of the solution of a parametric nonlinear program”. In: *Mathematical Programming* 70.1-3 (1995), pp. 159–172.

[61] Philipp Geiger and Christoph-Nikolas Straehle. “Learning game-theoretic models of multiagent trajectories using implicit layers”. In: *Proc. of the Conference on Advancements of Artificial Intelligence (AAAI)*. Vol. 35. 6. 2021, pp. 4950–4958.

[62] Forrest Laine et al. “Multi-hypothesis interactions in game-theoretic motion planning”. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE. 2021, pp. 8016–8023.

[63] Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

[64] Stephen C Billups, Steven P Dirkse, and Michael C Ferris. “A comparison of large scale mixed complementarity problem solvers”. In: *Computational Optimization and Applications* 7.1 (1997), pp. 3–25.

[65] Ankur A Kulkarni and Uday V Shanbhag. “On the variational equilibrium as a refinement of the generalized Nash equilibrium”. In: *Automatica* 48.1 (2012), pp. 45–55.

[66] Francisco Facchinei and Jong-Shi Pang. “Finite-Dimensional Variational Inequalities and Complementarity Problems”. In: Springer Verlag, 2003.

[67] Steven P Dirkse and Michael C Ferris. “The PATH solver: A nommonotone stabilization scheme for mixed complementarity problems”. In: *Optimization methods and software* 5.2 (1995), pp. 123–156.

[68] Jeff Bezanson et al. “Julia: A fresh approach to numerical computing”. In: *SIAM Review (SIREV)* 59.1 (2017), pp. 65–98. URL: https://doi.org/10.1137/141000671.

[69] Shashi Gowda et al. “High-performance symbolic-numerics via multiple dispatch”. In: *ACM Communications in Computer Algebra* 55.3 (2021).

[70] J. Revels, M. Lubin, and T. Papamarkou. “Forward-Mode Automatic Differentiation in Julia”. In: *arXiv:1607.07892 [cs.MS]* (2016). URL: https://arxiv.org/abs/1607.07892.

[71] Michael Innes. “Don’t unroll adjoint: Differentiating ssa-form programs”. In: *arXiv preprint arXiv:1810.07951* (2018).

[72] Michael Innes et al. “Fashionable Modelling with Flux”. In: *CoRR* abs/1811.01457 (2018). arXiv: 1811.01457. URL: https://arxiv.org/abs/1811.01457.

[73] Mike Innes. “Flux: Elegant Machine Learning with Julia”. In: *Journal of Open Source Software* (2018). DOI: 10.21105/joss.00602.

[74] Stanford Artificial Intelligence Laboratory et al. *Robotic Operating System*. Version ROS Melodic Morenia. May 23, 2018. URL: https://www.ros.org.

[75] Raunak P Bhattacharyya et al. “Online parameter estimation for human driver behavior prediction”. In: *Proc. of the IEEE American Control Conference (ACC)*. 2020.

[76] Yutaka Kanayama et al. “A stable tracking control method for an autonomous mobile robot”. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*. 1990.

[77] R. Duncan Luce. *Individual Choice Behavior*. Oxford, England: John Wiley, 1959. URL: https://psycnet.apa.org/record/1960-03588-000.

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

[79] Lasse Peters et al. “Contingency Games for Multi-Agent Interaction”. In: *arXiv preprint arXiv:2304.05483* (2023).

[80] Haimin Hu et al. “Emergent Coordination through Game-Induced Nonlinear Opinion Dynamics”. In: *arXiv preprint arXiv:2304.02687* (2023).

[81] Mikko Lauri, David Hsu, and Joni Pajarinen. “Partially Observable Markov Decision Processes in Robotics: A Survey”. In: *IEEE Trans. on Robotics (TRO)* (2022).

[82] Haimin Hu and Jaime F Fisac. “Active uncertainty reduction for human-robot interaction: An implicit dual control approach”. In: *Intl. Workshop on the Algorithmic Foundations of Robotics (WAFR)*. Springer. 2022, pp. 385–401.

[83] Haimin Hu, Kensuke Nakamura, and Jaime F Fisac. “SHARP: Shielding-aware robust planning for safe and efficient human-robot interaction”. In: *IEEE Robotics and Automation Letters (RA-L)* 7.2 (2022), pp. 5591–5598.

[84] Jaime F Fisac et al. “Probabilistically safe robot planning with confidence-based human predictions”. In: *Proc. of Robotics: Science and Systems (RSS)* (2018).

[85] David Fridovich-Keil et al. “Confidence-aware motion prediction for real-time collision avoidance”. In: *Intl. Journal of Robotics Research (IJRR)* 39.2-3 (2020), pp. 250–265.

[86] Andrea Bajcsy et al. “A robust control framework for human motion prediction”. In: *IEEE Robotics and Automation Letters (RA-L)* 6.1 (2020), pp. 24–31.

[87] Daniel S Brown and Scott Niekum. “Deep bayesian reward learning from preferences”. In: *arXiv preprint arXiv:1912.04472* (2019).

[88] Jaedeug Choi and Kee-Eung Kim. “Map inference for bayesian inverse reinforcement learning”. In: *Proc. of the Advances in Neural Information Processing Systems (NIPS)* 24 (2011).

[89] Alex J Chan and Mihaela van der Schaar. “Scalable bayesian inverse reinforcement learning”. In: *Proc. of the Int. Conf. on Learning Representations (ICLR)* (2021).

[90] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational inference: A review for statisticians”. In: *Journal of the American Statistical Association (JASA)* 112.518 (2017), pp. 859–877.

[91] Matthew D Hoffman et al. “Stochastic variational inference”. In: *Journal on Machine Learning Research (JMLR)* (2013).

[92] Jingqi Li et al. “Scenario-Game ADMM: A Parallelized Scenario-Based Solver for Stochastic Noncooperative Games”. In: *arXiv preprint arXiv:2304.01945* (2023).

[93] Oscar de Groot et al. “Scenario-based trajectory optimization in uncertain dynamic environments”. In: *IEEE Robotics and Automation Letters (RA-L)* 6.3 (2021), pp. 5389–5396.

[94] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL: `http://probml.github.io/book2`.

[95] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL: `probml.ai`.

[96] Diederik Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *Proc. of the Int. Conf. on Learning Representations (ICLR)*. San Diega, CA, USA, 2015.

[97] L Sun, X Jia, and A Dragan. “On Complementing End-To-End Human Behavior Predictors with Planning.” In: *Proc. of Robotics: Science and Systems (RSS)*. 2021.

[98] Lars Lindemann and Dimos V Dimarogonas. “Robust motion planning employing signal temporal logic”. In: *Proc. of the IEEE American Control Conference (ACC)*. IEEE. 2017, pp. 2950–2955.

[99] Somil Bansal et al. “Hamilton-jacobi reachability: A brief overview and recent advances”. In: *Proc. of the Conference on Decision Making and Control (CDC)*. IEEE. 2017, pp. 2242–2253.

[100] Katerina Fragkiadaki et al. “Recurrent network models for human dynamics”. In: *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*. 2015, pp. 4346–4354.

[101] Ashesh Jain et al. “Structural-RNN: Deep learning on spatio-temporal graphs”. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5308–5317.

[102] Wilko Schwarting et al. “Stochastic dynamic games in belief space”. In: *IEEE Transactions on Robotics* 37.6 (2021), pp. 2157–2172.

[103] Xinjie Liu, Lasse Peters, and Javier Alonso-Mora. “Learning to Play Trajectory Games Against Opponents With Unknown Objectives”. In: *IEEE Robotics and Automation Letters (RA-L)* 8.7 (2023). Copyright © 2023, IEEE. Reprinted, with permission, pp. 4139–4146. DOI: `10.1109/LRA.2023.3280809`.