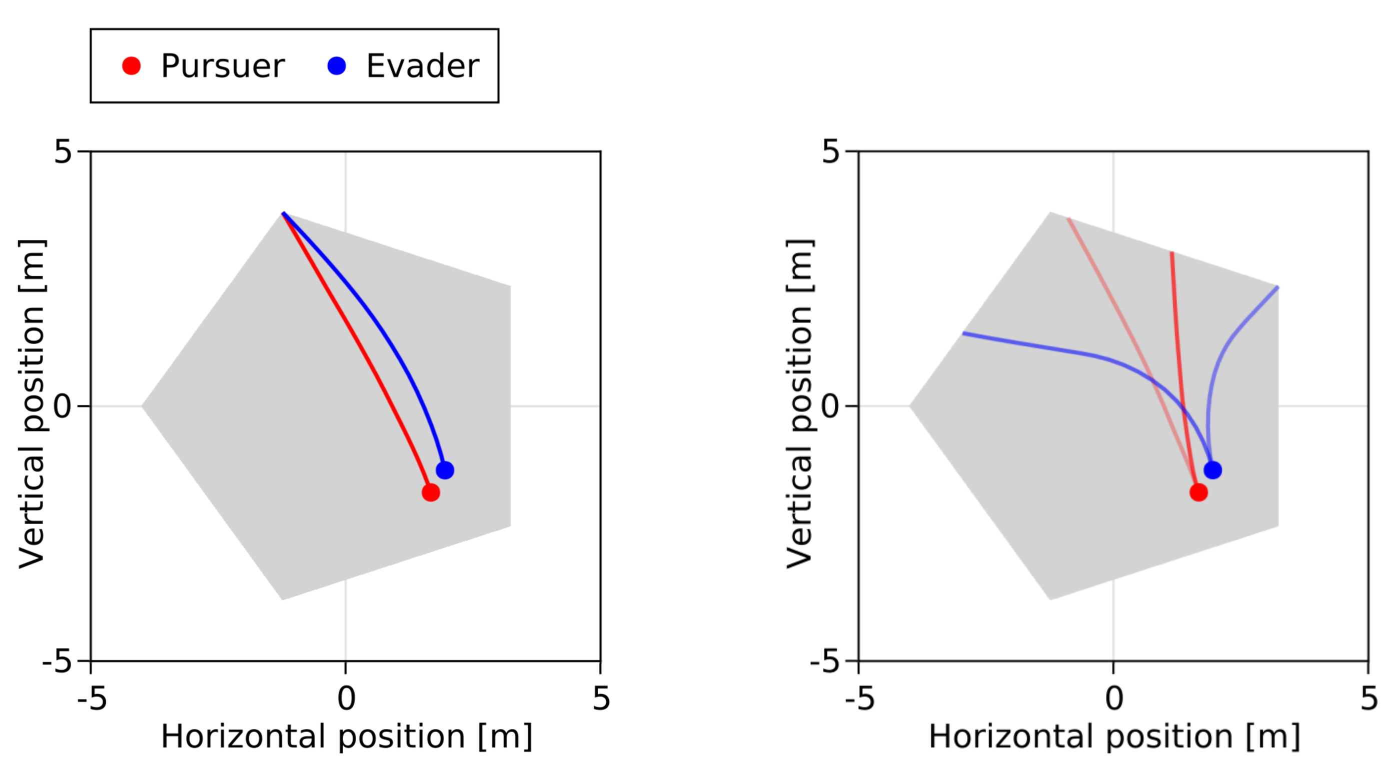
Learning Mixed Strategies in Trajectory Games

Lasse Peters, David Fridovich-Keil, Laura Ferranti,
Cyrill Stachniss, Javier Alonso-Mora, Forrest Laine

TU Delft · TEXAS The University of Texas at Austin · UNIVERSITÄT BONN · VANDERBILT School of Engineering

Abstract

- Dynamic game theory studies the strategic interaction of multiple players with individual objectives over time.
- **Challenge**: Pure strategies may be exploited
  - Example, Fig 1: In the tag game, a competitive evader may randomize their actions to remain unpredictable.
- We propose a **lifted formulation** of trajectory games which naturally admits **mixed strategy solutions**.
- We enable efficient solutions of these lifted games by combining **differentiable games** with trajectory optimization in a **structured learning** framework.



(a) Pure strategies (b) Mixed Strategies

Fig 1: Two player tag game solutions.

Website

lasse-peters.net/pub/lifted-games

Method

Vanilla (Non-Lifted) Trajectory Games

- Players optimize a **single trajectory** to minimize their respective cost

$$\min_{\tau_1} f_1(\tau_1, \tau_2) \quad \text{s.t. } \tau_1 \in \mathcal{K}_1 \qquad \min_{\tau_2} f_2(\tau_1, \tau_2) \quad \text{s.t. } \tau_2 \in \mathcal{K}_2$$

- Nash equilibrium

$$\forall i \in [N], \forall \tau_i \in \mathcal{K}_i\colon f_i(\tau_i^*, \tau_{\neg i}^*) \leq f_i(\tau_i, \tau_{\neg i}^*)$$

Lifted Trajectory Games

- Each player minimizes their **expected cost** by optimizing the parameters of a categorical **distribution over trajectories**.

$$\min_{q_1, \tau_1^1, \dots, \tau_1^{n_1}} \mathop{\mathbb{E}}_{\substack{\tau_1 \sim \mathcal{T}_1 \\ \tau_2 \sim \mathcal{T}_2}} [f_1(\tau_1, \tau_2)] \qquad \min_{q_2, \tau_2^1, \dots, \tau_2^{n_2}} \mathop{\mathbb{E}}_{\substack{\tau_1 \sim \mathcal{T}_1 \\ \tau_2 \sim \mathcal{T}_2}} [f_2(\tau_1, \tau_2)]$$

$$\text{where } \forall i \in [N]: \quad \mathcal{T}_i := \text{Cat}\Big(\underbrace{\{\tau_i^1, \dots, \tau_i^{n_i}\}}_{\text{trajectory candidates}}, \overbrace{q_i}^{\text{mixing weights}} \Big), \ \forall j \in [n_i]: \tau_i^j \in \mathcal{K}_i$$

Reducing Run-Time Computation

- For given trajectory candidates of all players, the problem reduces to a bimatrix game (BMG).
- We show that this BMG can be differentiated efficiently and use this gradient signal to train **trajectory generators** for all players.
- A **differentiable trajectory optimization** layer ensures feasibility of the candidate plans.

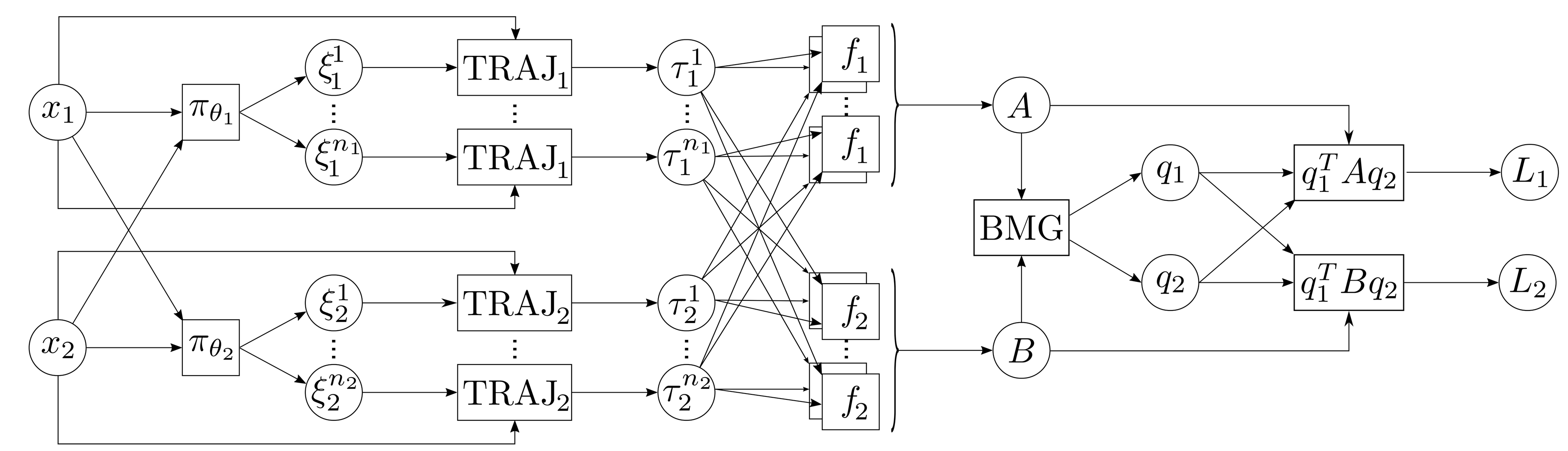


Fig 2: Pipeline overview. Note that the structure of the computation graph admits efficient parallelization.

Results

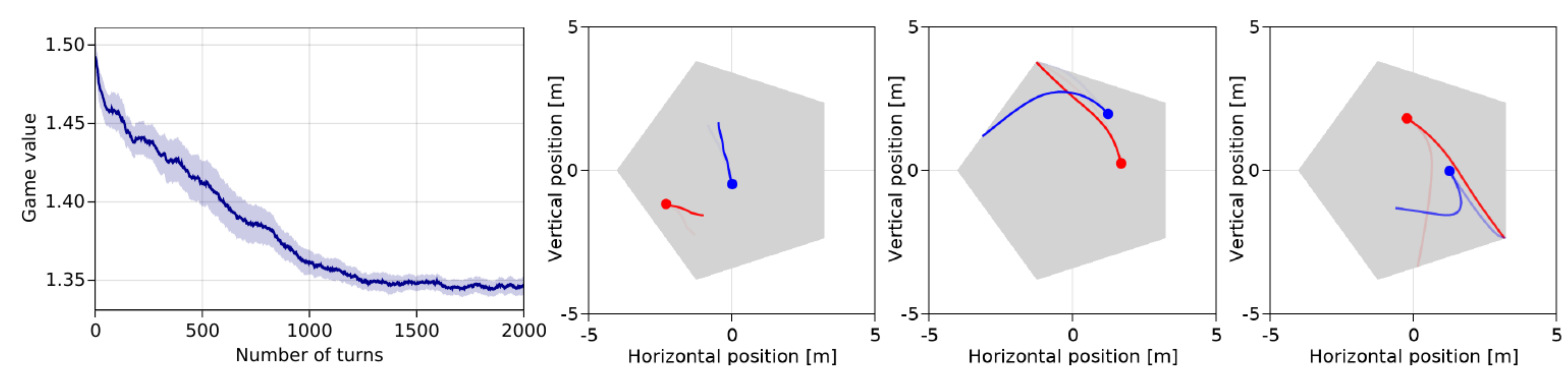
- Lifting converges to qualitatively different solutions compared to a non-lifted formulation (Fig 1).
- In a two-player tag game, **lifting provides a competitive advantage**
  An evader that uses a lifted strategy can secure a higher average distance to the pursuer. The pursuer's best response is then to also use lifting.
- Reference generators can be easily trained in self-play within 2min on a standard laptop. A forward pass of the pipeline takes less than 2ms.

TABLE I: Open-loop competition.

Pursuer	Evader: Lifted	Evader: Pure
Lifted	1.577 ± 0.021	1.502 ± 0.022
Pure	1.672 ± 0.022	1.370 ± 0.027

TABLE II: Receding-horizon competition.

Pursuer	Evader: Lifted	Evader: Pure
Lifted	1.360 ± 0.003	1.289 ± 0.005
Pure	1.463 ± 0.004	0.903 ± 0.009



(a) Closed-loop game value (b) Initial strategies (c) Intermediate strategies (d) Final strategies

Fig 3: Training trajectory generators in self-play.