Constraint Learning in
Multi-Agent Dynamic Games from Demonstrations of Local Nash Interactions

Zhouyu Zhang[1], Chih-Yuan Chiu[1], and Glen Chou[1]

May 14, 2025

***Abstract*— To ensure long-term safety and reliability, autonomous agents sharing the same environment must be able to infer each other's constraints. However, existing methods cannot recover constraints that depend upon the states of multiple interacting agents, such as collision avoidance. To address this gap, we use dynamic game theory and inverse optimal control (IOC) to learn parametric constraints from a given dataset of local Nash interactions between multiple agents. Specifically, we introduce mixed-integer linear programs (MILP) encoding the Karush-Kuhn-Tucker (KKT) conditions of each interacting agent, which recover constraints that are consistent with the Nash stationarity of the interaction demonstrations. We also demonstrate that the interaction constraints recovered by our method can be used to design motion plans which robustly satisfy the underlying constraints. Finally, we illustrate via numerical simulation that our method can successfully recover constraints from interaction demonstrations of multiple agents with spherical and polytope-shaped collision-avoidance constraints, and can additionally design robust, interactive motion plans for the aforementioned agents.**

I. Introduction

Learning from demonstrations (LfD) is a powerful paradigm for enabling robots to learn constraints in their workspace [1–5]. In particular, [1–3] recast constraint inference as the problem of solving for the constraint parameters that best explains a set of approximately optimal demonstration trajectories. However, existing constraint inference methods are typically designed for robots operating in isolation, and do not account for interactions between robots and surrounding strategic agents. These methods thus cannot infer coupled constraints that simultaneously depend on the states or control inputs of multiple agents, such as collision avoidance.

To address this key limitation, we use tools from dynamic game theory and inverse optimal control (IOC) to learn constraints from the demonstrations of *interactions* between multiple strategic agents. Concretely, we recover unknown constraint parameters by posing an inverse optimization problem that uses Nash equilibrium constraints to encode steady-state agent interactions. We also show that the inferred multi-agent constraints can be used for robust motion planning. Although IOC and dynamic games have been previously applied to enable *cost* inference in multi-agent settings [6–9], to the best of our knowledge, our work is the first to formulate a game-theoretic algorithm for multi-agent *constraint* inference.

Specifically, our main contributions are threefold:

1) We formulate a feasibility problem for learning parameterized constraints from demonstrations of multi-agent interactions. Our method generalizes the inverse optimal control-based constraint learning method in [1] to the multi-agent setting.
2) We use our multi-agent constraint learning framework to design robust motion plans through implicit constraint verification via Model Predictive Path Integral (MPPI) control.
3) We evaluate our method in settings where multiple agents interact while adhering to inter-agent distance or polyhedral collision avoidance constraints. Our numerical results confirm that our inverse learning and MPPI-based planning algorithms can successfully recover *a priori* unknown constraint parameters and generate robust motion plans, respectively[1].

II. Related Work

A. Constraint Learning via Inverse Optimal Control (IOC)

LfD via IOC has been applied to empower robots to learn new tasks [10, 11], deduce the intent of other strategic agents [6–8], and infer static constraints in their environment [1–3]. In particular, [11] proposes an IOC-based method to infer convex constraints from a finite set of possible costs and constraints, Meanwhile, [12–14] use a single demonstration to infer local constraints on the trajectory level. Our method is most closely aligned with the work of Chou et al. [1], which recovers unknown constraint parameters by enforcing the KKT conditions of a given set of locally optimal demonstrations. However, the methods mentioned above focus only on single-agent scenarios with decoupled objectives and feasible sets. In contrast, we leverage demonstrations of the strategic interactions of multiple agents to infer constraints that are coupled across agent states and controls.

B. Dynamic Games for Motion Planning and Cost Inference

Motion planning and intent inference in multi-agent scenarios are naturally posed as dynamic games [15], which provide a powerful theoretical framework for reasoning about strategic multi-agent robotic interactions. In particular, [8, 16–19]present computationally tractable algorithms for approximating solving general-sum *forward dynamic games*, thus enabling efficient motion planning for interactive multi-agent scenarios. Meanwhile, [6–9, 20] developed *inverse dynamic game* algorithms to efficiently infer the unknown costs of strategic agents operating in a shared environment, by leveraging the KKT conditions that encode the Nash stationarity of a given set of interaction demonstrations. However, to our knowledge, existing inverse game-theoretic methods are focused on identifying the unknown *costs* of strategic agents, and are not directly applicable to *constraint* inference tasks.

III. Forward Dynamic Game Formulation

We introduce *forward* dynamic games, which model the steady-state motion plans of N strategic agents interacting in a shared environment. Then, in Section IV we present an *inverse* dynamic game framework for inferring the constraints of strategic agents based on a demonstration dataset of their observed interactions.

Consider an N-agent, T-stage discrete-time *forward* dynamic game $\mathcal{G}$, in which $x_t^i \in \mathbb{R}^{n_i}$ represents the state vector of each agent $i \in [N] := \{1,\cdots,N\}$ at each time $t \in [T] := \{1,\cdots,T\}$. We denote by $x_t := (x_t^1,\cdots,x_t^N) \in \mathbb{R}^n$ the *system state* at each time $t \in [T]$, where $n := \sum_{i\in[N]} n_i$. Similarly, we concatenate the agent controls to form the overall control vector at each time $t \in [T]$, denoted $u_t := (u_t^1,\cdots,u_t^N) \in \mathbb{R}^m$, where $m := \sum_{i\in[N]} m_i$. Finally, we define $\xi := (x,u) \in \mathbb{R}^{(n+m)T}$ to be the state-control system trajectory.

[1] Georgia Tech. {zzhang3097, cyc, chou}@gatech.edu

[1]Our poster contains some experiment results not included in this manuscript.

Each agent $i \in [N]$ aims to minimize its cost $J^i(\xi)$, whose value depends on the system trajectory ξ. Moreover, each agent $i \in [N]$ is associated with a finite set of equality constraints $C^{\text{eq},i} := \{h^i_j(\xi) = 0 : j \in [N^{\text{eq},i}]\}$ and inequality constraints $C^{\text{ineq},i} := \{g^i_j(\xi) \leq 0 : j \in [N^{\text{ineq},i}]\}$[2]. The constraints $h^i_j(\cdot)$ and $g^i_j(\cdot)$ for each agent $i \in [N]$ can encode the system dynamics $x^i_{t+1} = f_t(x^i_t, u^i_t)$ obeyed by the states and controls of each agent, as well as obstacle avoidance and inter-agent collision avoidance constraints.

The objective of the *forward* dynamic game is to compute the steady-state system trajectory by solving the following coupled optimization problems for each player $i \in [N]$:

$$\min_{x^i, u^i} \quad J^i(\xi) \tag{1a}$$

$$\text{s.t.} \quad h^i_j(\xi) = 0, \forall\, j \in [N^{\text{eq},i}], \tag{1b}$$

$$g^i_j(\xi) \leq 0, \forall\, j \in [N^{\text{ineq},i}]. \tag{1c}$$

We call $u^\star$ a *Nash equilibrium solution* to (1) with trajectory $x^\star$ if $\xi^\star := (x^\star, u^\star)$ satisfies (1). We call $u^\star$ a *local Nash equilibrium solution* to (1) with corresponding trajectory $x^\star$ if the state-control trajectory $\xi^\star := (x^\star, u^\star)$ satisfies the following condition: There exists a neighborhood $N(\xi^\star)$ of $\xi^\star$ such that for any agent $i \in [N]$ and any feasible trajectory $\{(x^i_t, u^i_t) : t \in [T]\}$ of agent i, we have $J^i(\xi^\star) = J^i(x^\star_t, u^\star_t) \leq J^i(x^{-i\star}_t, u^{-i\star}_t, x^i_t, u^i_t)$. Here, the notation $-i$ refers to all agent indices in $[N]$ apart from i.

IV. Multi-Agent Constraint Inference and Motion Planning

A. Inverse Games for Multi-Agent Constraint Inference

In contrast to the forward dynamic game described above, the *multi-agent constraint inference* problem concerns the deduction of unknown agent constraints from a given dataset of observed agent interactions. Specifically, suppose the inequality constraint set $C^{\text{ineq},i}$ for each agent $i \in [N]$ is partitioned into a set of *known* inequality constraints $C^{\text{ineq},i}_k := \{g^i_{j,k}(\xi) = 0 : j \in [N^{\text{ineq},i}_k]\}$ and *unknown*, parameterized inequality constraints $C^{\text{ineq},i}_{\neg k} := \{g^i_{j,\neg k}(\xi, \theta) = 0 : j \in [N^{\text{ineq},i}_{\neg k}]\}$, where $N^{\text{ineq},i}_k + N^{\text{ineq},i}_{\neg k} = N^{\text{ineq},i}$, and θ denotes an unknown parameter [3]. We aim to infer the unknown parameter θ from a given demonstration set of D local Nash equilibrium trajectories $\mathcal{D} := \{\xi^{\text{loc}}_d : d \in [D]\}$ satisfying the KKT conditions corresponding to the forward dynamic game (1)[4]. In other words, for each demonstration trajectory ξ^{loc}_d in $\mathcal{D}$ and each agent $i \in [N]$, there exist Lagrange multipliers $\boldsymbol{\lambda}^{i,d}_k := \{\lambda^{i,d}_{j,k} : j \in [N^{\text{ineq},i}_k]\}$, $\boldsymbol{\lambda}^{i,d}_{\neg k} := \{\lambda^{i,d}_{j,\neg k} : j \in [N^{\text{ineq},i}_{\neg k}]\}$, and $\boldsymbol{\nu}^{i,d}, := \{\nu^{i,d}_j : j \in [N^{\text{eq},i}]\}$ satisfying:

$$h^i_j(\xi^{\text{loc}}_d) = 0, \forall\, j \in [N^{\text{eq},i}] \tag{2a}$$

$$g^i_{j,k}(\xi^{\text{loc}}_d) \leq 0, \forall\, j \in [N^{\text{eq},i}] \tag{2b}$$

$$g^i_{j,\neg k}(\xi^{\text{loc}}_d, \theta) \leq 0, \forall\, j \in [N^{\text{eq},i}], \tag{2c}$$

$$\lambda^{i,d}_{j,k} \geq 0, \forall\, j \in [N^{\text{ineq},i}_k], \tag{2d}$$

$$\lambda^{i,d}_{j,\neg k} \geq 0, \forall\, j \in [N^{\text{ineq},i}_{\neg k}], \tag{2e}$$

$$\lambda^{i,d}_{j,k} \odot g^i_{j,k}(\xi^{\text{loc}}_d) = 0, \forall\, j \in [N^{\text{ineq},i}_k], \tag{2f}$$

$$\lambda^{i,d}_{j,\neg k} \odot g^i_{j,\neg k}(\xi^{\text{loc}}_d, \theta) = 0, \forall\, j \in [N^{\text{ineq},i}_k], \tag{2g}$$

$$\begin{aligned} \nabla_\xi J^i(\xi^{\text{loc}}_d) &+ \sum_{j \in [N^{\text{ineq},i}_k]} (\lambda^{i,d}_{j,k})^\top \nabla_\xi g^i_{j,k}(\xi^{\text{loc}}_d) \\ &+ \sum_{j \in [N^{\text{ineq},i}_{\neg k}]} (\lambda^{i,d}_{j,\neg k})^\top \nabla_\xi g^i_{j,\neg k}(\xi^{\text{loc}}_d, \theta) \\ &+ \sum_{j \in [N^{\text{eq},i}]} (\nu^{i,d}_j)^\top \nabla_\xi h^i_j(\xi^{\text{loc}}_d) = 0. \end{aligned} \tag{2h}$$

where $\odot$ denotes elementwise multiplication. Above, (2a)-(2c) encode primal feasibility, (2d)-(2e) encode the non-negativity of the Lagrange multipliers corresponding to inequality constraints, (2f)-(2g) encode complementary slackness, while (2h) encodes first-order stationarity. We collectively denote (2) by $\text{KKT}^i(\xi^{\text{loc}}_d)$.

Thus, given the set $\mathcal{D}$ of locally optimal demonstrations, there must exist Lagrange multiplier values $\boldsymbol{\lambda}^{i,d}_k, \boldsymbol{\lambda}^{i,d}_{\neg k}, \boldsymbol{\nu}^{i,d}$ such that the unknown constraint parameter θ, together with $\boldsymbol{\lambda}^{i,d}_k, \boldsymbol{\lambda}^{i,d}_{\neg k}, \boldsymbol{\nu}^{i,d}$, solves the following feasibility problem:

$$\text{find} \quad \theta, \boldsymbol{\lambda}^{i,d}_k, \boldsymbol{\lambda}^{i,d}_{\neg k}, \boldsymbol{\nu}^{i,d}, \quad i \in [N], d \in [D] \tag{3a}$$

$$\text{s.t.} \quad \text{KKT}^i(\xi^{\text{loc}}_d), \quad i \in [N], d \in [D]. \tag{3b}$$

In the event that the demonstration data is not perfectly optimal, we can relax (3) by replacing the stationarity constraints (2h) with a cost term given by the norm squared of the stationarity terms $\{\text{stat}^i(\xi^{\text{loc}}_d) : d \in [D], i \in [N]\}$, given by the left-hand side of (2h). Then, to recover the unknown constraint parameter θ from approximately locally-optimal interaction demonstrations, we solve the following optimization program:

$$\min_{\theta, \boldsymbol{\lambda}^{i,d}_k, \boldsymbol{\lambda}^{i,d}_{\neg k}, \boldsymbol{\nu}^{i,d}} \quad \sum_{d \in [D]} \sum_{i \in [N]} \|\text{stat}^i(\xi^{\text{loc}}_d)\|_2^2, \tag{4a}$$

$$\text{s.t.} \quad (2a)-(2g), \quad i \in [N], d \in [D]. \tag{4b}$$

As shown in [3], Sec. 4, when the constraints $h^i_j(\cdot)$, $g^i_{j,k}(\cdot)$, and $g^i_{j,\neg k}(\cdot)$ encode spherical or polytopic constraint sets, (4a) can be written as a mixed-integer linear program (MILP) or mixed-integer bilinear program (MIBLP), respectively, and solved via off-the-shelf solvers like Gurobi [21].

B. Robust Motion Planning via Implicit Constraint Checking

Since multiple candidate values of the constraint parameter θ may in general be consistent with the demonstration dataset $\mathcal{D}$, the KKT conditions (2) only pose *necessary, but not sufficient* criteria that the true value of θ must satisfy. To design robust motion plans despite this uncertainty over θ, we perform robust constraint checking to ensure that each designed, candidate trajectory is marked safe by a θ value consistent with (2). To this end, we present an *implicit* approach for constraint checking via Model Predictive Path Integral (MPPI) control [22, 23].

Specifically, given a nominal state-control trajectory $\xi_{\text{nom}} := (x_{\text{nom}}, u_{\text{nom}})$, we enforce constraint satisfaction by iteratively applying Algorithm 1 to update the trajectory ξ_{nom} until convergence. In each iterate of Algorithm 1, we first draw M sample controls $\{u^{(s)} : s \in [M]\}$ i.i.d. from a Gaussian distribution centered on u_{nom}, which we then unroll using the dynamics f to generate the sample state-control trajectories $\{\xi^{(s)} : s \in [M]\}$. Next, for each $\xi^{(s)}$, we wish to determine whether there exists a feasible constraint parameter value θ which is consistent with the demonstrated interactions $\mathcal{D}$, but not with $\xi^{(s)}$. In other words, we wish to solve the feasibility problem (3), augmented with the stipulation that the sample trajectory $\xi^{(s)}$ *violates* the unknown primal inequality constraints $\bigwedge_{i \in [N], j \in [N^{\text{ineq},i}_{\neg k}]} \{g^i_{j,\neg k}(\xi^{(s)}, \theta) \leq 0\}$:

$$\text{find} \quad \theta, \boldsymbol{\lambda}^{i,d}_k, \boldsymbol{\lambda}^{i,d}_{\neg k}, \boldsymbol{\nu}^{i,d}, \quad i \in [N], d \in [D] \tag{5a}$$

[2] Our methods readily extend to the case where each agent's constraint set is the *union of intersections* of equality and inequality constraints. For simplicity, our formulation here considers only *intersections*.

[3] We assume without loss of generality that all unknown constraints are *in*equality constraints, since each equality constraint $h^i_j(\xi) = 0$ can be expressed using two inequality constraints: $h^i_j(\xi) \leq 0$ and $-h^i_j(\xi) \leq 0$

[4] Our method can be refined to learn constraints from *approximate* Nash equilibrium demonstrations; see (4) and the accompanying discussion.

$$\text{s.t.} \quad \text{KKT}^i(\xi_d^{\text{loc}}), \qquad i \in [N], d \in [D], \tag{5b}$$

$$\bigvee_{i \in [N], j \in [N^{\text{ineq},i}_{\neg k}]} \left[g^i_{j,\neg k}(\xi^{(s)}, \theta) > 0 \right]. \tag{5c}$$

If (5) returns feasible, there exists some constraint parameter value $\theta^{(s)}$ consistent with the demonstrations $\mathcal{D}$ such that $\xi^{(s)}$ violates the inequality constraints $\bigwedge_{i,j} \{g^i_{j,\neg k}(\xi^{(s)}, \theta^{(s)}) \leq 0\}$. In this case, we compute the total constraint violation for each agent i, denoted c^i_{cv} below, as follows:

$$c^i_{\text{cv}}(\xi^{(s)}, \theta^{(s)}) := \sum_{j \in [N^{\text{ineq},i}_{\neg k}]} \max\{g^i_{j,\neg k}(\xi^{(s)}, \theta^{(s)}), 0\}. \tag{6}$$

Otherwise, if (5) is infeasible, we set $c^i_{\text{cv}}(\xi^{(s)}) := 0$ for each $i \in [N]$. We then update the nominal control trajectory for each agent $i \in [N]$ by taking a weighted combination of the sample control trajectories $\{u^{(s)} : s \in [M]\}$, with weights depending on the corresponding cost value $J^i(\xi^{(s)})$ and degree of constraint violation $c_{\text{cv}}(\xi^{(s)})$:

$$\tilde{u}^i_{\text{nom}} := \frac{\sum_{s \in [M]} \exp(-J^i(\xi^{(s)}) - c_{\text{cv}}(\xi^{(s)}, \theta^{(s)})) u^{(s)}}{\sum_{s \in [M]} \exp(-J^i(\xi^{(s)}) - c_{\text{cv}}(\xi^{(s)}, \theta^{(s)}))}. \tag{7}$$

Finally, we unroll the updated nominal control trajectory $\{\tilde{u}^i_{\text{nom}} : i \in [N]\}$ to generate the updated nominal state-control trajectory $\tilde{\xi}^i$.

Algorithm 1: Inverse KKT-Guided MPPI control-based Sample Trajectory Update (1 Iterate)

Data: Nominal state-control trajectory ξ_{nom}, dynamics model f, demonstrations $\mathcal{D}$, number of samples M

```
{ξ^(s) : s ∈ [M]} ←
    N sample state-control trajectories generated by perturbing
    ξ_nom via a Gaussian distribution, and enforcing the dynamics f.
for s ∈ [M] do
    Solve augmented inverse KKT problem (5)
    if (5) is feasible then
        θ^(s) ← Feasible θ value from solving (5)
        c^i_cv(ξ^(s), θ^(s)) ← (6), ∀ i ∈ [N].
    else
        c^i_cv(ξ^(s), θ^(s)) ← 0
{ũ^i_nom : i ∈ [N]} ← (7), using c^i_cv(ξ^(s), θ^(s)), J^i(ξ^(s)), and u^(s).
{ξ̃^i_nom : i ∈ [N]} ← Unroll {ũ^i_nom : i ∈ [N]} using dynamics f
return Updated nominal state-control trajectory ξ̃_nom
```

V. Experiments

To evaluate our constraint learning and robust motion planning methods, we present a simulation study of collision avoidance scenarios with interacting agents whose constraint sets are either spherical (Section V-A) or polytope-shaped (Section V-B). All experiments are implemented with YALMIP [24] and Gurobi [21].

A. Spherical Collision Avoidance Constraints

Consider the setting in which $N = 3$ agents navigate in a shared 2D environment over the time horizon $T = 10$. The state of each agent $i \in [N]$ is given by $x^i_t := (p^i_{x,t}, p^i_{y,t}, v^i_{x,t}, v^i_{y,t}) \in \mathbb{R}^4$ for each $t \in [T]$, and the system state is given by $x_t := (x^1_t, \cdots, x^N_t)$. Each agent follows the double integrator dynamics discretized at intervals of $\Delta t = 1$s, and optimizes the following smoothness cost:

$$J^i = \sum_{t=1}^{T-1} \left[\|p^i_{x,t+1} - p^i_{x,t}\|_2^2 + \|p^i_{y,t+1} - p^i_{y,t}\|_2^2 \right] \tag{8}$$

while ensuring that their trajectory satisfies the following spherical collision avoidance constraints at each time $t \in [T]$, which are a priori unknown to the constraint learner:

$$g^i_{t,\neg k}(\xi, \theta^i) = -\|p^i_{x,t} - p^j_{x,t}\|_2^2 + (\theta^i)^2 \leq 0. \tag{9}$$

Above, θ^i denotes the radius of the spherical collision avoidance set for each agent $i \in [N]$. Finally, the trajectory of each agent $i \in [N]$ is constrained by a prescribed set of origin and goal positions, given by:

$$h^i_t(\xi) = \begin{bmatrix} p^i_0 - \bar{p}^i_o \\ p^i_T - \bar{p}^i_d \end{bmatrix} = 0, \tag{10}$$

where $p^i_t := (p^i_{x,t}, p^i_{y,t})$ for each $i \in [N]$ and $t \in [T]$, while $\bar{p}^i_o \in \mathbb{R}^2$ and $\bar{p}^i_d \in \mathbb{R}^2$ respectively denote the origin and destination positions for agent i, and are fixed at the following values:

$$\bar{p}^1_o = (0,0), \quad \bar{p}^2_o = (20,20), \quad \bar{p}^3_o = (0,20), \tag{11a}$$

$$\bar{p}^1_d = (20,20), \quad \bar{p}^2_d = (0,0), \quad \bar{p}^3_d = (20,0). \tag{11b}$$

a) Constraint Inference: We conduct two experiments to validate our constraint learning method, with ground truth constraint parameter values set to $(\theta^1, \theta^2, \theta^3) = (4, 5, 6)$ and $(\theta^1, \theta^2, \theta^3) = (6, 7, 8)$, respectively. For each experiment, we generate an interaction demonstration ξ that activates the constraints (9), by computing local Nash equilibrium solutions corresponding to the coupled optimization problem (1). Specifically, we solve the KKT conditions (2) using the costs and constraints given by (8)-(11). In Fig. 1, we plot ξ while emphasizing timesteps at which the inequality constraints (9) were active.

By solving (4) using the demonstration ξ, we successfully infer the correct values of θ^2 and θ^3 in each experiment. Our Gurobi solver converged in 90 ms. Note that since Agent 1's collision avoidance constraint, as parameterized by θ^1, is the least restrictive among the agents, it is not recoverable from the demonstration ξ.

b) MPPI-based Motion Planning: To evaluate our MPPI-based motion planning approach (as detailed in Sec. IV-B), we first generate a single demonstration trajectory using the costs and constraints given in (8)-(10), with origin and destination coordinates:

$$\bar{p}^1_o = (10,0), \quad \bar{p}^2_o = (0,10), \tag{12a}$$

$$\bar{p}^1_d = (0,10), \quad \bar{p}^2_d = (10,0). \tag{12b}$$

and ground truth constraint parameters $\theta = (\theta^1, \theta^2) = (5, 5)$. To design robust trajectories, we run Algorithm 1 for 70 iterations using $M = 16$ samples, with time horizon $T = 20$ and discretization time $\Delta t = 0.1$s. As illustrated in Figure 2, our method generates trajectories for each agent which satisfy the spherical inter-agent collision avoidance constraints (9) with radii $\theta^1 = \theta^2 = 5$. Our Gurobi solve time was 293.36 s.

B. Polytope Collision Avoidance Constraints

We also numerically evaluate our constraint inference method on the setting where the interacting agents have polytope-shaped, rather than spherical, collision avoidance constraints. Concretely, consider the interactions, over $T = 20$ time steps, of $N = 2$ agents with states, dynamics, costs (8), and origin and destination constraints (11) as formulated in Section V-A. We define:

$$A = \begin{bmatrix} -0.2545 & -0.9671 \\ 0.9487 & -0.3162 \\ 0.2169 & 0.9762 \\ -0.9285 & 0.3714 \end{bmatrix}, \quad b = \begin{bmatrix} 10.0779 \\ 9.4868 \\ 11.6058 \\ 9.4705 \end{bmatrix}. \tag{13}$$

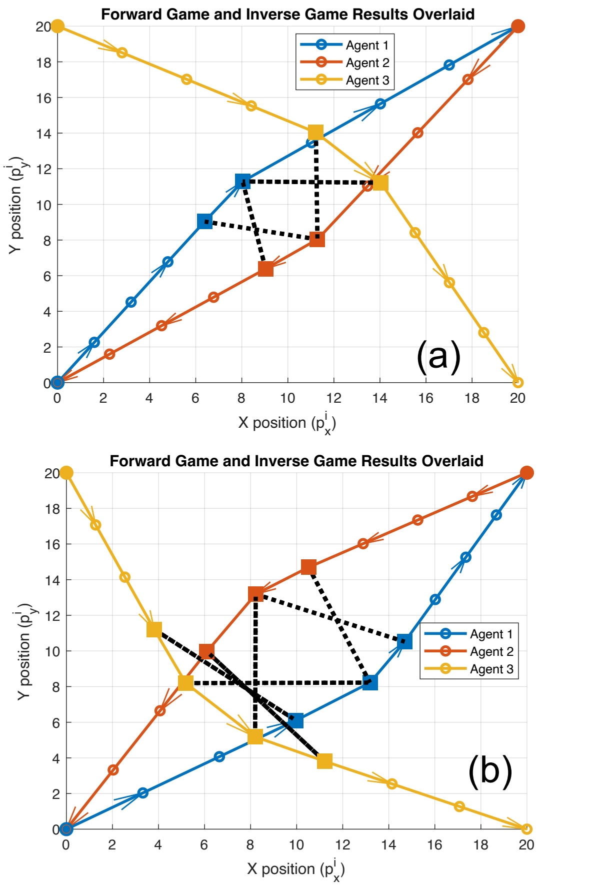

Fig. 1: Demonstrations (3 agents) for Sec. V-A. We indicate timesteps when the inter-agent collision constraints are active (resp., inactive) with filled rectangular (resp., unfilled circular) nodes. Dashed black lines connect pairs of agent states at which the constraints are active.

and constrain each agent's trajectory lies within the following constraint set, which is a priori unknown to the constraint learner:

$$C^{\text{ineq},i} = \bigwedge_{t=1}^{T} \bigvee_{r=1}^{4} \left\{ A_{r1}(p_{x,t}^{i} - p_{x,t}^{3-i}) + A_{r2}(p_{y,t}^{i} - p_{y,t}^{3-i}) \leq b_r \right\}, \quad (14)$$

To validate our constraint learning method, we generate two local Nash equilibrium trajectories $\xi^{(1)}, \xi^{(2)}$, which we then use to solve solve (4). As shown in Fig. 3, our method successfully recovers the exact inter-agent polytope constraint (14). Our Gurobi solve time was 55.93 s.

VI. Conclusion and Future Work

Using tools from inverse optimal control and dynamic game theory, we formulated a constraint learning framework that can infer the coupled constraints of a set of strategic agents from demonstrations of their equilibrium interactions. Moreover, we show the applicability of our constraint learning method for robust motion planning. Across numerical studies of multiple interacting agents with spherical or polytope-shaped collision avoidance constraints, we show that our method can accurately infer agent constraints and generate robust motion plans. Future work includes generalizing our approach to infer temporally-extended constraints [25] and constraints with unknown parameterization via a Gaussian process-based modeling approach [26].

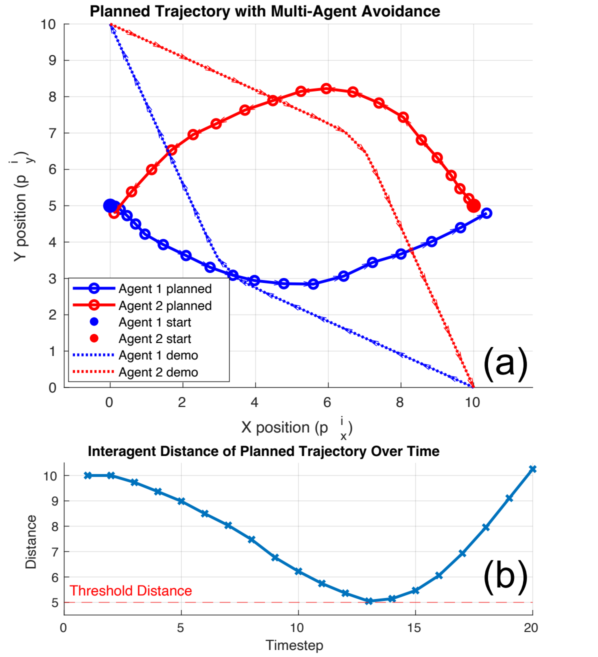

Fig. 2: (a) The multi-agent trajectories & (b) inter-agent distance checking w.r.t ground truth threshold. The planner can sample trajectories that satisfy **unknown / intangible** constraints embedded in the demonstration, without explicitly solving for the actual constraint parameters.

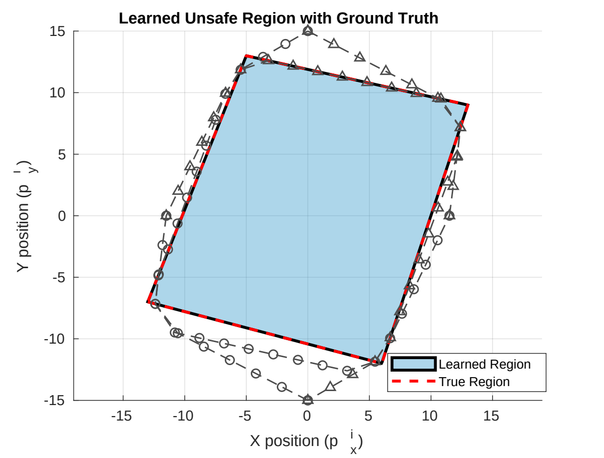

Fig. 3: Results of Sec. V-B. We plot the relative distances between agents from the demonstrations in gray. We plot the learned region (blue), which coincides with the true constraint (red). Dashed black lines mark the trajectories of Agent 1 (circle) and Agent 2 (triangle) from the two demonstrations, $\xi^{(1)}$ and $\xi^{(2)}$.

References

[1] Glen Chou, Necmiye Ozay, and Dmitry Berenson. "Learning Constraints From Locally-Optimal Demonstrations Under Cost Function Uncertainty". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3682–3690. DOI: 10.1109/LRA.2020.2974427.

[2] Glen Chou, Necmiye Ozay, and Dmitry Berenson. "Learning Parametric Constraints in High Dimensions from Demonstrations". In: *Conference on Robot Learning*

(CoRL). Ed. by Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura. Vol. 100. Proceedings of Machine Learning Research. PMLR, Oct. 2020, pp. 1211–1230.

[3] Glen Chou, Dmitry Berenson, and Necmiye Ozay. "Learning Constraints from Demonstrations". In: *Algorithmic Foundations of Robotics XIII*. Ed. by Marco Morales, Lydia Tapia, Gildardo Sánchez-Ante, and Seth Hutchinson. Cham: Springer International Publishing, 2020, pp. 228–245. ISBN: 978-3-030-44051-0.

[4] David L. McPherson, Kaylene C. Stocking, and S. Shankar Sastry. "Maximum Likelihood Constraint Inference from Stochastic Demonstrations". In: *2021 IEEE Conference on Control Technology and Applications (CCTA)*. 2021, pp. 1208–1213.

[5] Kaylene C. Stocking, David Livingston McPherson, Robert Peter Matthew, and Claire J. Tomlin. "Discretizing Dynamics for Maximum Likelihood Constraint Inference". In: *ArXiv* abs/2109.04874 (2021).

[6] Lasse Peters, David Fridovich-Keil, Vicenc Rubies-Royo, Claire J. Tomlin, and Cyrill Stachniss. "Inferring Objectives in Continuous Dynamic Games from Noise-Corrupted Partial State Observations". In: *Robotics: Science and Systems*. 2021.

[7] Jingqi Li, Chih-Yuan Chiu, Lasse Lasse Peters, Somayeh Sojoudi, Claire Tomlin, and David Fridovich-Keil. "Cost Inference for Feedback Dynamic Games from Noisy Partial State Observations and Incomplete Trajectories". In: *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '23. London, United Kingdom: International Foundation for Autonomous Agents and Multiagent Systems, 2023, pp. 1062–1070. ISBN: 9781450394321.

[8] Negar Mehr, Mingyu Wang, Maulik Bhatt, and Mac Schwager. "Maximum-Entropy Multi-Agent Dynamic Games: Forward and Inverse Solutions". In: *IEEE Transactions on Robotics* 39.3 (2023), pp. 1801–1815.

[9] 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* 6.3 (2021), pp. 5485–5492.

[10] Peter Englert, Ngo Anh Vien, and Marc Toussaint. "Inverse KKT: Learning Cost Functions of Manipulation Tasks from Demonstrations". In: *The International Journal of Robotics Research* 36.13-14 (2017), pp. 1474–1488. DOI: 10.1177/0278364917745980.

[11] Marcel Menner, Peter Worsnop, and Melanie N. Zeilinger. "Constrained Inverse Optimal Control With Application to a Human Manipulation Task". In: *IEEE Transactions on Control Systems Technology* 29.2 (2021), pp. 826–834.

[12] Negar Mehr, Roberto Horowitz, and Anca D. Dragan. "Inferring and Assisting with Constraints in Shared Autonomy". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. 2016, pp. 6689–6696.

[13] Changshuo Li and Dmitry Berenson. "Learning Object Orientation Constraints and Guiding Constraints for Narrow Passages from One Demonstration". In: *International Symposium on Experimental Robotics, ISER 2016*. Ed. by Dana Kulic, Yoshihiko Nakamura, Oussama Khatib, and Gentiane Venture. Vol. 1. Springer Proceedings in Advanced Robotics. Springer, 2016, pp. 197–210.

[14] Sylvain Calinon and Aude Billard. "A Probabilistic Programming by Demonstration Framework Handling Constraints in Joint Space and Task Space". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 367–372.

[15] Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*. SIAM, 1998.

[16] Bolei Di and Andrew Lamperski. "Newton's Method and Differential Dynamic Programming for Unconstrained Nonlinear Dynamic Games". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. 2019, pp. 4073–4078.

[17] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca Dragan, and Claire J Tomlin. "Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), pp. 1475–1481.

[18] Simon LeCleac'h, Mac Schwager, and Zachary Manchester. "ALGAMES: a Fast Augmented Lagrangian Solver for Constrained Dynamic Games". In: *Auton. Robots* 46.1 (Jan. 2022), pp. 201–215. ISSN: 0929-5593.

[19] Talha Kavuncu, Ayberk Yaraneri, and Negar Mehr. "Potential iLQR: A Potential-Minimizing Controller for Planning Multi-Agent Interactive Trajectories". In: *Robotics: Science and Systems* (July 2021).

[20] Chaitanya Awasthi and Andrew Lamperski. "Inverse Differential Games With Mixed Inequality Constraints". In: *2020 American Control Conference (ACC)*. 2020, pp. 2182–2187.

[21] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2024. URL: https://www.gurobi.com.

[22] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. "Aggressive Driving with Model Predictive Path Integral Control". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1433–1440. DOI: 10.1109/ICRA.2016.7487277.

[23] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. "Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving". In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1603–1622. DOI: 10.1109/TRO.2018.2865891.

[24] J. Löfberg. "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.

[25] Glen Chou, Necmiye Ozay, and Dmitry Berenson. "Learning temporal logic formulas from suboptimal demonstrations: theory and experiments". In: *Auton. Robots* 46.1 (2022), pp. 149–174.

[26] Glen Chou, Hao Wang, and Dmitry Berenson. "Gaussian Process Constraint Learning for Scalable Chance-Constrained Motion Planning From Demonstrations". In: *IEEE Robotics Autom. Lett.* 7.2 (2022), pp. 3827–3834.