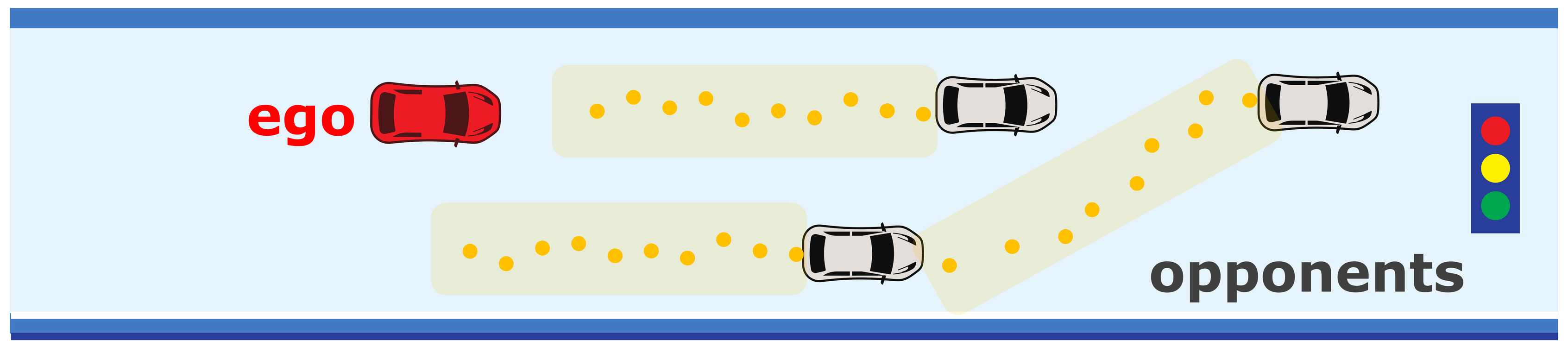
Learning to Play Trajectory Games Against Opponents with Unknown Objectives

Xinjie Liu*, Lasse Peters*, Javier Alonso-Mora | Delft University of Technology
IEEE Robotics and Automation Letters (RA-L) *equal contribution

TU Delft — Cognitive Robotics

Abstract

- **Game theory** provides a principled tool for modeling **multi-agent interaction**.
- However, **objectives of other agents**, such as humans, are often initially **unknown** to a robot.
- **Main Contribution:** We propose an **adaptive** model-predictive **game solver** that **estimates opponents' objectives online** based on past observations of their behavior.



Background: Parametric Trajectory Games

$$\forall i \in [N] \begin{cases} \min\limits_{X^i, U^i} J^i(X^i, U^i, \mathbf{X}^{\neg i}, \mathbf{U}^{\neg i}; \theta^i) \\ \text{s.t. } h^i(X^i, U^i) = 0 \\ \quad {}^p g^i(X^i, U^i) \geq 0 \\ \quad {}^s g(X^i, U^i, \mathbf{X}^{\neg i}, \mathbf{U}^{\neg i}) \geq 0 \end{cases}$$

e.g., 120 km/h, left lane

- **Coupled** trajectory **optimization** problems between players, solved in receding horizon.
- Solution: Generalized Nash equilibrium (GNE)

$$J^i(\mathbf{X}^*, \mathbf{U}^*; \theta^i) \leq J^i(X^i, U^i, \mathbf{X}^{\neg i*}, \mathbf{U}^{\neg i*}; \theta^i)$$

- Parameter $\boldsymbol{\theta}$: Players' intent, e.g., desired driving speed, lane preference, aggressiveness, etc.

Problem Formulation: Inverse Games as Equilibrium-Constrained Optimization

- We formulate objective inference as **equilibrium-constrained** maximum likelihood estimation:

observation

$$\max_{\boldsymbol{\theta}, \mathbf{X}, \mathbf{U}} \; p(\mathbf{Y} \mid \mathbf{X}, \mathbf{U})$$
$$\text{s.t. } (\mathbf{X}, \mathbf{U}) \text{ is a GNE of Game}(\boldsymbol{\theta})$$

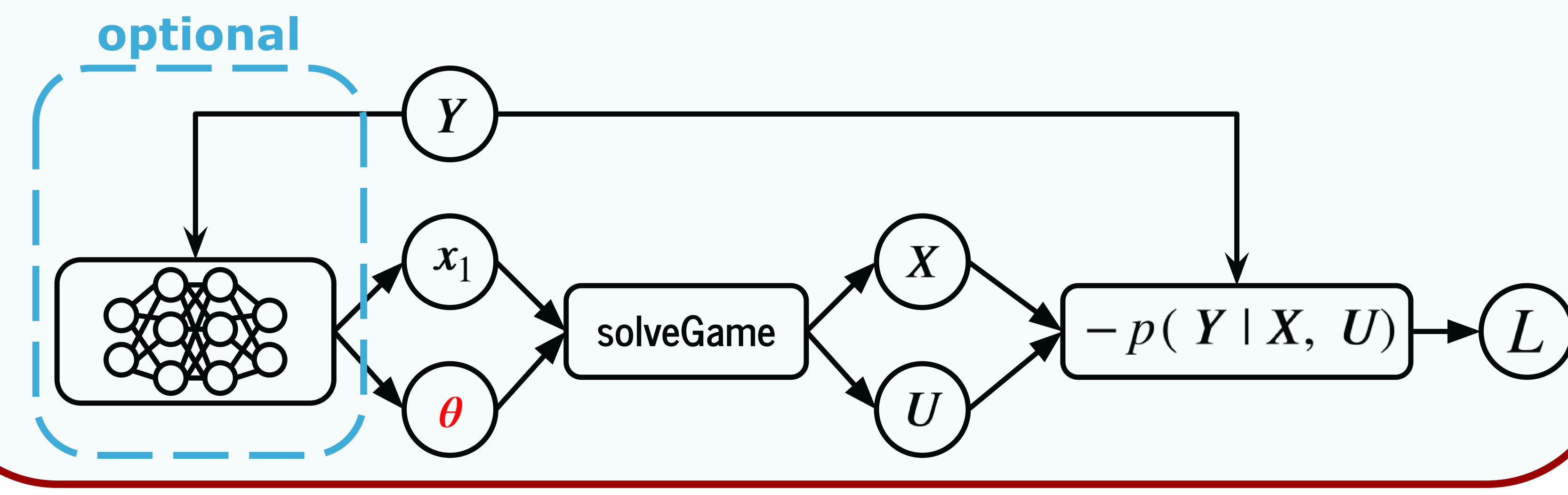
Karush–Kuhn–Tucker (KKT) conditions of the forward problem

- Challenges:
  - Nonconvexity
  - Complementarity constraints can present
  - Online computation

Approach: Solving Inverse Games via Differentiable Programming

Key Idea

We estimate $\boldsymbol{\theta}$ via **gradient descent** and by **differentiation** through the following computation graph:



Advantages / Features:

- Handles forward game **inequality constraints** in the inverse problem (complementarity constraints).
- Can be **combined with other differentiable modules**, such as neural networks.

High-Level Algorithm

Algorithm 1: Adaptive model-predictive game play (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($\tilde{\boldsymbol{\theta}}$) $\boldsymbol{z}^*$: **game solution**
  $\nabla_{\boldsymbol{\theta}} \mathcal{L} \leftarrow$ composeGradient($\boldsymbol{z}^*, \nabla_{\boldsymbol{\theta}} \boldsymbol{z}^*, \mathbf{Y}$)
  $\tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} - \nabla_{\boldsymbol{\theta}} \mathcal{L} \cdot \mathrm{lr}$
end
/* forward game */
$\boldsymbol{z}^* \leftarrow$ solveMCP($\tilde{\boldsymbol{\theta}}$)
applyFirstEgoInput($\boldsymbol{z}^*$)
return $\tilde{\boldsymbol{\theta}}, \mathbf{Y}$

Game as a Mixed Complementarity Problem (MCP)

- We compute gradients of the game solution by inspecting its first-order necessary conditions in a mixed complementarity problem (MCP), where only one of the following holds for each element of the decision variable $\boldsymbol{z}$:

$$z_j^* = \ell_j, F_j(\boldsymbol{z}^*; \boldsymbol{\theta}) \geq 0$$
$$\ell_j < z_j^* < u_j, F_j(\boldsymbol{z}^*; \boldsymbol{\theta}) = 0$$
$$z_j^* = u_j, F_j(\boldsymbol{z}^*; \boldsymbol{\theta}) \leq 0,$$

where

$$\boldsymbol{z} = [\mathbf{X}, \mathbf{U}, \boldsymbol{\mu}, {}^p\lambda^1, \dots, {}^p\lambda^N, {}^s\boldsymbol{\lambda}],$$
$$F(\boldsymbol{z}; \boldsymbol{\theta}) = [\nabla_{(X^1,U^1)} \mathcal{L}^1, \dots, \nabla_{(X^N,U^N)} \mathcal{L}^N, h, {}^p g^1, \dots, {}^p g^N, {}^s g],$$
$$\ell = [-\infty, \dots, -\infty, -\infty, 0, \dots, 0, 0],$$
$$u = [\infty, \dots, \infty, \infty, \infty, \dots, \infty, \infty].$$

Differentiation through an MCP

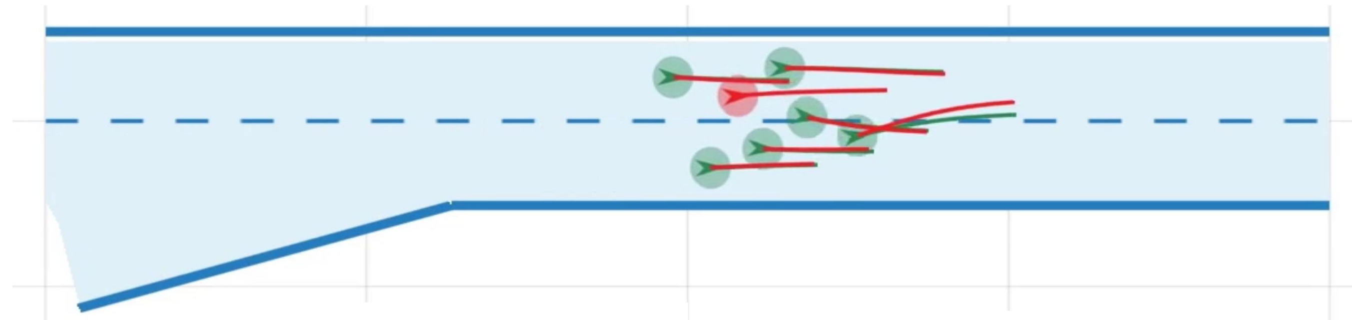
- We apply the **implicit function theorem** to the MCP solution:
  - For active bound indices:
    $\tilde{\mathcal{I}} := \{k \in [n] \mid z_k^* = \ell_k \vee z_k^* = u_k\}$
    $\nabla_{\boldsymbol{\theta}} \tilde{z}^* = 0$
  - For inactive bound indices:
    $\bar{\mathcal{I}} := \{k \in [n] \mid F_k(\boldsymbol{z}^*; \boldsymbol{\theta}) = 0, \ell_k < z_k^* < u_k\}$

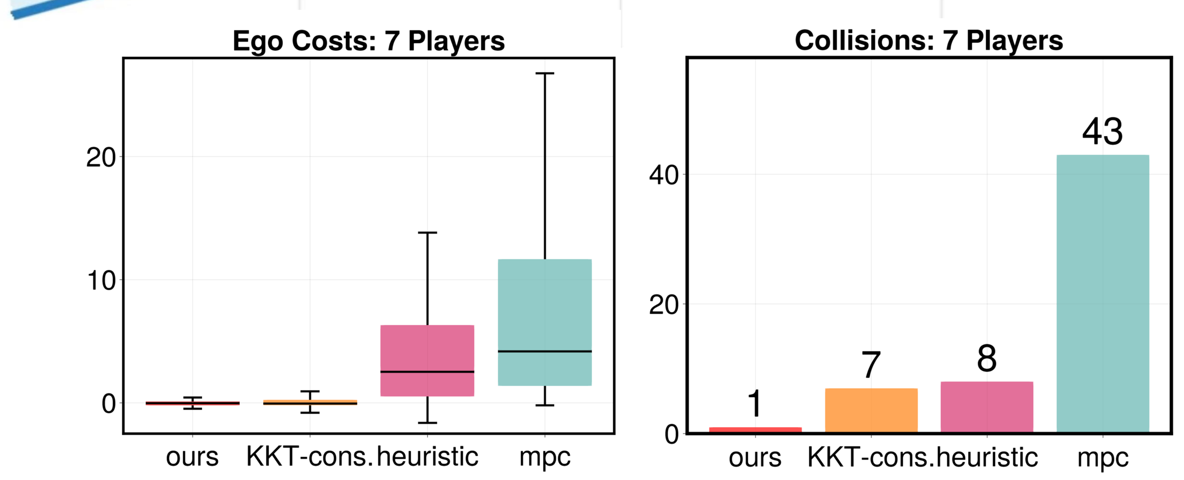
$$0 = \nabla_{\boldsymbol{\theta}} \left[\bar{F}(\boldsymbol{z}^*(\boldsymbol{\theta}), \boldsymbol{\theta})\right] = \nabla_{\boldsymbol{\theta}} \bar{F} + (\nabla_{\bar{z}^*} \bar{F})(\nabla_{\boldsymbol{\theta}} \bar{z}^*) + (\nabla_{\tilde{z}^*} \bar{F}) \underbrace{(\nabla_{\boldsymbol{\theta}}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

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

Experimental Results

Simulated Interaction

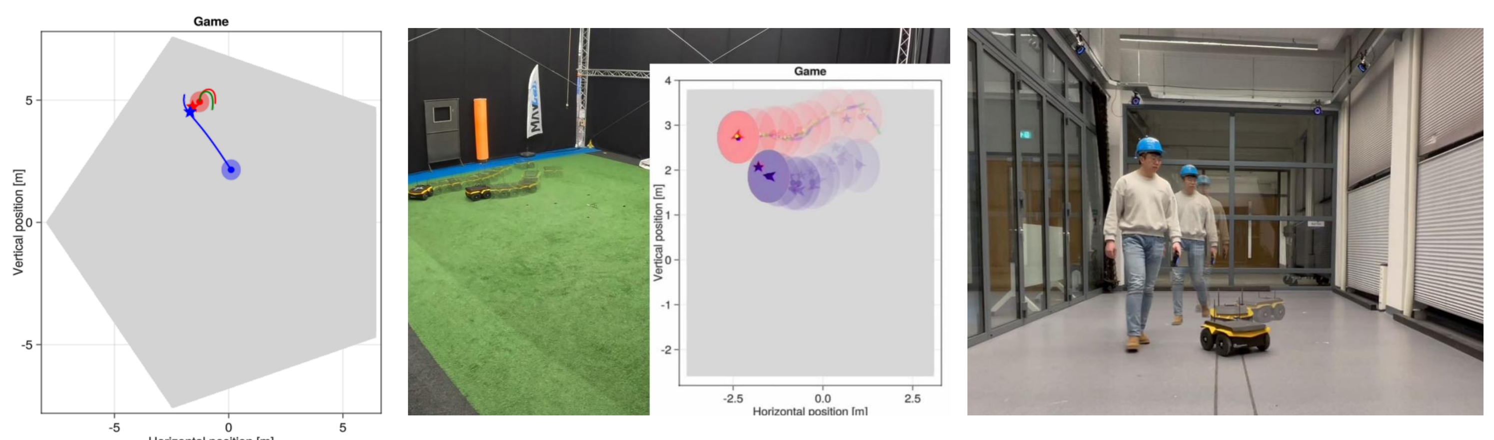




- Objective inference improves both **efficiency** and **safety**. Modeling game **inequalities** in inverse games improves safety in dense scenarios.

Hardware Experiments

- A robot infers the unknown opponent's goal location and tracks them while avoiding collisions.



Project Website

https://xinjie-liu.github.io/projects/game/