2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

MPOGames: Efficient Multimodal Partially Observable Dynamic Games

Oswin So[1,2,*], Paul Drews[2], Thomas Balch[2], Velin Dimitrov[2], Guy Rosman[2], Evangelos A. Theodorou[1]

***Abstract*— Game theoretic methods have become popular for planning and prediction in situations involving rich multi-agent interactions. However, these methods often assume the existence of a single local Nash equilibria and are hence unable to handle uncertainty in the intentions of different agents. While maximum entropy (MaxEnt) dynamic games try to address this issue, practical approaches solve for MaxEnt Nash equilibria using linear-quadratic approximations which are restricted to *unimodal* responses and unsuitable for scenarios with multiple local Nash equilibria. By reformulating the problem as a POMDP, we propose MPOGames, a method for efficiently solving MaxEnt dynamic games that captures the interactions between local Nash equilibria. We show the importance of uncertainty-aware game theoretic methods via a two-agent merge case study. Finally, we prove the real-time capabilities of our approach with hardware experiments on a 1/10th scale car platform.**

I. Introduction and Related Work

In recent years, robots have been seen increasing use in applications with multiple interacting agents. In these applications, actions taken by the robot cannot be considered in isolation and must take the responses of other agents into account, especially in areas such as autonomous driving [1], robotic arms [2] and surgical robotics [3] where failures can be catastrophic. For example, in the autonomous driving setting, two agents at a merge must negotiate and agree on which agent merges first (Fig. 1). Failure to consider the responses of the other agent could result in serious collisions.

Game-theoretic (GT) approaches have become increasingly popular for planning in multi-agent scenarios [4–12]. By assuming that agents act rationally, these approaches decouple the problem of planning and prediction into the problems of finding objective functions for each agent and solving for Nash equilibrium of the resulting non-cooperative game. This provides a more principled and interpretable alternative to the prediction of other agents when compared to black-box approaches that use deep neural networks [13, 14].

In practice, assuming rational agents is too strict. Humans face cognitive limitations and make irrational decisions that are satisfactory but suboptimal, an idea known as "bounded rationality" or "noisy rationality" [15, 16]. Even autonomous agents are rarely *exactly* optimal and often make suboptimal decisions due to finite computational resources. One method of addressing this disconnect is the Maximum Entropy (MaxEnt) framework, which has been applied to diverse areas

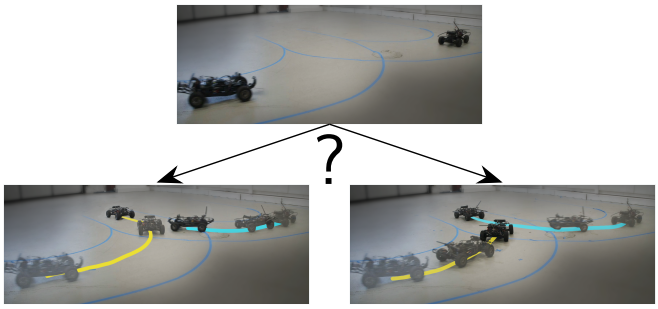

Fig. 1: The ego agent (left) seeks to merge into a lane safely without collisions. With multiple local minima present, MPOGames *predicts* likely plans for other agents, *infers* their probabilities, then *hedges* against all possibilities. Methods that fail to do so may result in catastrophic collisions.

such as inverse reinforcement learning [7, 17], forecasting [18] and biology [19]. In particular, this combination of MaxEnt and GT has been proposed for learning objectives for agents from a traffic dataset [7]. However, since the exact solution is computationally intractable, the authors use a linear quadratic (LQ) approximation of the game for which a closed-form solution can be efficiently obtained. The LQ approximation, popular among recent approaches (e.g., [5, 7]), only considers a single local minimum and consequently produces *unimodal* predictions for each agent. This fails to capture complex uncertainty-dependent hedging behaviors which mediate between other agents' possible plans in the true solution of the MaxEnt game.

These hedging behaviors are closely related to partially observable Markov decision process (POMDP) in the single-player setting, where solutions must consider the *distribution* of states when solving for the optimal controls. Similar multi-agent scenarios have been considered via POMDPs in [20–22]. Also related are partially observable stochastic games which are usually intractable [1] and mostly limited to the discrete setting [23–26].

One big challenge for methods that seek to address partial observability and multimodality is the problem of computational efficiency and scalability (e.g., in the case of POMDPs). Furthermore, only a few of the existing GT methods perform experiments on hardware [10, 11, 27], where the robustness of the method to noise and delays in the system is crucial. To tackle both of these issues, we present MPOGames, a framework for efficiently solving multimodal MaxEnt dynamic games by reformulating the problem as a game of incomplete information and efficiently finding approximate solutions to the equivalent POMDP problem. We showcase the real-time potential of our method via extensive hardware comparisons.

[1] Georgia Institute of Technology.
[2] Toyota Research Institute.
[*] Corresponding Author. oswinso@gatech.edu.

This work was supported by the Toyota Research Institute (TRI). This article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. Their support is gratefully acknowledged.

Our contributions in this work are as follows.

1) A computationally efficient framework for GT planning in multi-agent scenarios that produces more accurate solutions in the presence of multiple local minima compared to previous works.
2) A case-study in simulation comparing how GT methods handle multimodality that show the importance of inference and handling uncertainty.
3) Analysis of the performance of GT methods in multimodal situations with a 1/10th scale car platform. These results demonstrate the robustness and tractability of MPOGames on hardware.

II. Preliminaries

A. Discrete Dynamic Games and Nash Equilibria

We consider a discrete dynamic game with N players. Let $\mathbf{u}_t = [u_t^1, ..., u_t^N] = [u_t^i, \mathbf{u}_t^{\neg i}] \in \mathbb{R}^{n_\mathbf{u}}$ denote the joint control vector for all players, where we use $(\cdot)^i$ and $(\cdot)^{\neg i}$ to denote the partition of $(\cdot)$ into parts belonging to agent i and other agents respectively. We denote by $\mathbf{x}_t \in \mathbb{R}^{n_\mathbf{x}}$ the full state of the system at timestep t with nonlinear dynamics

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, u_t^1, ..., u_t^N) = f(\mathbf{x}_t, \mathbf{u}_t). \tag{1}$$

Time indices are dropped when they are clear from the context. Each agent's objective is to minimize a corresponding finite-horizon cost function J^i with running cost l^i and terminal cost Φ^i with respect to the control trajectory $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_{T-1}]$

$$\min_{U^i \in \mathcal{U}^i} J^i(U^i, \mathbf{U}^{\neg i}) = \min_{U^i \in \mathcal{U}^i} \Phi^i(\mathbf{x}_T) + \sum_{t=1}^{T-1} l^i(\mathbf{x}_t, \mathbf{u}_t), \tag{2}$$

where $\mathcal{U}^i$ denotes the set of feasible control trajectories for agent i. Note that the objective function J^i is a function of both U^i and $\mathbf{U}^{\neg i}$, but the optimization is only over U^i. Hence, the quantity of interest here is the Nash Equilibrium (NE), i.e., find controls $\mathbf{U}^*$ such that for each agent i,

$$J^i(U^{i*}, \mathbf{U}^{\neg i*}) \leq J^i(U^i, \mathbf{U}^{\neg i*}), \quad \forall U^i \in \mathcal{U}^i. \tag{3}$$

B. Maximum Entropy Dynamic Games

Agents rarely act *exactly* optimally due to cognitive or computational limitations and settle for satisfactory decisions [16]. We can model suboptimality via the noisy rationality model [15], where controls are stochastic according to the MaxEnt framework [17, 28]. Consider a *stochastic* control policy $\pi^i(u^i, \mathbf{x})$ and introduce an entropy regularization term to the original objective (2). We then consider the expected cost under all agents' policies $\boldsymbol{\pi}$

$$J^i(\boldsymbol{\pi}) = \mathop{\mathbb{E}}_{\mathbf{u}_t \sim \boldsymbol{\pi}} \left[\Phi^i(\mathbf{x}_T) + \sum_{t=1}^{T-1} l^i(\mathbf{x}_t, \mathbf{u}_t) - \frac{1}{\beta} H[\pi^i(\cdot|\mathbf{x}_t)] \right], \tag{4}$$

where $\beta > 0$ denotes inverse temperature or the rationality coefficient, and $H[\pi]$ is the Shannon entropy of π, defined as

$$H[\pi] := - \mathop{\mathbb{E}}_{u \sim \pi} [\log \pi(u)] = - \int \pi(u) \log \pi(u) \mathrm{d}u. \tag{5}$$

The resulting MaxEnt NE problem (referred to as "Entropic Cost Equilibrium" in [7]) is then to find stochastic policies $\boldsymbol{\pi}^*$ such that the following holds for each agent

$$J^i(\pi^{i*}, \pi^{\neg i*}) \leq J^i(\pi^i, \pi^{\neg i*}), \quad \forall \pi^i \in \Pi^i, \tag{6}$$

for feasible control policies Π^i from any initial state. Let the value function V^i for agent i given the policies of other agents $\boldsymbol{\pi}^{\neg i}$ be

$$V_t^i(\mathbf{x}_t) := \inf_{\pi^i} \{ J^i(\mathbf{x}, \pi^i, \boldsymbol{\pi}^{\neg i}) \}. \tag{7}$$

The optimal policy π^{i*} takes the form [7, 9]

$$\pi^{i*}(u^i|\mathbf{x}_t) = Z^i(\mathbf{x}_t)^{-1} \exp\big(-\beta Q(\mathbf{x}_t, u^i)\big), \tag{8}$$

where

$$Q^i(\mathbf{x}_t, u_t^i) := \mathop{\mathbb{E}}_{\boldsymbol{\pi}^{\neg i}} \big[{V^i}_{t+1}(\mathbf{x}_{t+1}) + l^i(\mathbf{x}_t, \mathbf{u}) \big], \tag{9}$$

and Z^i denotes the partition function

$$Z^i(\mathbf{x}_t) := \int \exp\left(-\beta \mathop{\mathbb{E}}_{\boldsymbol{\pi}^{\neg i}} \big[{V^i}_{t+1}(\mathbf{x}_{t+1}) + l^i(\mathbf{x}_t, \mathbf{u}) \big] \right) \mathrm{d}u^i. \tag{10}$$

Then, the value function V^i takes the form [29, Appendix A]

$$V_t^i(\mathbf{x}_t) = -\ln Z^i(\mathbf{x}_t). \tag{11}$$

While this gives us an expression for π^{i*} and V^i, it is generally intractable to compute the integral (10) and solve for $\boldsymbol{\pi}^*$ and Z^i under general nonlinear costs and dynamics.

One way of resolving this problem is to approximate the problem as a linear-quadratic (LQ) game [7, 9] in a manner similar to iLQR [30], an approach also taken in the iLQGames [5, 31] family of NE solvers. By doing so, the approximate game can be solved exactly [9, Lemma 1 and Lemma 2].

C. LQ is problematic for Multiodal MaxEnt NE

While taking LQ approximations yields a computationally efficient closed-form simulation for both the original NE and the MaxEnt NE problem, this approximation can be especially problematic in the MaxEnt NE setting. While the value function in the deterministic case does not depend on the cost function at states away from the NE, this is not the case for MaxEnt NE due to the integral in (8) and (11).

Illustrative Toy Example. We illustrate this using a toy example where the MaxEnt NE can be solved for in closed-form. Consider the following two-agent single timestep dynamic game with $\mathbf{x} = [x^1, x^2] \in \mathbb{R}^2$ and $\mathbf{u} = [u^1, u^2] \in \mathbb{R}^2$ with single-integrator dynamics and costs defined by

$$J^1 = \frac{1}{2}(u_0^1)^2 + \frac{3}{2}(x_1^1 - x_1^2)^2, \tag{12}$$

$$J^2 = \frac{1}{2}(u_0^2)^2 + \Phi^2(x_1^2), \tag{13}$$

$$\Phi^2(x_1^2) = \sigma_{\mathrm{SM}}\left(\frac{3}{2}(x_1^2 - 1)^2, \frac{3}{2}(x_1^2 + 1)^2 + \epsilon \right), \tag{14}$$

$$\sigma_{\mathrm{SM}}(a, b) := -\ln\big(e^{-a} + e^{-b}\big), \tag{15}$$

where $\epsilon > 0$ controls the "height" of the left local minima (Fig. 2 top right) and $x_0^1 = x_0^2 = 0$. The σ_{SM} in (14) acts

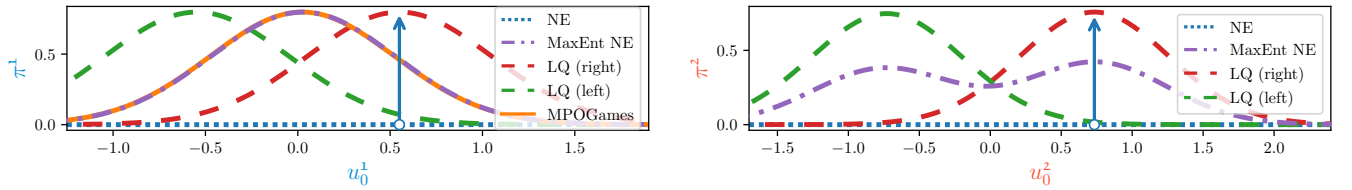

Fig. 2: Optimal policies for the ego agent P1 (**left**) and non-ego agent P2 (**right**) on the toy problem with $\epsilon = 0.1$. For π^1, only MaxEnt and POMDP hedges against both local NE of π^2. The deterministic policy of NE is denoted by the Dirac delta.

as a smooth minimum of the two quadratic state costs. In other words, P2 wants to reach either -1 or 1 (with a small preference for 1), while P1 wants to track P2.

NE: In the regular NE case, the optimization problem to solve for u_0^1 is exactly quadratic and gives $u_0^{1*} = \frac{3}{4}u_0^2$. Note that solution is *independent* of ϵ. We can also solve for u_0^{2*} numerically to obtain $u_0^{2*} = x_1^{2*} \approx 0.73$.

Exact MaxEnt NE: When solving the MaxEnt NE case, π^{2*} is a *Gaussian mixture*. Note the "shape" of the cost function J^2 determines the form of π^2. Consequently, the optimal π^{1*} is Gaussian with mean $0.25\tanh(\epsilon/2)$. In comparison to the NE case, both π^{1*} and π^{2*} depend on ϵ. Moreover, for $\epsilon \to 0$, the mean of π^{1*} approaches zero, which is *qualitatively different* behavior to the previous case.

LQ Approx MaxEnt NE: Using the LQ approximation, we obtain the following two sets of local LQ NE:

$$\pi^{1*,(1)} \approx \mathcal{N}(+0.55, 0.5), \quad \pi^{2*,(1)} \approx \mathcal{N}(+0.73, 0.53),$$
$$\pi^{1*,(2)} \approx \mathcal{N}(-0.55, 0.5), \quad \pi^{2*,(2)} \approx \mathcal{N}(-0.73, 0.53).$$

For each method, we plot π for both agents in Fig. 2. While the true MaxEnt π^1 hedges against both modes of π^2 with near zero mean, neither LQ approximations capture this.

III. Solving MaxEnt Dynamic Games as POMDPs

A. POMDPs: Better Approximations for MaxEnt NE

The main problem with LQ approximations for MaxEnt NE is that they work poorly when the non-ego agents' policies are not well approximated by a unimodal Gaussian. In the previous example, both LQ approximations alone are terrible approximations of true MaxEnt policy for π^2 (Fig. 2). On the other hand, if we use a more complex distribution such as a mixture model to model the optimal policy for each timestep, the number of mixture components will grow exponentially in the time horizon and is not computationally tractable.

To resolve this issue, we consider the following reformulation of the problem proposed in [9]. Taking inspirations from bounded rationality, we assume the non-ego agents are only capable of solving MaxEnt NE with LQ structure, but the ego agent does not know which LQ approximation is used. To model this, consider a set of M different LQ NE $\{(\mathbf{X}^{(z)}, \mathbf{U}^{(z)})_{z=1}^Z\}$, and let $z \in \{1, \dots, Z\}$ be a discrete latent random variable that determines which approximation is used by the non-ego agents. We now consider an extension of the MaxEnt Dynamic Game (6) where all non-ego agents have full knowledge of the value of z and are playing the corresponding optimal policy $\pi^{(z)}$. However, the non-ego agents incorrectly assume the ego agent knows what the true mode is. This is now a *dynamic game with incomplete information* — the ego agent only has a *belief* b of what the true game is but does not know what the true dynamics are [9, 32]. As shown in [9], the problem of computing the *Bayesian equilibrium* [32, Chapter 10] is equivalent to solving the following POMDP problem with state $\tilde{\mathbf{x}} := [\mathbf{x}, b]$ (where we assume the ego agent is agent 1)

$$V_t^1(\tilde{\mathbf{x}}_t) = \inf_{\pi^1} \mathop{\mathbb{E}}_{z \sim b_t} \mathop{\mathbb{E}}_{\mathbf{u}_t} \left[l^1(\mathbf{x}_t, \mathbf{u}_t) + V_{t+1}^1(\tilde{\mathbf{x}}_{t+1}) \right], \tag{16}$$

where $\mathbf{u}^{\neg 1} \sim \boldsymbol{\pi}^{\neg 1, z}$ and the dynamics of the belief b follow the Bayesian filtering update law

$$b_{t+1}(z) \propto \pi_t^{\neg 1, z}(\mathbf{u}_t^{\neg 1} | z) b_t(z). \tag{17}$$

Since computing b_t requires $\pi_{t-1}^{\neg 1, z}$, the LQ NE is solved from $\mathbf{x}_{t-1}$. We follow [9] and choose the prior belief b_0 as the MaxEnt distribution over the sum of agent value functions

$$b_0 = \inf_b \mathop{\mathbb{E}}_{z \sim b} \left[\sum_{i=1}^N V^{i,z}(\mathbf{x}_0) \right] - \frac{1}{\beta} H[b]. \tag{18}$$

Remark. *Unlike the usual POMDP setup in literature with discrete state and action spaces [33–35], the above setup can be considered a POMDP with hybrid states $[\mathbf{x}, z]$ ($\mathbf{x}$ is continuous but z is discrete), infinite-dimensional controls π^1 and continuous observations $\mathbf{x}_t$. Since $\mathbf{x}$ is fully observable, the belief $\tilde{b}$ is degenerate on the $\mathbf{x}$ coordinates.*

B. MPOGames

Leveraging this POMDP reformulation, we now present the MPOGames algorithm. First, we solve for a set of different LQ MaxEnt NE, which can be done in parallel. Since we assume each non-ego agent has picked one of these local NE and executes $\boldsymbol{\pi}^{\neg 1, z}$, we perform Bayesian inference using (17) to form a belief b over z. This belief is used to solve a POMDP to give an optimal policy π^1 for the ego agent. In practice, only the mean ego policy is used. A diagram of MPOGames is summarized in Fig. 3. Intuitively, this reformulation improves the fidelity of $\pi^{\neg 1}$ from a single to a mixture of Gaussians. This enables a more accurate expectation over non-ego agents' policies $\pi^{\neg 1}$ in (9), which is key to the behavior of the MaxEnt solution.

Returning to the toy example, MPOGames recovers similar hedging behavior to the exact MaxEnt case (ego P1).

MPOGames: The value functions for P1 take on the values $V^{1,(1)} = 0.17$ and $V^{1,(2)} = 0.09$. Consequently, for $\epsilon = 0.1$,

$$b_0 = [0.52, 0.48], \quad \pi^1 = \mathcal{N}(0.0, 0.5), \tag{19}$$

which displays similar hedging behavior as the exact MaxEnt NE solution (see Fig. 2 left).

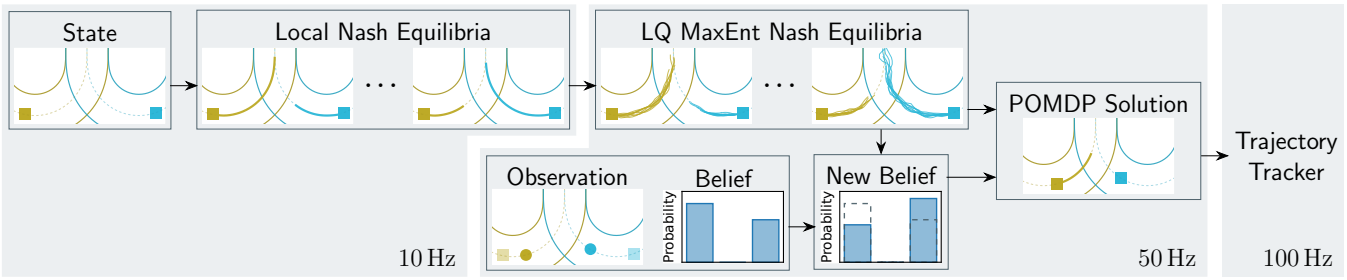

Fig. 3: Diagram of the proposed algorithm. In this work, we use QMDP for the POMDP solver. Belief updates and QMDP solving are performed at a much faster rate than the computation of local NE.

C. Approximate Solutions for POMDPs

Note that MPOGames does not prescribe a POMDP solver. However, exact solutions for POMDPs are generally intractable [33]. Two popular approximations for solving POMDPs are Q_{MDP} [35] and fast informed bound (FIB) [34]. Q_{MDP} has been used successfully in practice [36] and performs well when the particular action has little impact on the reduction in state uncertainty [33]. It can be shown that when the state $\mathbf{x}_t$ of all agents is directly observable, the Q_{MDP} and FIB approximations are identical and equal to the MaxEnt NE value function. Moreover, if the value function V for each mode is known, then the Q_{MDP} can be computed easily.

An alternative approach for solving POMDPs is to use point-based methods which yield more accurate solutions [33]. Instead of changing the information structure (i.e., assuming the state is known), these methods work with the full value function on beliefs. For example, [20, 37, 38] consider a scenario tree based structure for solving the POMDP. However, these methods still scale exponentially in the time horizon and have not been demonstrated to run in real-time in existing works [20, 21]. In comparison, QMDP with known V has constant complexity in the time horizon.

In this work, we implement MPOGames using the QMDP approximation for its effectiveness at low computational costs. While the value function $V^{1,z}$ for each mode is not known exactly, we take the quadratic approximation of V obtained from solving the (feedback) NE for each mode. The POMDP is solved with QMDP using the new belief and value functions, i.e.,

$$\min_{\pi_t^1} \mathbb{E}_{u_t^1} \mathbb{E}_{z\sim b_t} \left[l^{1,z} + V_{t+1}^{1,z}(\mathbf{x}_{t+1}^z) \right], \tag{20}$$

and the resulting solution is sent to a lower level trajectory tracker. Due to the efficiency of QMDP, we are able to solve it much faster than we can solve for local NE. Hence, we decouple the two parts in our algorithm to enable more responsive belief updates. The low level trajectory tracking is also decoupled to enable fast disturbance rejection from model errors in practice. The three decoupled nodes and their respective runtimes are presented in Fig. 3.

IV. Simulation Experiments

A. Autonomous Vehicle Merge

We consider a merge scenario with two agents merging into the same lane while respecting track boundaries and avoiding collisions and denote the ego agent by P0.

Agents are modeled as kinematic bicycles. To easily handle both types of state constraints, we consider a *hybrid* Cartesian-Curvilinear state representation similar to [39]. Let $x_i = [p_x, p_y, v, \theta, \zeta, s, n, \xi] \in \mathbb{R}^8$ and $u^i = [\mathrm{d}\zeta, a]$ denote the state and control for agent i. The tracks are represented as a constant width lane around a centerline. Both the signed curvature κ and the Cartesian coordinates of the centerline are represented as a cubic spline in the arclength s.

The cost for each agent consists of a quadratic cost for staying close to the centerline, maintaining a target velocity, and a soft constraint for staying inside the road boundaries. We handle collisions by treating them as soft-costs using a quadratic cost function on the Cartesian coordinates. Since this is concave in the position, there are two local NE — either the ego agent goes first, or the non-ego agent goes first (Fig. 4). Consequently, we do not know which NE the non-ego agent is considering and must perform inference.

We consider the following game-theoretic methods in this scenario:

1) (`NoInf`) No inference is done for the non-ego agent's considered mode, and we assume that the non-ego agent considers the same mode as the one we are considering. This is the approach taken by methods such [4, 5] which only consider a single local NE.
2) (`ML`) Inference is done. However, instead of solving the POMDP, the controls corresponding to the maximum-likelihood (ML) mode are used as in [12].
3) (`MPOGames`) Inference is done, and the resulting POMDP is solved using the QMDP approximation.

Although both `ML` and `MPOGames` perform inference, `ML` naively takes the most probable mode and ignores the rest. If the inferred probabilities are $b = [0.5+\epsilon, 0.5-\epsilon]$ for small ϵ, `ML` applies the policy for the first mode, while `MPOGames` will consider both modes almost equally.

Each method is run in a receding-horizon fashion with the results shown in Fig. 4 and Fig. 5. While `NoInf` performs well when the considered mode is the same as the non-ego agent, it fails when the mode is different — either both agents believe the other agent will yield and crash into each other, or both agents yield resulting in both agents stopping. Despite modeling the response of other agents, the "freezing" behavior here happens because the predicted response is incorrect. Methods that perform inference (`ML`, `MPOGames`) infer the correct mode in the noiseless case and successfully perform the merge. Note that the likelihood of the wrong

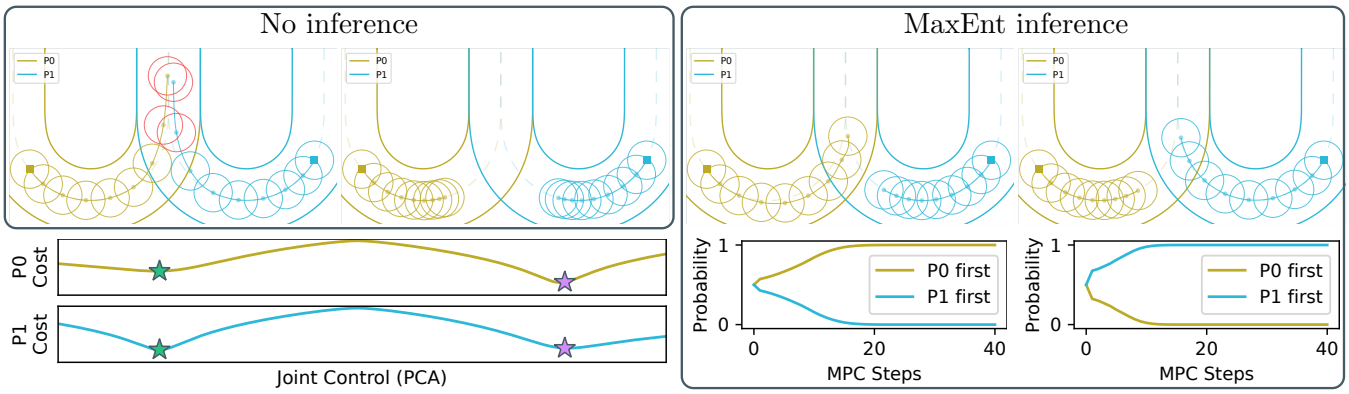

Fig. 4: **Top Left, Top Right**: Simulation snapshots of a merge scenario. The ego agent is on the left and in **gold**. The circles denote the radius for each robot. **Red** circles indicate collision. Being unable to consider multiple modes can lead to catastrophic failures such as collisions or freezing if the inferred mode is incorrect. **Bottom Left**: Visualization of the two local Nash equilibria for this problem. **Bottom Right**: Inferred probabilities for each mode via MaxEnt inference.

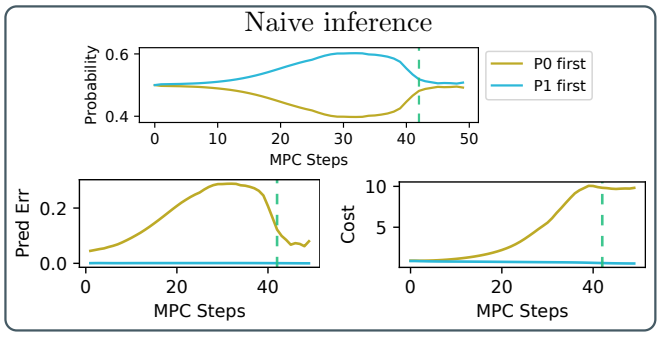

Fig. 5: In contrast to the MaxEnt probabilities in Fig. 4, naive inference results in both modes having *similar probabilities* near the end (eg. dashed line) due to both modes having similar prefixes at the merge point, despite the wrong mode having much higher cost. **Top**: Inferred probabilities for each mode via MaxEnt inference. **Left**: Prediction error for each mode. **Right**: Cost of each predicted mode.

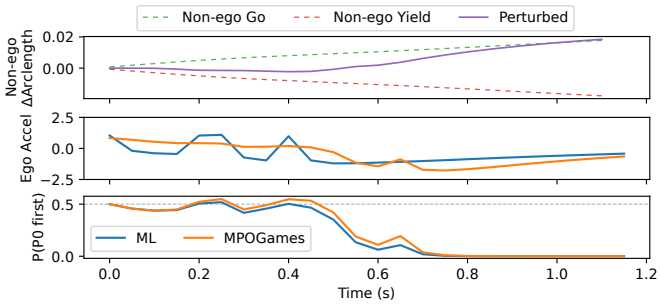

Fig. 6: Comparison between the `ML` and `MPOGames` methods when the non-ego agent uses suboptimal controls. The discontinuous nature of ML solution means that small perturbations in the non-ego agent's controls (**top**) results in a discontinuous, jerky ego control (**center**) when the uncertainty oscillates around 0.5 (**bottom**). On the other hand, `MPOGames` is a continuous function of the belief with smoother controls.

hypothesis when using `ML` actually *increases* at the end of the trajectory in Fig. 5. This occurs because the trajectory prefix of both modes becomes increasingly similar at the merge point despite the cost of the wrong mode being much larger.

We now consider the case where the non-ego agent's plans are not perfectly rational and perturb the non-ego agent's controls slightly such that there is large uncertainty regarding the non-ego agent's mode (Fig. 6). The high uncertainty of the true mode means that the maximum-likelihood mode oscillates between the two modes. Consequently, the discontinuity of `ML` results in a discontinuous, jerky ego control. In comparison, `MPOGames` hedges between the two possibilities and outputs a smooth, continuous control.

Choice of inference technique. Additionally, we investigate the difference between a naive method of inference via a user-specified Gaussian measurement model [12] and the MaxEnt inference presented in this paper. The results for naive inference are shown in Fig. 5 (cf. Fig. 4). Despite inferring the correct mode, this probability actually *decreases* towards the end despite larger costs.

Implementation Details. The symmetric structure of the coupling costs in the problems we consider makes this a potential game. Consequently, we solve for the (feedback) NE by reformulating the problem as an optimal control problem in a manner similar to [27, 40]. However, we note that the NE can be solved with other "backends" such as iLQGames [5], ALGames [4] or SQP [39], with the value function and policy recovered by performing an additional backwards pass of iLQGames. Next, although the belief update (17) requires access to the control inputs of other players, this information may not be directly available. Instead, we perform a least-squares estimate of $\mathbf{u}_t^{\neg 1}$ using $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. We also assume prior knowledge on how to find a comprehensive set of NE. In this work, this is done using a heuristic initialization process and fixed to two NE. Determining how many NE exist and how to find this set of NE is beyond the scope of this work. We use CasADi [41] and acados [42] to solve the resulting optimal control problem with $\Delta t = 0.05$s and horizon $T = 60$.

V. Hardware Experiments

Finally, we validate our approach on hardware in a merge scenario. We use the TRIKart [43], a modified version of the

TABLE I: Merge success rate on the scale car platform over ten trials from the same initial conditions.

		Non-ego Agent			
		Yield	NoYield	ML	MPOGames
Ego Agent	Yield	0.0			
	NoYield	1.0	0.2		
	ML	1.0	1.0	1.0	
	MPOGames	1.0	1.0	1.0	1.0

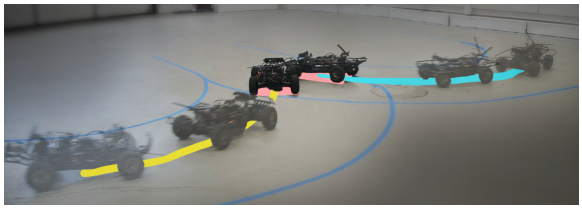

Fig. 7: NoYield-NoYield setup. Collisions occur when agents do not agree on the same local NE. Left is ego agent.

F1Tenth platform [44]. Optitrack [45] is used for localization.

A. No Perturbations

We first investigate how each method performs in the cases where the non-ego agent is cooperative (Yield), selfish (NoYield), or using the ML or MPOGames strategies. The first two cases (Yield, NoYield) are realized by performing NoInf with an initialization strategy that is biased towards yielding or not yielding respectively. The success rate of merging over ten trials is shown in Table I, where success is defined loosely as both agents reaching the center lane without colliding or stopping.

Yield, NoYield — Mismatch of Local NE: As predicted from Section IV-A, failures occur when there is a mismatch in local NE between the two ages (i.e., Yield-Yield and NoYield-NoYield). In the Yield-Yield case, all failures arose from both agents coming to a complete stop before the merge point. Due to noise, one of the agents eventually merges, but this only happens after a long delay. The two successes in the NoYield-NoYield example arose when one car was much closer to the merge point than the other car due to hardware differences, resolving the ambiguity. However, NoYield-NoYield was also the only combination that resulted in collisions (see Fig. 7).

ML — Vulnerablility to system latency: The simulation experiments in Section IV-A make the standard assumption that there is no delay between receiving observations and outputting the controls. However, this is not true in the real world. In the ML-ML setup, although all trials result in successful merges, both agents exhibit large oscillatory controls (Fig. 8). This behavior is inherent to the discontinuous behavior of ML. Despite having the same system latency, MPOGames-MPOGames handles this situation more gracefully due to accounting for uncertainty (Fig. 9).

B. Controls with High Uncertainty

Finally, we compare how each method performs when the non-ego agent has explicitly suboptimal controls. Specifically, we add sinusoidal noise to the non-ego agent's controls such that it is difficult to determine the non-ego agent's intentions.

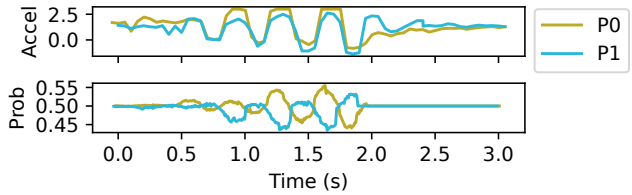

Fig. 8: Results on the scale car hardware when both agents use ML. The processing latency leads to an unstable, oscillatory system. **Top:** Output accelerations. **Bottom:** inferred probabilities of $P0$ (Ego) going first.

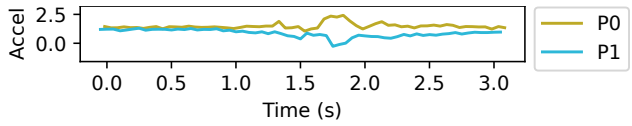

Fig. 9: Same setup as Fig. 8, but both agents use MPOGames.

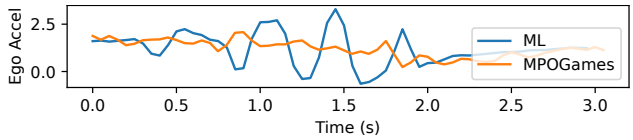

Fig. 10: Comparison of accelerations between ML and MPOGames *on hardware* when the non-ego agent applies high uncertainty controls. Note the oscillation for ML.

Here, we only compare ML and MPOGames and show the results in Fig. 10. This is similar to the scenario in Fig. 6, and indeed we see the same oscillations that ML displays on hardware. However, in this case, the non-ego agent is intentionally suboptimal. Again, MPOGames handles this uncertainty gracefully and correctly hedges against both NE.

VI. Discussion

Summary: We have proposed a MaxEnt game-theoretic framework for planning and inference in multi-agent situations with multiple local Nash equilibrium. Through simulation and hardware case studies, we have shown that MPOGames provides a real-time method for hedging against the uncertainty of other agents and improves performance compared to previous game-theoretic approaches.

Limitations and future work: Using a GT planner for joint planning and prediction assumes the knowledge of the cost function and dynamics for all agents. However, this is rarely true and consequently must be *estimated* using techniques such as inverse reinforcement learning [17, 46]. While the proposed reformulation into a POMDP relies on the key assumption that the non-ego agents know the true mode, we demonstrate that this restriction may be relaxed (e.g., MPOGames-MPOGames in Section V). Also, our framework assumes that non-ego agents are only capable of solving LQ games. Relaxing this assumption to model more intelligent non-ego agents is left as future work. Moreover, we have assumed prior knowledge on finding all NE. Doing so efficiently for general games is left as future work. Finally, the current work assumes that there are a discrete set of modes, but there exist scenarios when there are a *continuum* of NE. We leave this direction as future work.

References

[1] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state," *Autonomous Robots*, vol. 42, no. 7, pp. 1405–1426, 2018.

[2] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[3] T. Zhou and J. P. Wachs, "Early prediction for physical human robot collaboration in the operating room," *Autonomous Robots*, vol. 42, no. 5, pp. 977–995, 2018.

[4] S. L. Cleac'h, M. Schwager, and Z. Manchester, "Algames: A fast solver for constrained dynamic games," *arXiv preprint arXiv:1910.09713*, 2019. [Online]. Available: https://arxiv.org/abs/1910.09713

[5] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1475–1481.

[6] F. Laine, D. Fridovich-Keil, C.-Y. Chiu, and C. Tomlin, "Multi-hypothesis interactions in game-theoretic motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8016–8023.

[7] N. Mehr, M. Wang, and M. Schwager, "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *arXiv preprint arXiv:2110.01027*, 2021. [Online]. Available: https://arxiv.org/abs/2110.01027

[8] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, "Stochastic dynamic games in belief space," *IEEE Transactions on Robotics*, 2021.

[9] O. So, K. Stachowicz, and E. A. Theodorou, "Multimodal maximum entropy dynamic games," *arXiv preprint arXiv:2201.12925*, 2022. [Online]. Available: https://arxiv.org/abs/2201.12925

[10] Z. Wang, R. Spica, and M. Schwager, "Game theoretic motion planning for multi-robot racing," in *Distributed Autonomous Robotic Systems*. Springer, 2019, pp. 225–238.

[11] Z. Wang, T. Taubner, and M. Schwager, "Multi-agent sensitivity enhanced iterative best response: A real-time game theoretic planner for drone racing in 3d environments," *Robotics and Autonomous Systems*, vol. 125, p. 103410, 2020.

[12] L. Peters, D. Fridovich-Keil, C. J. Tomlin, and Z. N. Sunberg, "Inference-based strategy alignment for general-sum differential games," *arXiv preprint arXiv:2002.04354*, 2020. [Online]. Available: https://arxiv.org/abs/2002.04354

[13] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision*. Springer, 2020, pp. 683–700.

[14] Y. Ban, X. Li, G. Rosman, I. Gilitschenski, O. Meireles, S. Karaman, and D. Rus, "A deep concept graph network for interaction-aware trajectory prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8992–8998.

[15] R. D. Luce, "Individual choice behavior, a theoretical analysis," *Bull. Amer. Math. Soc*, vol. 66, no. 1960, pp. 259–260, 1960.

[16] H. A. Simon, "Theories of bounded rationality," *Decision and organization*, vol. 1, no. 1, pp. 161–176, 1972.

[17] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[18] B. P. Evans and M. Prokopenko, "A maximum entropy model of bounded rational decision-making with prior beliefs and market feedback," *Entropy*, vol. 23, no. 6, p. 669, 2021.

[19] A. De Martino and D. De Martino, "An introduction to the maximum entropy approach and its application to inference problems in biology," *Heliyon*, vol. 4, no. 4, p. e00596, 2018.

[20] D. Qiu, Y. Zhao, and C. L. Baker, "Latent belief space motion planning under cost, dynamics, and intent uncertainty," in *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

[21] H. Hu and J. F. Fisac, "Active uncertainty learning for human-robot interaction: An implicit dual control approach," *arXiv preprint arXiv:2202.07720*, 2022. [Online]. Available: https://arxiv.org/abs/2202.0772

[22] H. Hu, K. Nakamura, and J. F. Fisac, "Sharp: Shielding-aware robust planning for safe and efficient human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5591–5598, 2022.

[23] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, "Approximate Solutions For Partially Observable Stochastic Games with Common Payoffs," p. 8.

[24] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic Programming for Partially Observable Stochastic Games," p. 6.

[25] K. Horák and B. Bošanský, "Solving Partially Observable Stochastic Games with Public Observations," vol. 33, pp. 2029–2036. [Online]. Available: https://www.aaai.org/ojs/index.php/AAAI/article/view/4032

[26] A. Kumar and S. Zilberstein, "Dynamic Programming Approximations for Partially Observable Stochastic Games," p. 6.

[27] T. Kavuncu, A. Yaraneri, and N. Mehr, "Potential ilqr: A potential-minimizing controller for planning multi-agent interactive trajectories," *arXiv preprint arXiv:2107.04926*, 2021. [Online]. Available: https://arxiv.org/abs/2107.04926

[28] E. T. Jaynes, "On the rationale of maximum-entropy methods," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 939–952, 1982.

[29] O. So, Z. Wang, and E. A. Theodorou, "Maximum entropy differential dynamic programming," *arXiv preprint arXiv:2110.06451*, 2021. [Online]. Available: https://arxiv.org/abs/2110.06451

[30] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *ICINCO (1)*. Citeseer, 2004, pp. 222–229.

[31] D. Fridovich-Keil, V. Rubies-Royo, and C. J. Tomlin, "An iterative quadratic method for general-sum differential games with feedback linearizable dynamics," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2216–2222.

[32] M. Maschler, S. Zamir, and E. Solan, *Game theory*. Cambridge University Press, 2013.

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

[34] M. Hauskrecht, "Value-function approximations for partially observable markov decision processes," *Journal of artificial intelligence research*, vol. 13, pp. 33–94, 2000.

[35] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling, "Learning policies for partially observable environments: Scaling up," in *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 362–370.

[36] K. Hauser, "Online planning in continuous pomdps with open-loop information-gathering plans," in *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2011.

[37] D. Bernardini and A. Bemporad, "Stabilizing Model Predictive Control of Stochastic Constrained Linear Systems," vol. 57, no. 6, pp. 1468–1480. [Online]. Available: http://ieeexplore.ieee.org/document/6082380/

[38] A. Mesbah, "Stochastic model predictive control with active uncertainty learning: A Survey on dual control," vol. 45, pp. 107–117. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1367578817301232

[39] E. L. Zhu and F. Borrelli, "A sequential quadratic programming approach to the solution of open-loop generalized nash equilibria," *arXiv preprint arXiv:2203.16478*, 2022. [Online]. Available: https://arxiv.org/abs/2203.16478

[40] M. Bhatt, A. Yaraneri, and N. Mehr, "Efficient constrained multi-agent interactive planning using constrained dynamic potential games," *arXiv preprint arXiv:2206.08963*, 2022. [Online]. Available: https://arxiv.org/abs/2206.08963

[41] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[42] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.

[43] V. Dimitrov, P. Drews, T. Balch, X. Cui, A. A. Allaban, G. Rosman, and S. McGill. (2022, April) Rapid iterative design and testing with trikart. [Online]. Available: https://medium.com/toyotaresearch/rapid-iterative-design-and-testing-with-trikart-f5a3f59f2dd9

[44] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1tenth: An open-source evaluation environment for continuous control and reinforcement learning," *Proceedings of Machine Learning Research*, vol. 123, 2020.

[45] Optitrack systems. https://optitrack.com. (accessed on 4 September 2022).

[46] S. L. Cleac'h, M. Schwager, and Z. Manchester, "Lucidgames: Online

unscented inverse dynamic games for adaptive trajectory prediction and planning," *arXiv preprint arXiv:2011.08152*, 2020.