Markov Near-Potential Functions

A New Paradigm for Design and Analysis of Multi-agent Systems

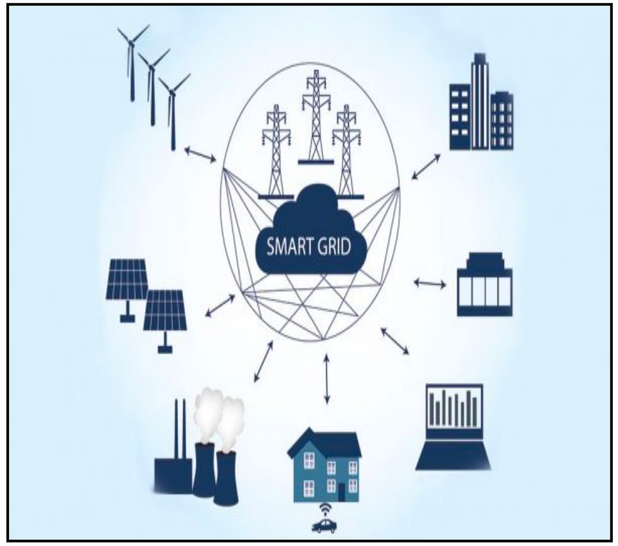
Chinmay Maheshwari
Johns Hopkins University

Game On! Seminar
April 7, 2026

Data science and AI based systems are increasingly permeating society

Mobility Systems

Energy Systems



Robotics

Online Marketplaces

amazon mechanical turk

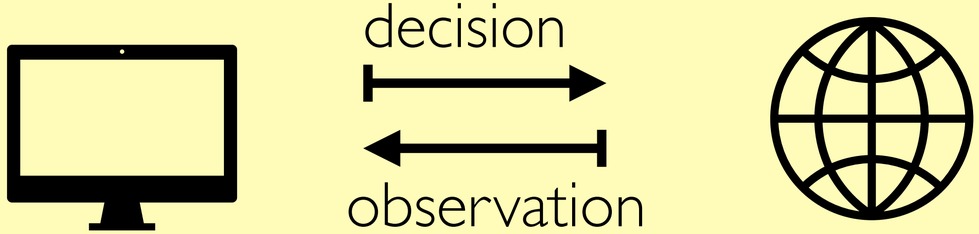
match.

Google Ads

DOORDASH

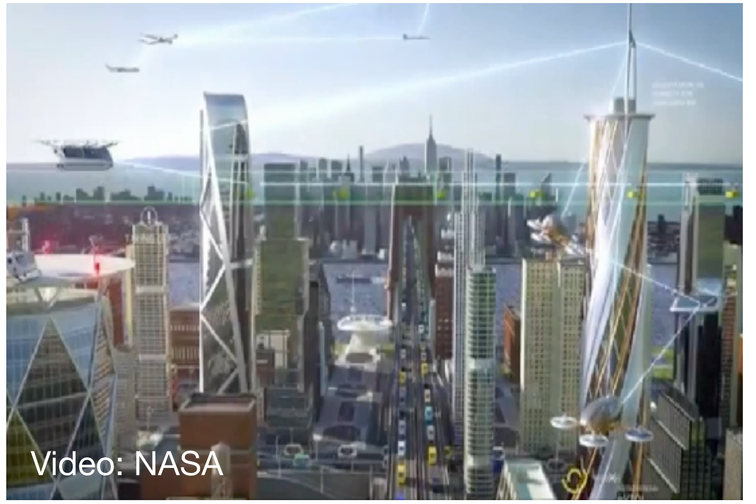
Robinhood

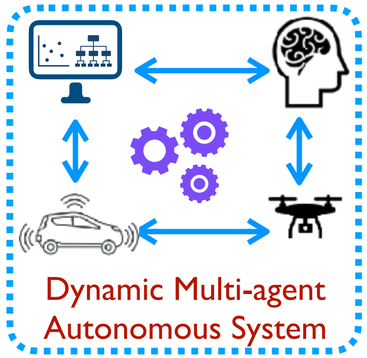
Dominant Design Philosophy

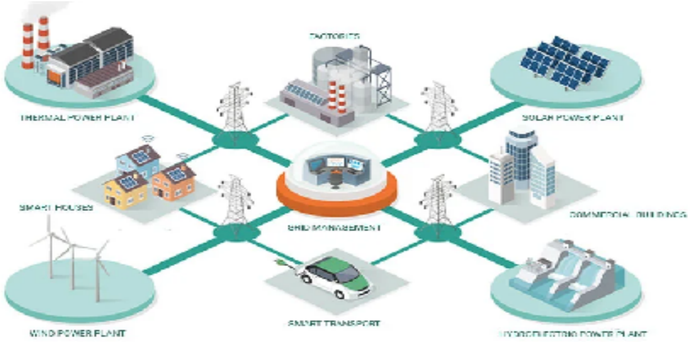


Challenge

Strategic interaction in dynamic multi-agent environment

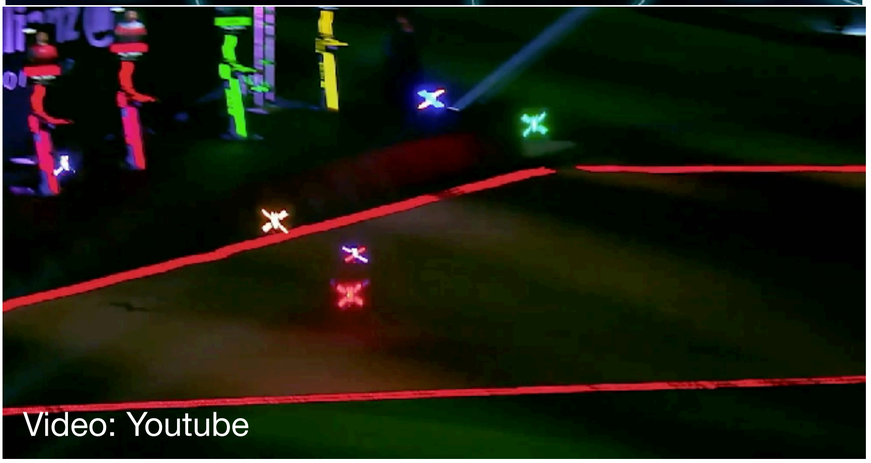


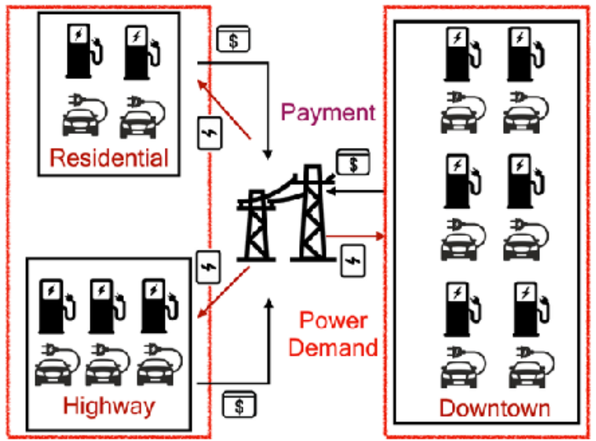




AI and **IoT-Driven** Smart Grid
Technologies for Smart Energy Management

www.genuspower.com

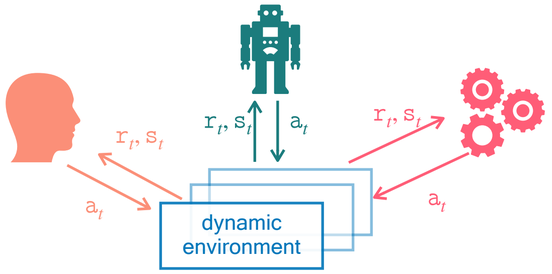




Today's Talk

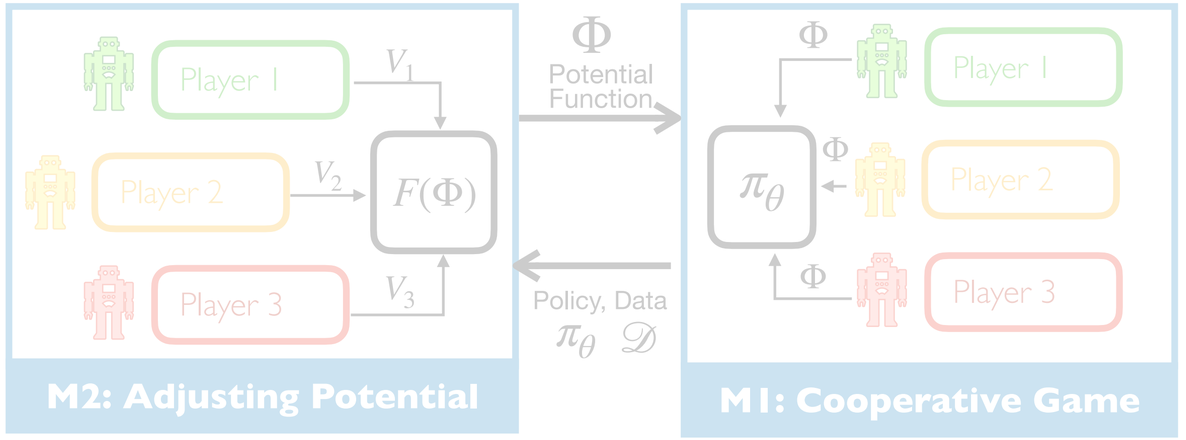
Challenge — Strategic interaction in dynamic multi-agent environment

Interaction between decentralized RL agents



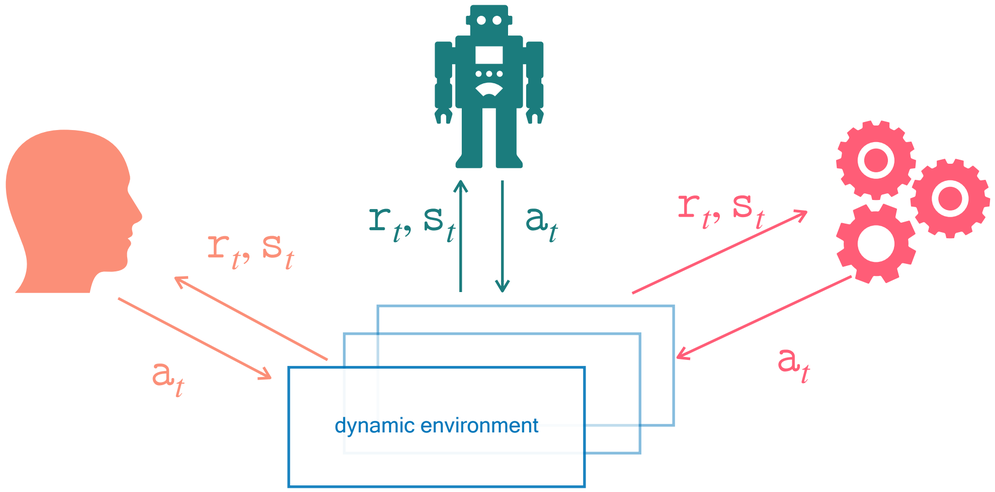
Designing strategies in multi-car racing

Deep Multi-agent RL for general-sum games



Approach — Markov Near-Potential Functions

Convergence of Decentralized Actor-Critic Algorithms in General-sum Markov Games

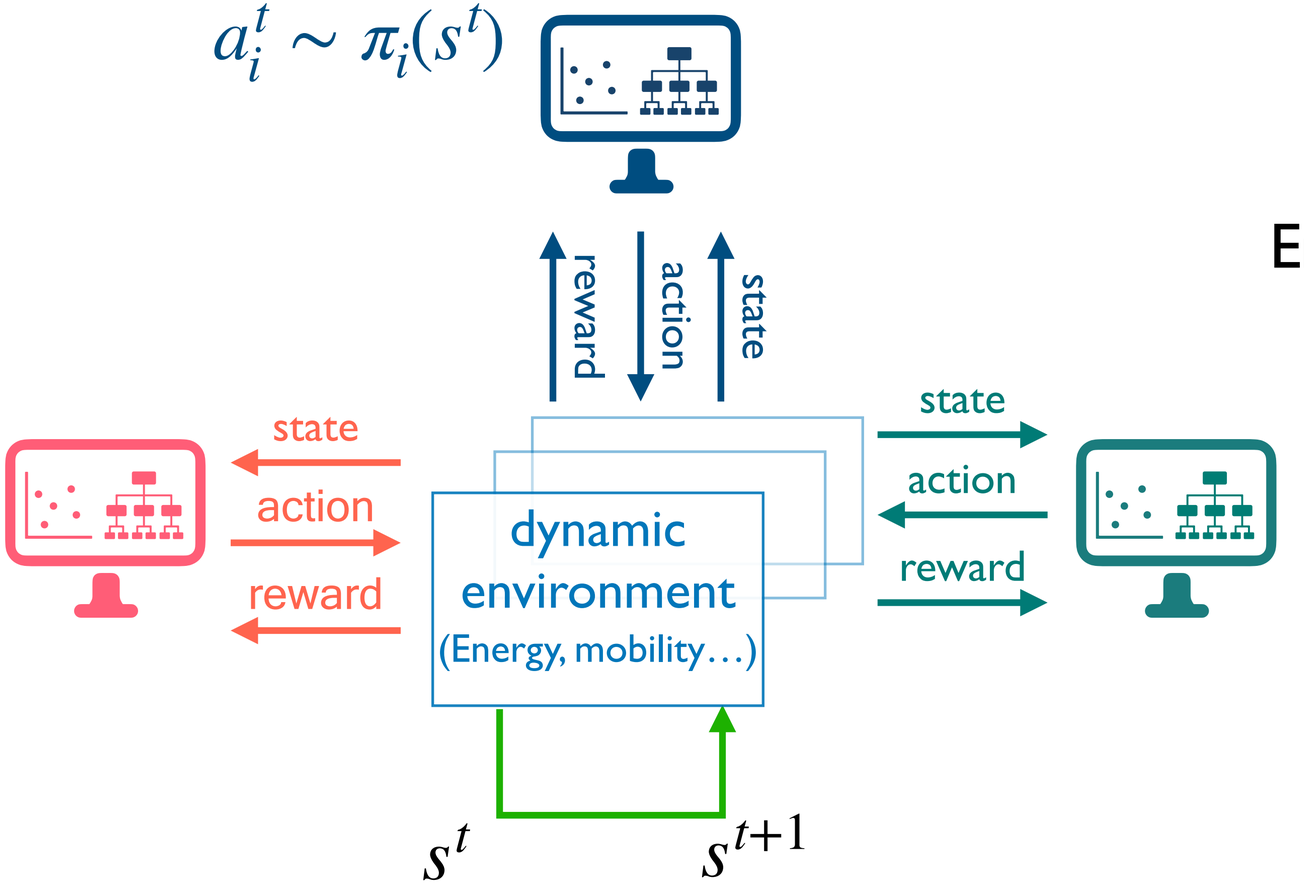


C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

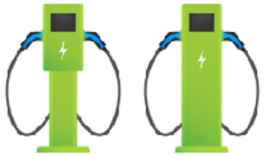
C. Maheshwari, M. Wu, D. Pai, S. Sastry. Independent and Decentralized Learning in Markov Potential Games. IEEE TAC 2025

Decentralized RL algorithms in shared environments

$a_i^t \sim \pi_i(s^t)$



Distributed
Energy Resources



EV
Charging

- Non-myopic agents

$$V_i(s,\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t r_i^t \,|\, s^0 = s\right]$$

Decentralized RL algorithms in shared environments

$a_i^t \sim \pi_i(s^t)$

$u_i(s^t, a_i^t, a_{-i}^t) = r_i^t$

Distributed
Energy Resources

EV
Charging

reward

action

state

(Energy, mobility…)

s^t $s^{t+1} \sim P(\cdot \mid s^t, a^t)$

What is the long-run outcome of interaction between decentralized MARL algorithms?

- Non-myopic agents

$$V_i(s, \pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t r_i^t \mid s^0 = s\right]$$

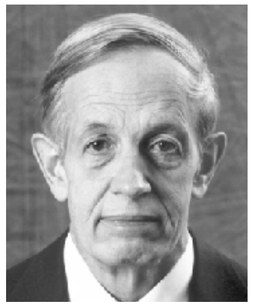
- Decentralized learning (actor-critic)
  as agents do no know actions of others
- Bandit feedback of state transition and
  heterogeneous rewards

Related Works

[Static games] *Fudenberg and Levine (1998), Leslie and Collins (2006), Marden, Arslan, and Shamma (2009), Cominetti, Melo, and Sorin (2010), Panageas and Piliouras (2016), …*

[Purely Competitive + Purely Cooperative Markov Games] *Leslie, Perkins and Xu (2020), Daskalakis, Foster, and Golowich (2020), Sayin, Zhang, Leslie, Basar, and Ozdaglar (2021), Sayin and Ulnu (2022), Guo, Fu, Yang, and Wang (2021), Sayin, Parise, Ozdaglar (2022), Baudin and Laraki (2022) …*

Outcome
Nash
equilibrium



A policy π^* is ϵ-Nash equilibrium if

$$V_i(\mu, \pi_i^*, \pi_{-i}^*) \geq V_i(\mu, \pi_i, \pi_{-i}^*) - \epsilon$$

$$\forall\ i \in I, \mu \in \Delta(S)$$

[Weaker Convergence] *Borkar (2002), Jin, Liu, Wang and Yu (2023), Li, Chi, Wei and Chen (2022), Mao and Basar (2022), Song, Mei and Bai (2022), Zhang, Kakade, Basar and Yang (2020), …*

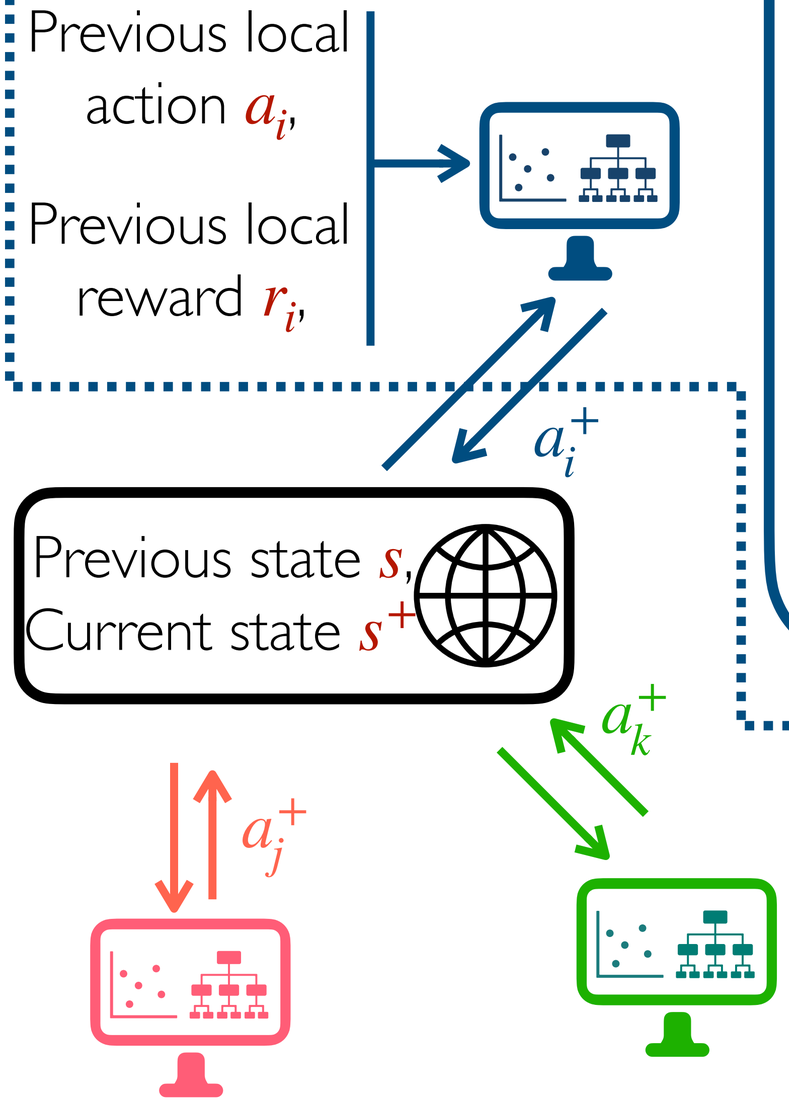
Our goal is to analyze the convergence of standard reinforcement learning algorithms in dynamic multi-agent settings, without assuming any structure on utility functions

Contributions

- First characterization of **convergent set** of interaction between decentralized actor-critic algorithms **in general-sum (aka mixed competitive and cooperative) Markov games**
- A **new framework (Markov near-potential function)** to study multi-agent interaction in dynamic games

C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024
C. Maheshwari, M. Wu, D. Pai, S. Sastry. Independent and Decentralized Learning in Markov Potential Games. IEEE TAC 2025

Decentralized Actor-Critic Algorithm



Critic: Update q-function only in previous state-action

$$q_i^+(s,a_i) = q_i(s,a_i) + \gamma_{\#(s,a_i)} \underbrace{\left(r_i + \delta \pi_i(s^+)^\top q_i(s^+) - q_i(s,a_i)\right)}_{\text{TD Error}}$$

Actor: Update policy only in previous state

$$\pi_i^+(s) \in \pi_i(s) + \beta_{\#s} \left(\arg\max_{p \in \Delta(A_i)} \sum_{a \in A_i} p^\top q_i(s) - \pi_i(s) \right)$$

Exploratory Action $\quad a_i^+ \sim (1-\theta)\pi_i(s) + \theta \cdot (1/|A_i|)\mathbf{1}_{A_i}$

$$\lim_{n \to \infty} \frac{\beta_n}{\gamma_n} = 0$$

Timescale Separation

Markov Near-potential Function

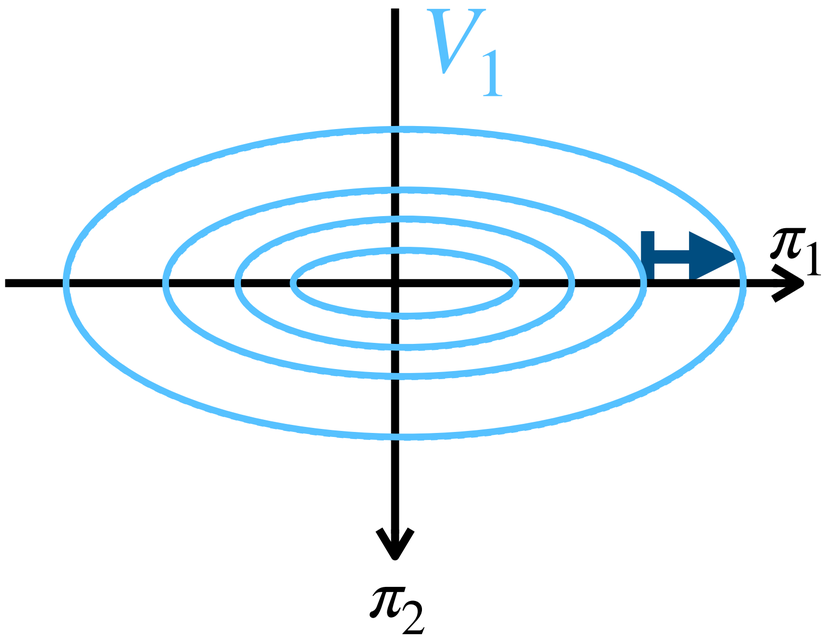
A function $\Phi : S \times \Pi \to \mathbb{R}$ is called a **Markov Near Potential Function (MNPF)** for game G with **closeness parameter** κ if for all $\pi_i, \pi_i' \in \Pi_i, \pi_{-i} \in \Pi_{-i}$

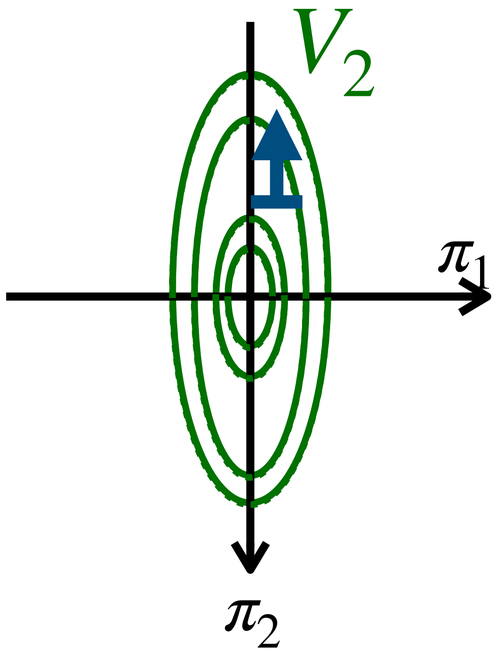
$$\left| \Phi(\mu, \pi_i', \pi_{-i}) - \Phi(\mu, \pi_i, \pi_{-i}) - (V_i(\mu, \pi_i', \pi_{-i}) - V_i(\mu, \pi_i, \pi_{-i})) \right| \leq \kappa \|\pi_i - \pi_i'\|$$

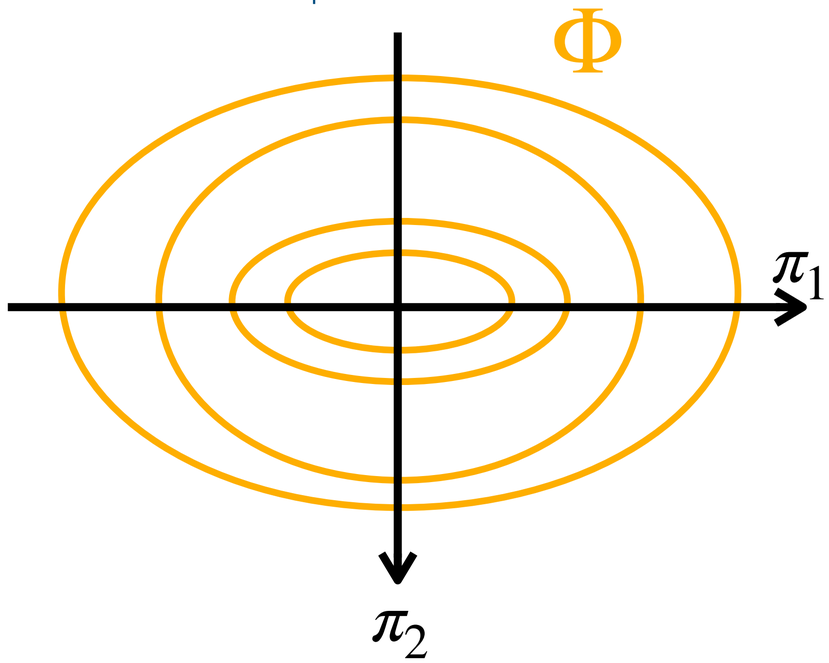
Change in potential due to unilateral shift

Change in value due to unilateral shift

Closeness parameter







X. Guo, X. Li*, **C. Maheshwari**, S. Sastry, M. Wu. Markov α-Potential Games. IEEE TAC 2025

C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

Markov Near-potential Function

A function $\Phi : S \times \Pi \to \mathbb{R}$ is called a Markov Near Potential Function (MNPF) for game G

Structural Properties

- Every **Markov game** admits a Markov near potential function with some closeness parameter κ
- **Closeness of gradients:** For any Markov game G and an associated near potential function Φ with closeness parameter κ,

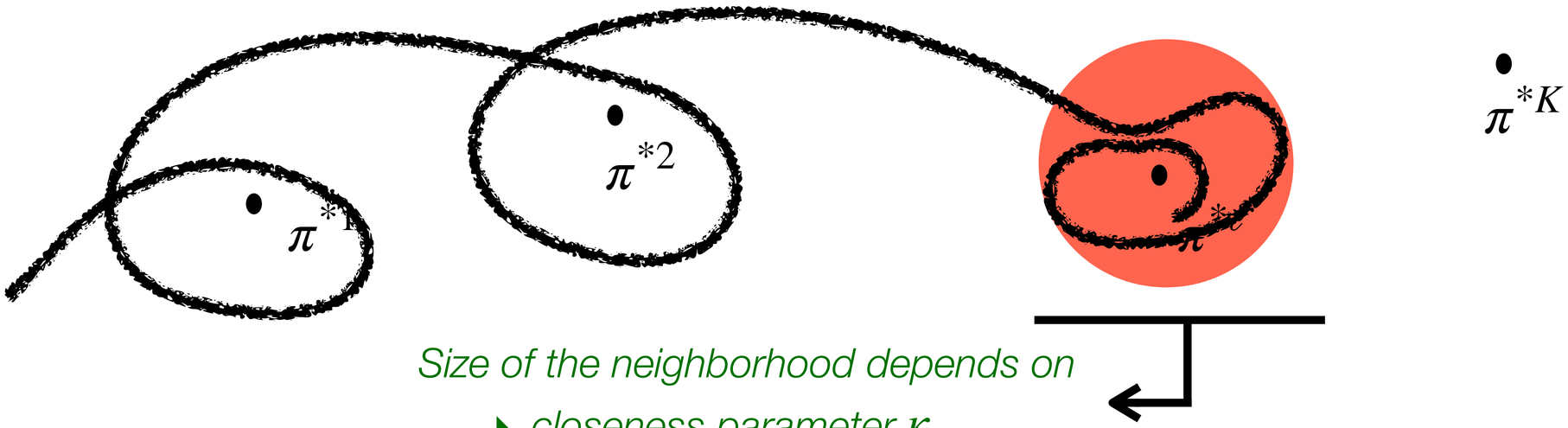
$$\left| v_i^\top \frac{\partial \Phi(\mu, \pi)}{\partial \pi_i} - v_i^\top \frac{\partial V_i(\mu, \pi)}{\partial \pi_i} \right| \leq \kappa \quad \forall\ i \in I, v_i \in \mathbb{R}^{|S| \times |A_i|}, \pi \in \Pi$$

X. Guo, X. Li*, **C. Maheshwari**, S. Sastry, M. Wu. Markov α-Potential Games. IEEE TAC 2025
C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

Theorem (informal)

Consider a Markov game G and an associated near-potential function Φ, with closeness parameter κ, such that game has finitely many Nash equilibria.

Then the **decentralized actor-critic learning algorithms** will converge to a neighborhood of one of these equilibria



Size of the neighborhood depends on

- *closeness parameter κ*
- *exploration rates of users θ*
- *ergodic properties of state transition*

C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

Sneak Peek Into Analysis

- Asynchronous two-timescale stochastic approximation theory

- **Fast time-scale:** Assume that policy is stationary and analyze q-dynamics

$$\pi^{\theta}_{-i}(s) = (1-\theta)\pi_{-i}(s) + \theta(1/|A_i|)\mathbf{1}_{A_i}$$

$$q_i^+(s,a_i) = q_i(s,a_i) + \gamma_{\#(s,a_i)}\left(r_i + \delta\pi_i(s^+)^\top q_i(s^+) - q_i(s,a_i)\right)$$

C. Maheshwari, M. Wu, D. Pai, S. Sastry. Independent and Decentralized Learning in Markov Potential Games. IEEE TAC 2025

Sneak Peek Into Analysis

- Asynchronous two-timescale stochastic approximation theory

- **Fast time-scale:** Assume that policy is stationary and analyze q-dynamics

$$\pi^{\theta}_{-i}(s) = (1-\theta)\pi_{-i}(s) + \theta(1/|A_i|)\mathbf{1}_{A_i}$$

$$q_i^{+}(s,a_i) = q_i(s,a_i) + \gamma_{\#(s,a_i)}\left(r_i + \delta\pi_i(s^{+})^{\top} q_i(s^{+}) - q_i(s,a_i)\right)$$

$$q_i(s) \to Q_i(s,\pi_i,\pi^{\theta}_{-i}) = u_i(s,a_i,\pi_{-i}) + \delta\sum_{s'\in S} P(s'\,|\,s,a_i,\pi^{\theta}_{-i})V_i(s;\pi_i,\pi^{\theta}_{-i})$$

Contraction property of TD-operator

C. Maheshwari, M. Wu, D. Pai, S. Sastry. Independent and Decentralized Learning in Markov Potential Games. IEEE TAC 2025

Sneak Peek Into Analysis

- **Slow time-scale:** Assume convergent q-dynamics, analyze the policy updates

$$\pi_i^+(s) \in \pi_i(s) + \beta_{\#s}\left(\arg\max_{p\in\Delta(A_i)} \sum_{a\in A_i} p^\top q_i(s) - \pi_i(s)\right)$$

$$q_i(s) \to Q_i(s, \pi_i, \pi_{-i}^\theta)$$

$$\frac{d}{d\tau}\pi_i^\tau(s) \in \pi_i(s) + \beta_{\#s}\left(\arg\max_{p\in\Delta(A_i)} \sum_{a\in A_i} p^\top Q_i(s, \pi_i^\tau, \pi_{-i}^\theta) - \pi_i^\tau(s)\right)$$

Main step: Take $\phi(\tau) = \Phi_{\max} - \Phi(\pi^\tau)$ as an **(approximate) Lyapunov function**

If π^τ is not a $C(\kappa+\theta)$-Nash equilibrium then $\frac{d}{d\tau}\phi(\tau) < 0$

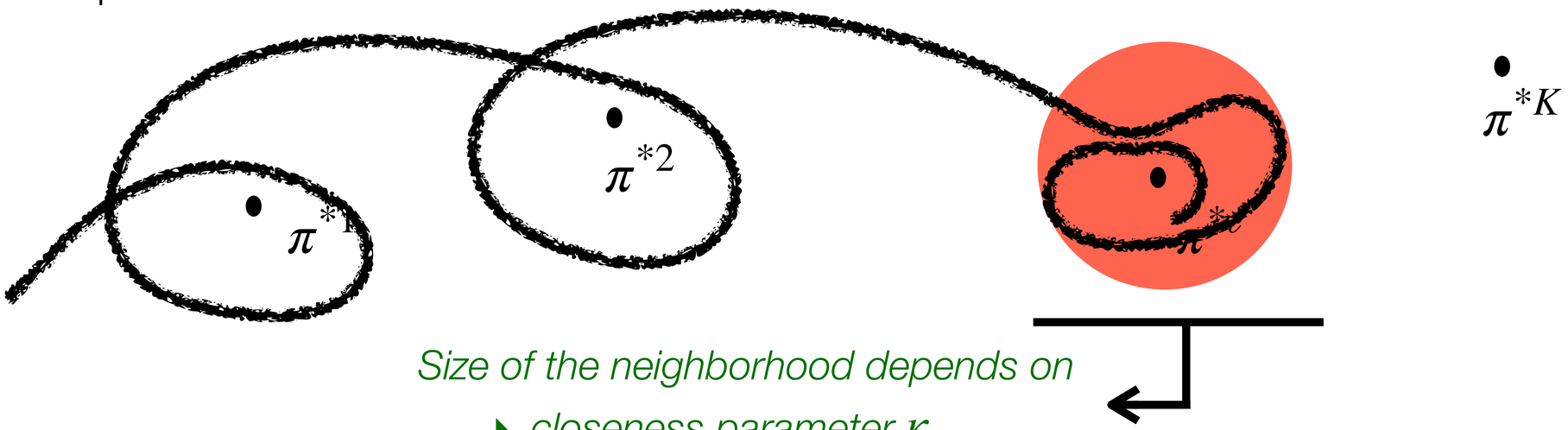
C. Maheshwari, M. Wu, D. Pai, S. Sastry. Independent and Decentralized Learning in Markov Potential Games. IEEE TAC 2025

Theorem (informal)

Consider a Markov game G and an associated near-potential function Φ, with closeness parameter κ, such that

1. Game has finitely many Nash equilibria, and
2. The near-potential function is Lipschitz continuous

Then the **decentralized actor-critic learning algorithms** will converge to a neighborhood of one of these equilibria



Size of the neighborhood depends on

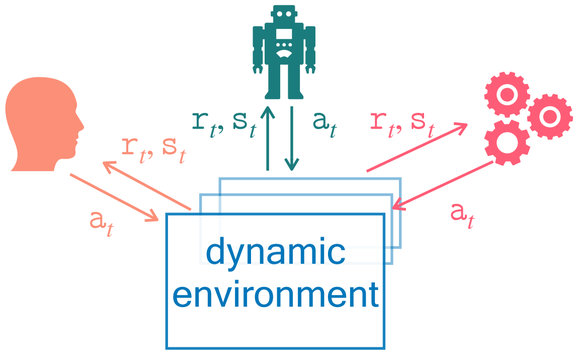
- *closeness parameter κ*
- *exploration rates of users θ*
- *ergodic properties of state transition*

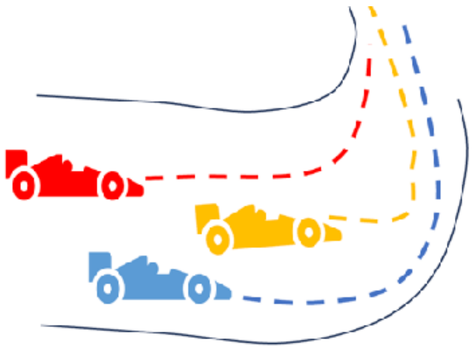
C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

Markov Near-potential Function

Analysis Tool

Design Tool





Today's Talk

Challenge	Strategic interaction in dynamic multi-agent environment

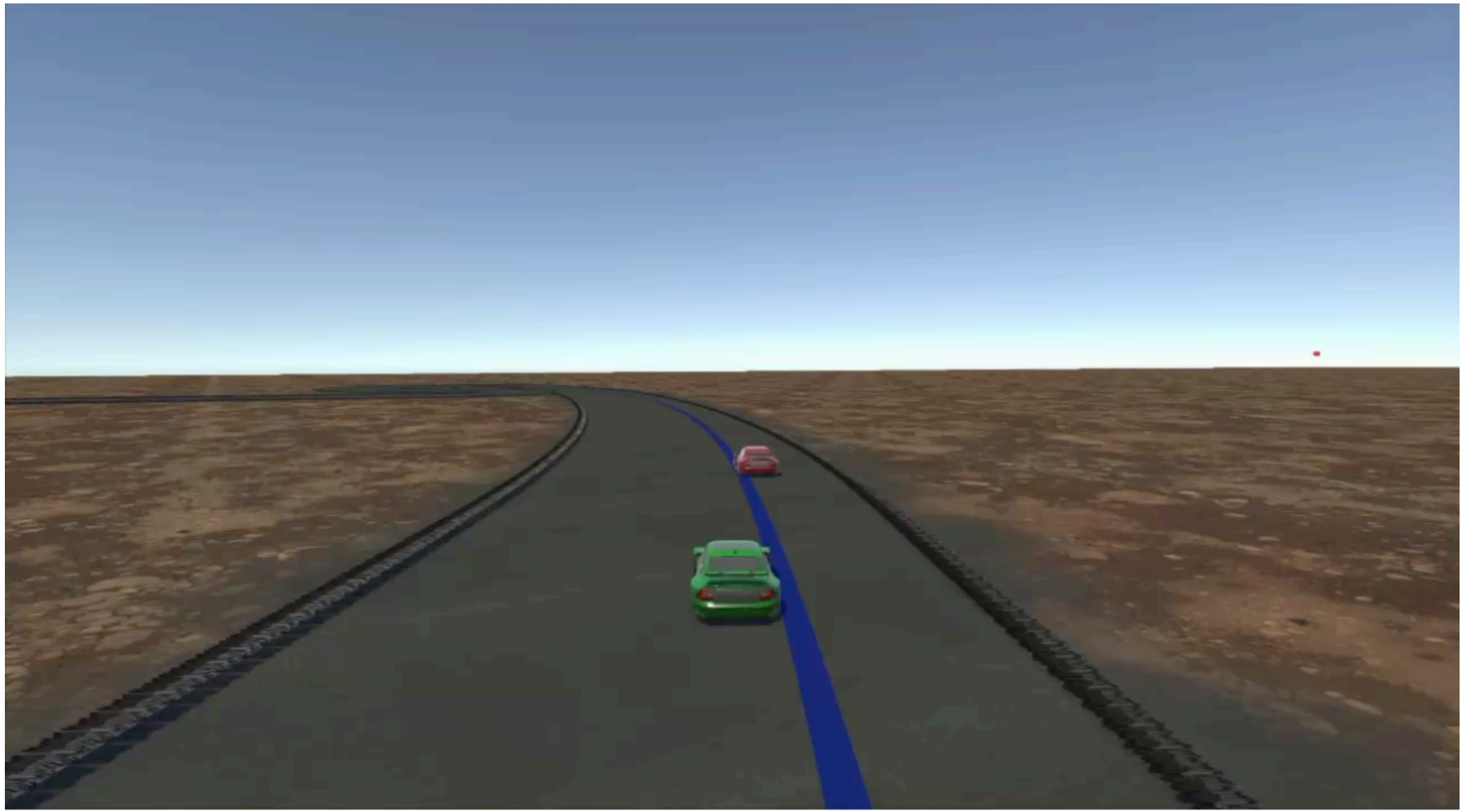
Interaction between decentralized RL agents

Designing strategies in multi-car racing

Deep Multi-agent RL for general-sum games

Approach	Markov Near-Potential Functions

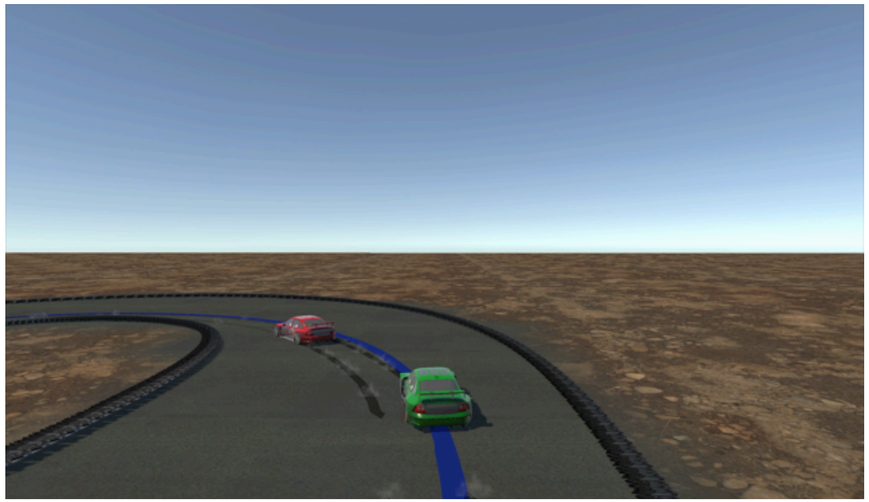
α-RACER: Real-time Algorithms for Autonomous Multi-Car Racing



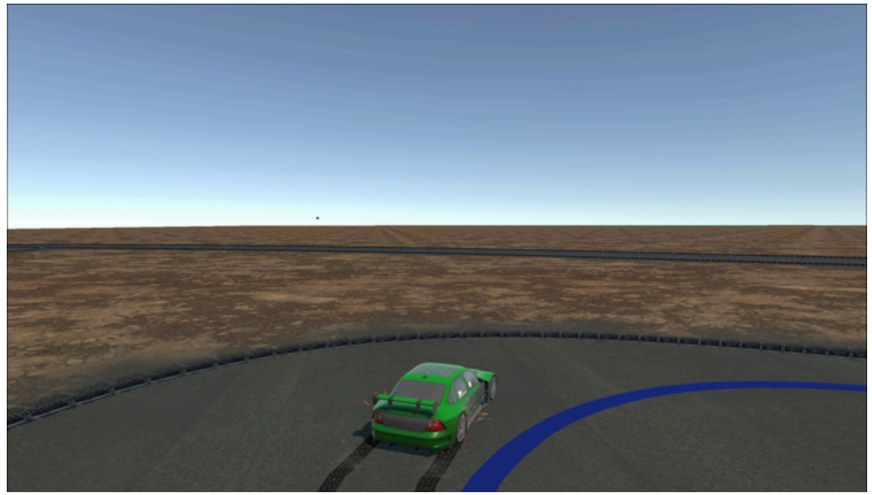
C. Maheshwari*, D. Kalaria*, S. Sastry. Real-Time Algorithms for Game-Theoretic Motion Planning and Control in Autonomous Racing using Near-Potential Function (L4DC 2025)

Autonomous Racing is a becoming a popular testbed to design real-time competitive strategies while operating vehicle at its limits

INDY AUTONOMOUS CHALLENGE



Overtake



Defend

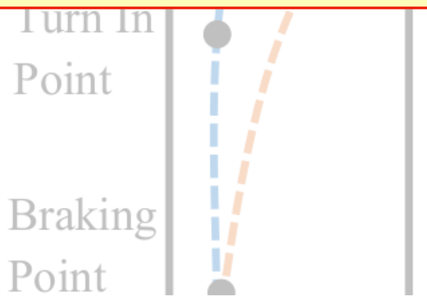
Optimizing Racing Lines for Single Car

- How to go fastest around the track while maintaining safety is a well studied problem

Key Question

How to adapt the optimal race-line in real-time to compute high-performing trajectories for autonomous car?

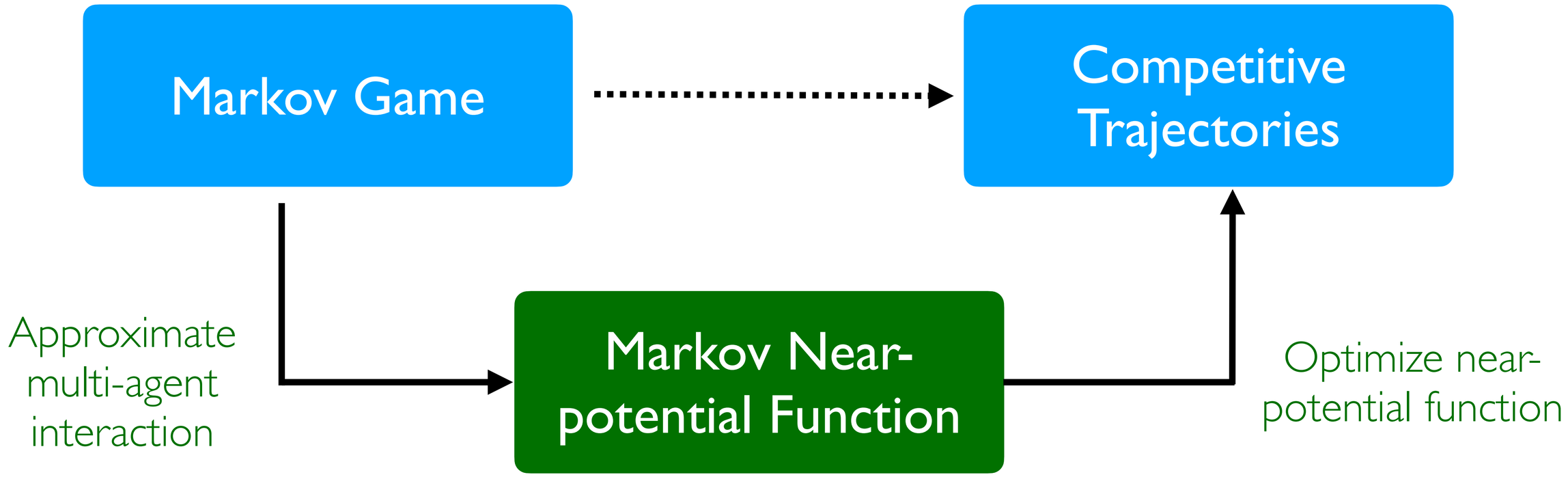
[Minimum Time Problems]
Optimal Control with non-linear vehicle model



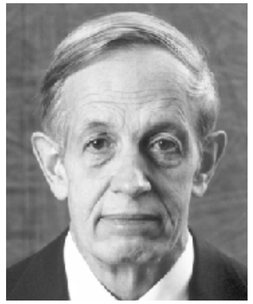
Betz J, Zheng H, Liniger A, Rosolia U, Karle P, Behl M, Krovi V, Mangharam R. Autonomous vehicles on the edge: A survey on autonomous vehicle racing. IEEE Open Journal of Intelligent Transportation Systems. 2022

Contributions

- Markov game with **novel policy parametrization** that enable competitive maneuvers
- Compute competitive trajectories by optimizing **near-potential function**
- **Testing on 3 car** racing that beats many baselines in simulations



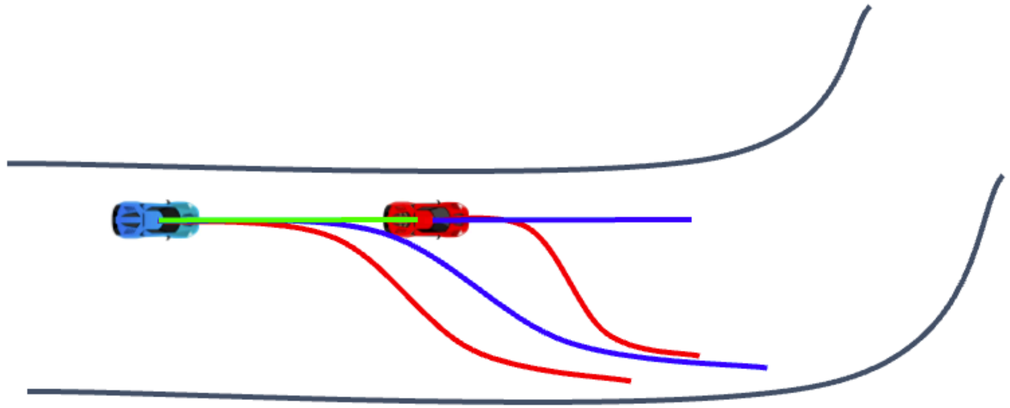
Outcome
Nash equilibrium



A policy θ^* is ϵ-Nash equilibrium if

$$V_i(\mu, \theta_i^*, \theta_{-i}^*) \geq V_i(\mu, \theta_i, \theta_{-i}^*) - \epsilon$$

$$\forall\ i \in I, \theta_i \in \Theta_i, \mu \in \Delta(S)$$

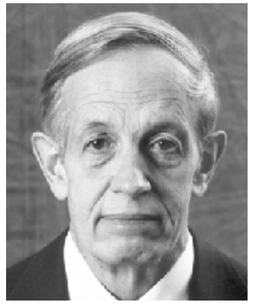


Car 2 

Car 1 

(V_1, V_2)	Defending	No Defending
Overtaking		
No Overtaking		

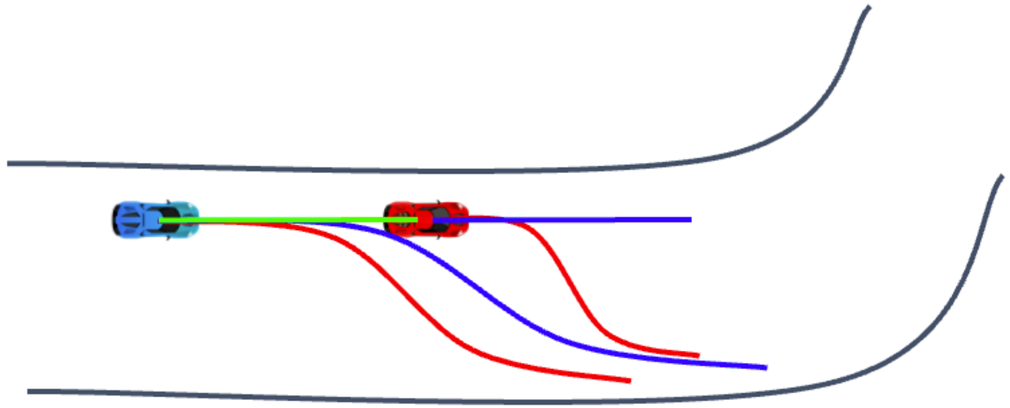
Outcome
Nash equilibrium



A policy θ^* is ϵ-Nash equilibrium if

$$V_i(\mu, \theta_i^*, \theta_{-i}^*) \geq V_i(\mu, \theta_i, \theta_{-i}^*) - \epsilon$$

$$\forall\ i \in I, \theta_i \in \Theta_i, \mu \in \Delta(S)$$



Car 2 

Car 1 

(V_1, V_2)	Defending	No Defending
Overtaking	(5, 5)	(10, 0)
No Overtaking	(0, 10)	(0, 20)

Computing Nash is hard

Can we approximate it?

Markov Near-potential Function

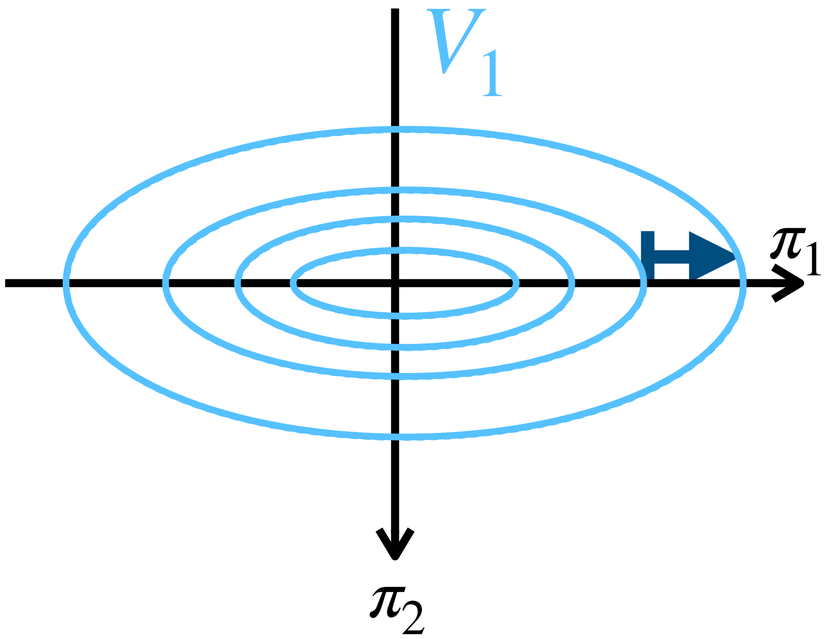
A function $\Phi : S \times \Pi \to \mathbb{R}$ is called a **Markov Near Potential Function (MNPF)** for game G with **closeness parameter** κ if for all $\pi_i, \pi_i' \in \Pi_i, \pi_{-i} \in \Pi_{-i}$

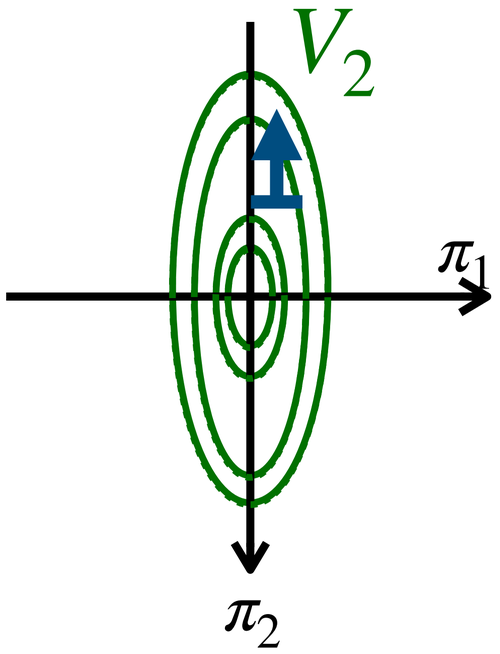
$$\left| \Phi(\mu, \pi_i', \pi_{-i}) - \Phi(\mu, \pi_i, \pi_{-i}) - (V_i(\mu, \pi_i', \pi_{-i}) - V_i(\mu, \pi_i, \pi_{-i})) \right| \leq \alpha$$

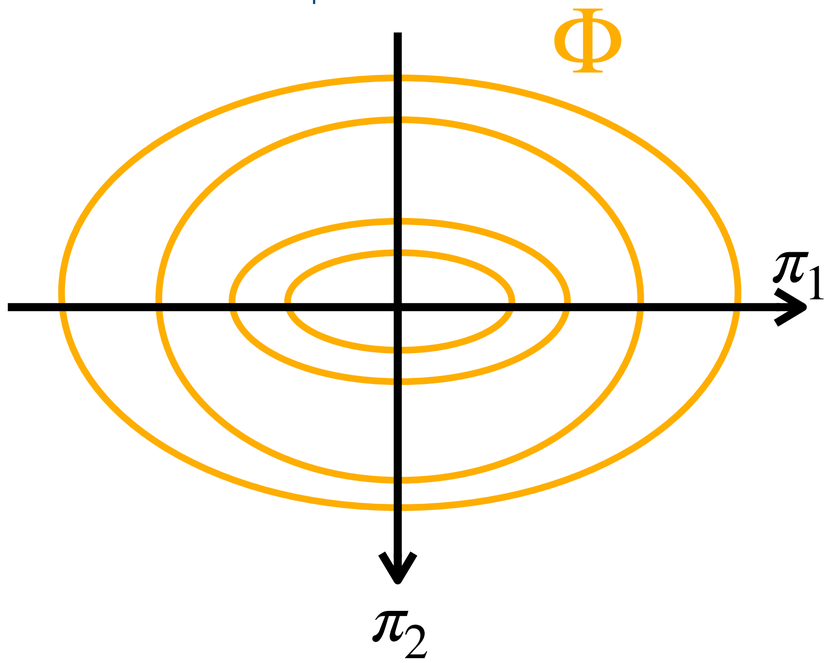
Change in potential due to unilateral shift

Change in value due to unilateral shift

Closeness parameter







X. Guo, X. Li*, **C. Maheshwari***, S. Sastry, M. Wu. Markov α-Potential Games. IEEE TAC 2025

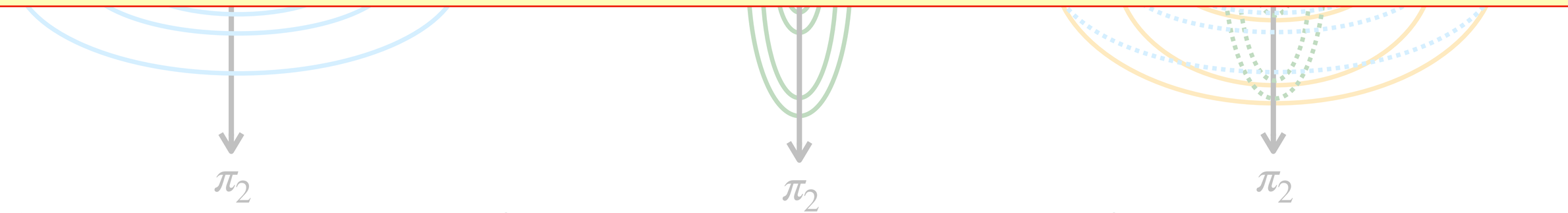
C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

Markov Near-potential Function

A function $\Phi : S \times \Pi \to \mathbb{R}$ is called a Markov Near Potential Function (MNPF) for game G with closeness parameter κ if for all $\pi_i, \pi_i' \in \Pi_i, \pi_{-i} \in \Pi_{-i}$

Structural Properties

- For any game, the maximizer of near-potential function yields $\alpha-$approximate Nash eq

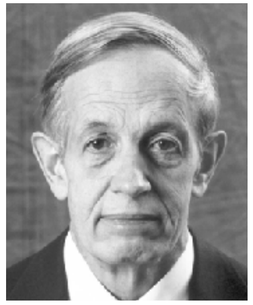


- Purely cooperative scenarios (or generally Markov potential games) are special case with $\kappa = 0$

X. Guo, X. Li*, **C. Maheshwari***, S. Sastry, M. Wu. Markov α-Potential Games. IEEE TAC 2025

C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

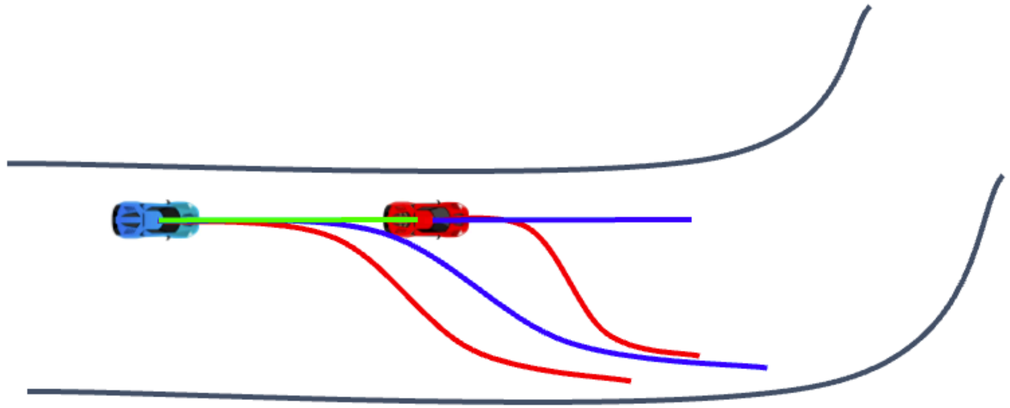
Outcome
Nash equilibrium



A policy θ^* is ϵ-Nash equilibrium if

$$V_i(\mu, \theta_i^*, \theta_{-i}^*) \geq V_i(\mu, \theta_i, \theta_{-i}^*) - \epsilon$$

$$\forall\ i \in I, \theta_i \in \Theta_i, \mu \in \Delta(S)$$



Car 2 

Car 1 

(V_1, V_2)	Defending	No Defending
Overtaking	(5, 5)	(10, 0)
No Overtaking	(0, 20)	(0, 10)

		Φ
Overtaking	Defending	15
Overtaking	No defending	10
No overtaking	Defending	10
No overtaking	No defending	0

Multi-agent Autonomous Racing as a Dynamic Game

- **[Set of player]** Each car is assumed to be competing for winning itself

- **[Set of states]** The state of i^{th} car is defined as

$$\mathbf{x}^i = (p_x^i, p_y^i, \phi^i, v_x^i, v_y^i, \omega^i)$$

Angular velocity

Position in Frenet frame

Orientation

Velocity in Frenet frame

- **[Set of actions]** The action of i^{th} car is defined as

$$\mathbf{u}^i = (d^i, \delta^i)$$

Throttle

Steering

Multi-agent Autonomous Racing as a Dynamic Game

- **[Dynamics]** We use **dynamic car model** that accounts for non-linear tire forces

$$\begin{bmatrix} p^i_{x,t+1} \\ p^i_{y,t+1} \\ \phi^i_{t+1} \\ \tilde{v}^i_{x,t+1} \\ \tilde{v}^i_{y,t+1} \\ \omega^i_{t+1} \end{bmatrix} = \begin{bmatrix} p^i_{x,t} \\ p^i_{y,t} \\ \phi^i_{t} \\ v^i_{x,t} \\ v^i_{y,t} \\ \omega^i_{t} \end{bmatrix} + \Delta t \begin{bmatrix} v^i_{x,t} \\ v^i_{y,t} \\ \omega^i_t - \frac{\kappa^i_t}{(1-\kappa^i_t p^i_{y,t})}(\tilde{v}^i_{x,t}\cos(\phi^i_t) - \tilde{v}^i_{y,t}\sin(\phi^i_t)) \\ \frac{1}{m^i}(F^i_{r,x,t} - F^i_{f,y,t}\sin(\delta^i_t) + m^i \tilde{v}^i_{y,t}\omega^i_t) \\ \frac{1}{m^i}(F^i_{r,y,t} + F^i_{f,y,t}\cos(\delta^i_t) - m^i \tilde{v}^i_{x,t}\omega^i_t) \\ \frac{1}{I^i_z}(F^i_{f,y,t} l^i_f \cos(\delta^i_t) - F^i_{r,y,t} l^i_r) \end{bmatrix}$$

- **[One-step Utility Function]** Maximize its relative progress on the track compared to previous time step

$$r^i(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) = \underbrace{(p^i_{x,t+1} - \max_{j \neq i} p^j_{x,t+1})}_{\text{Progress relative to others at time t+1}} - \underbrace{(p^i_{x,t} - \max_{j \neq i} p^j_{x,t})}_{\text{Progress relative to others at time t}}$$

Novel Policy Parameterization

- Model Predictive Control (MPC) to track **competitive reference trajectory**

Reference Trajectory

Control variation

$$\min_{(\mathbf{x}_t^{i,k})_{k=1}^{K},(\mathbf{u}_t^{i,k})_{k=0}^{K-1}} \sum_{k=1}^{K} \left\| \begin{pmatrix} p_{x,t}^{i,k} - p_{x,t}^{\text{ref},i,k} \\ p_{y,t}^{i,k} - p_{y,t}^{\text{ref},i,k} \end{pmatrix} \right\|_{\mathbf{Q}}^2 + \sum_{k=1}^{K-1} \left\| \begin{pmatrix} d_t^{i,k} - d_t^{i,k-1} \\ \delta_t^{i,k} - \delta_t^{i,k-1} \end{pmatrix} \right\|_{\mathbf{R}}^2$$

Weight Matrices

$$\text{s.t. } \mathbf{x}_t^{i,k+1} = f^i(\mathbf{x}_t^k, \mathbf{u}_t^{i,k})$$
$$\mathbf{x}_t^{i,0} = \mathbf{x}_t^i$$

Dynamics

$$\Delta\delta_{\min}^i \leq \delta_t^{i,k} - \delta_t^{i,k-1} \leq \Delta\delta_{\max}^i, \quad \forall k = 0, 1, ..., K-1$$
$$d_{\min}^i \leq d_t^{i,k} \leq d_{\max}^i, \quad \forall k = 0, 1, ..., K-1$$

Input Constraints

$$|p_{y,t}^{i,k}| \leq w_{\max}, \quad \forall k = 1, ..., K$$

Track Constraints

$$|p_{y,t}^{i,k} - p_{y,t}^{j,k}| \geq p_y^{\min}, \quad \forall k = 1, ..., K, \forall j \neq i$$
$$|p_{x,t}^{i,k} - p_{x,t}^{j,k}| \geq p_x^{\min}, \quad \forall k = 1, ..., K, \forall j \neq i,$$

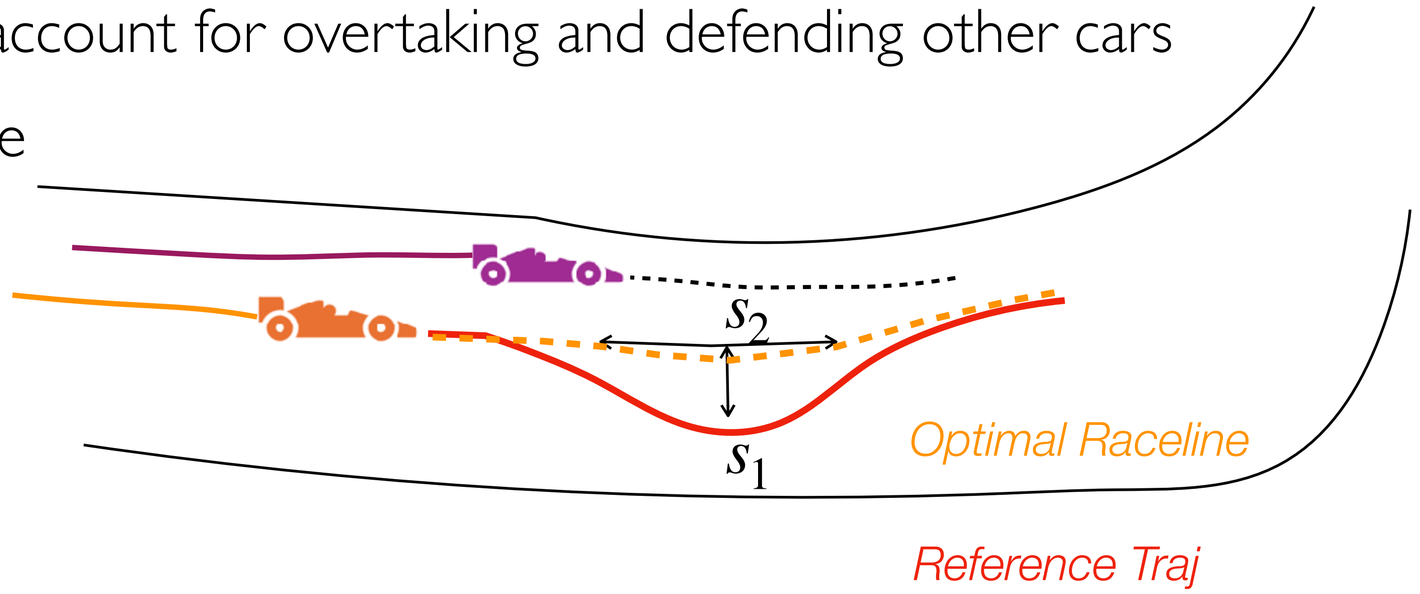
Collision Avoidance Constraints
(interpolated trajectory of other cars)

Design of Reference Trajectory

- The optimal race line is adjusted to account for overtaking and defending other cars

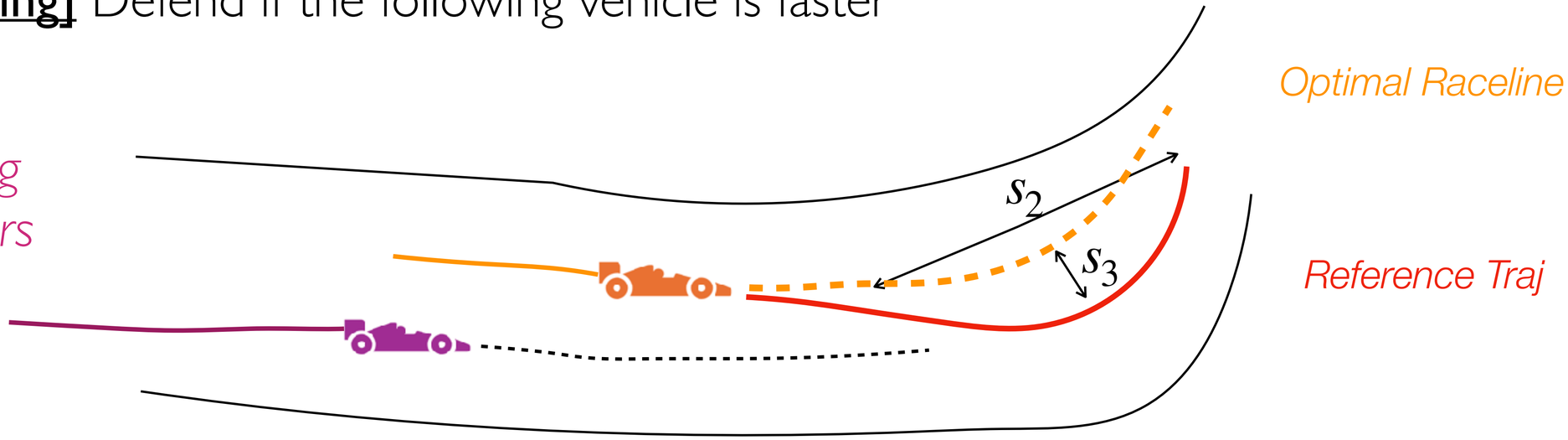
[Overtaking] Overtake from the side

Overtaking Parameters



[Defending] Defend if the following vehicle is faster

Defending parameters



Design of Reference Trajectory

- The optimal race line is adjusted to account for overtaking and defending other cars

  [Overtaking] Overtake from the side

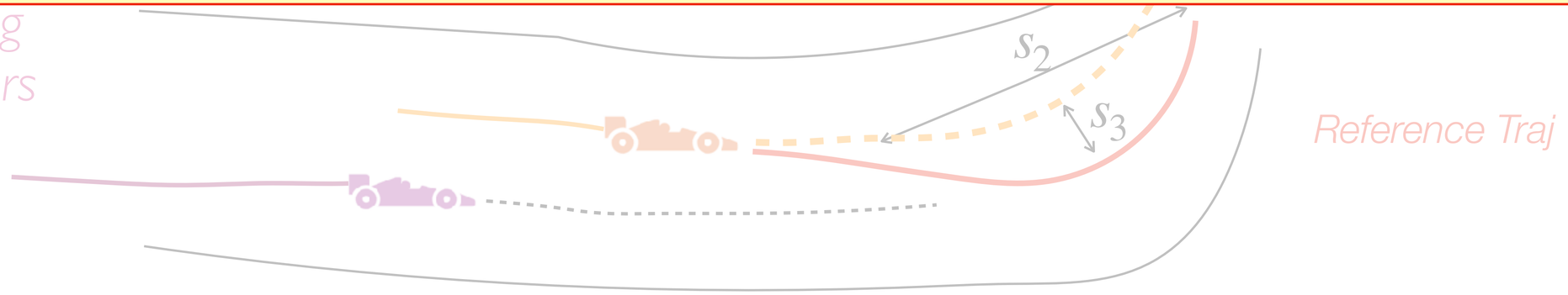
Policy Parametrization

$$\theta = (Q, R, s_1, s_2, s_3) \longrightarrow \text{MPC Policy}$$

MPC Objective Parameters

Overtaking and Defending Parameters

Defending parameters



Algorithmic Approach

[Offline Phase] Learning a nearest-potential function through simulated race data

Approximation Parameter

$$\min_{\alpha,\phi} \alpha$$

Parametrized potential function

Parametrized Value function (computed through simulated races)

$$\text{s.t. } |\Phi(\mathbf{x}, \theta_i, \theta_{-i}; \phi) - \Phi(\mathbf{x}, \theta_i', \theta_{-i}; \phi) - (V_i(\mathbf{x}, \theta_i, \theta_{-i}; \nu) - V_i(\mathbf{x}, \theta_i', \theta_{-i}); \nu)) | \leq \alpha$$

Change in potential due to unilateral shift

Change in value due to unilateral shift

$$\forall\ i \in I, \theta_i \in \Theta_i, \theta_{-i} \in \Theta_{-i}, \mathbf{x} \in \mathcal{X}$$

[Online Phase] Optimize the near-potential function

Computed Offline

$$\theta^* \in \arg\max_{\theta \in \Theta} \Phi(\mathbf{x}_t, \theta; \phi^*) \qquad \mathbf{u}_t^i = \pi^i(\mathbf{x}_t; \theta^{*,i})$$

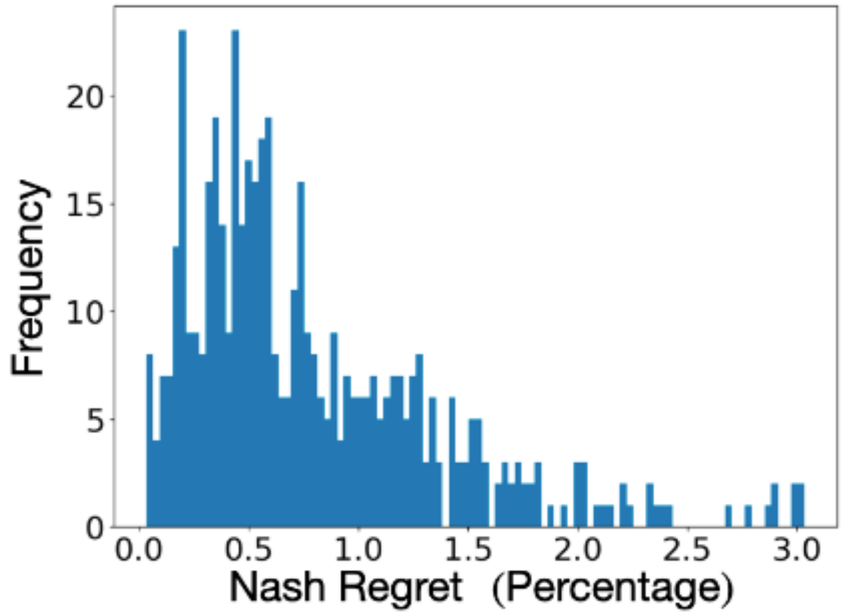
Policy parameter

Current state

Numerical Performance: 3 Car Racing

- **[Offline Training]** We generate simulated race data from 4000 races with randomly chosen policy parameters

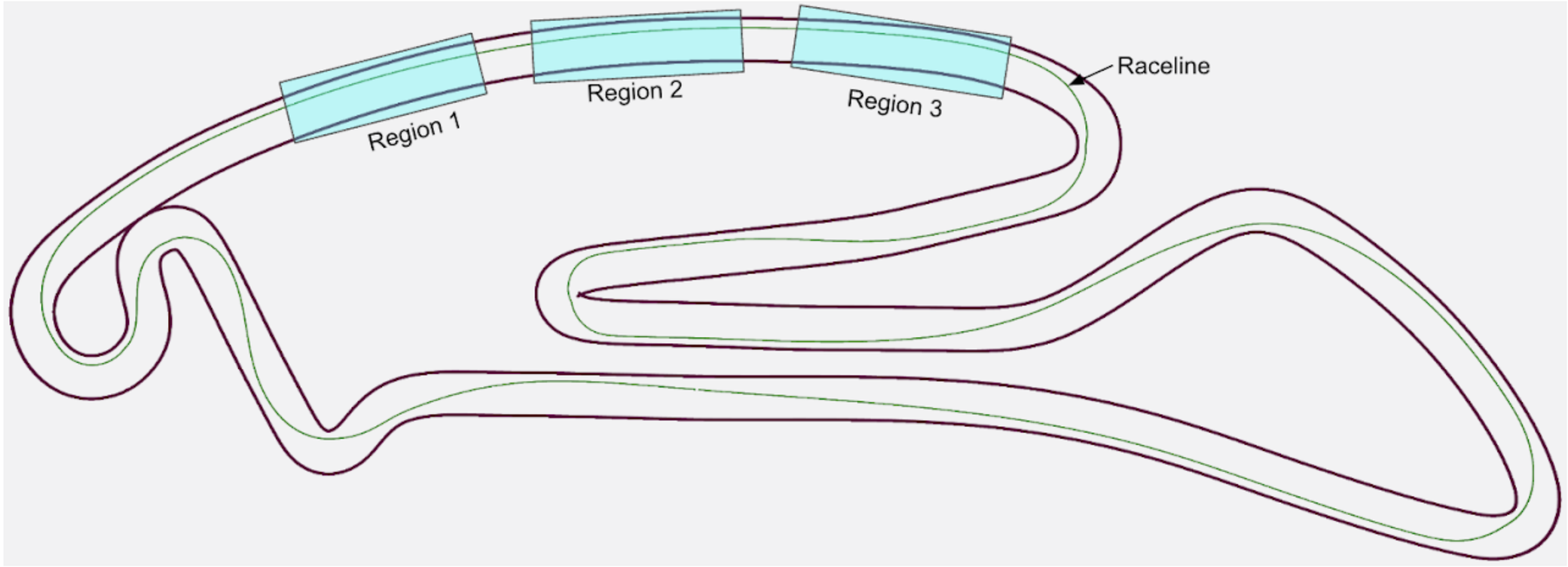
- **[Online Optimization of Potential Function]** We maximize the potential function online



Potential function maximization results
in Nash Regret within 3%

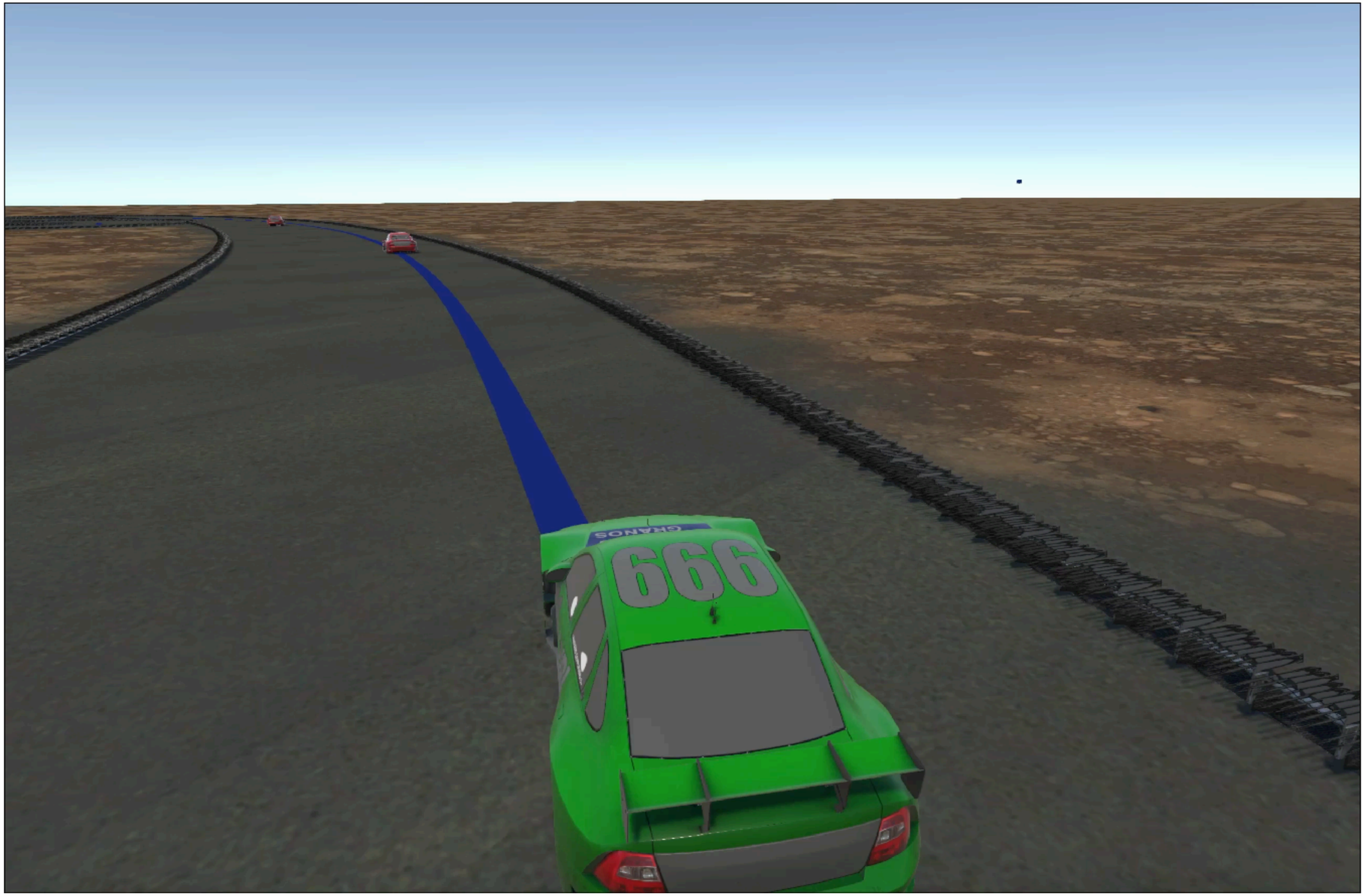
Comparison with Baselines

- The starting order of agents is Ego>O_1>O_2, O_1>Ego>O_2, O_1>O_2> Ego for 33 races each



Racing Scenario	# wins (Ours)	# wins (O_1)	# wins (O_2)
Opponent use Iterated Best Response	73	22	4
Opponent use self-play RL	91	7	1

Race with self-play RL



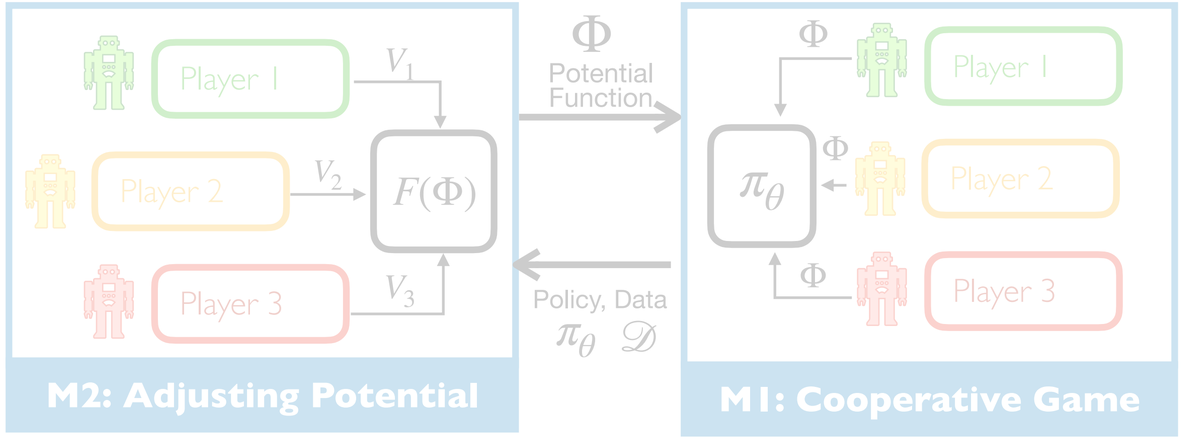
Today's Talk

Challenge | Strategic interaction in dynamic multi-agent environment

Interaction between decentralized RL agents

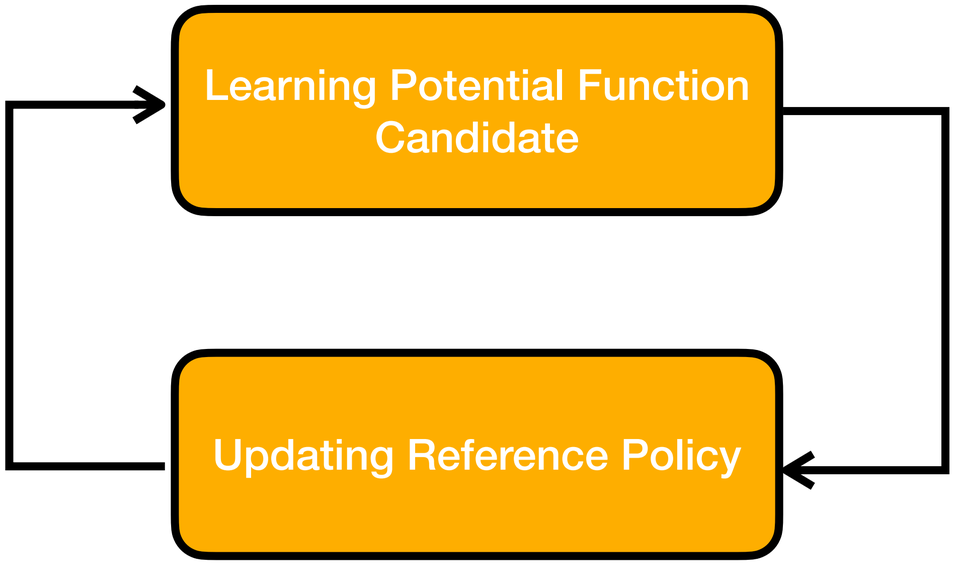
Designing strategies in multi-car racing

Deep Multi-agent RL for general-sum games



Approach | Markov Near-Potential Functions

Near-Potential Policy Optimization for General-sum Multi-agent Reinforcement Learning



A. Kalanther, S. Bharvirkar, S. Sastry, **C. Maheshwari.** Near-Potential Policy Optimization for General-sum Multi-agent Reinforcement Learning
Working paper: https://arxiv.org/abs/2603.06977

Multi-agent Reinforcement Learning

- MARL is a popular training pipeline that allows us to train autonomous agents in dynamic multi-agent environments

- Popular MARL methods:

  MAPPO / HAPPO: Designed for common-interest games

  IPPO: Commonly works well for adversarial games

  Self-play RL: Zero-sum games with identical roles or symmetric players

Can we develop a MARL algorithm for general-sum games with heterogeneous players?

Contributions

- Leveraging the **near-potential structure** to develop a deep MARL pipeline that can be used in **general-sum dynamic games**

- Compare performance against state of the art MARL algorithms

A. Kalanther, S. Bharvirkar, S. Sastry, **C. Maheshwari.** Near-Potential Policy Optimization for General-sum Multi-agent Reinforcement Learning
Working paper: https://arxiv.org/abs/2603.06977

Markov Near-potential Function

A function $\Phi : S \times \Pi \to \mathbb{R}$ is called a **Markov Near Potential Function (MNPF)** for game G with **closeness parameter** κ if for all $\pi_i, \pi_i' \in \Pi_i, \pi_{-i} \in \Pi_{-i}$

$$\left| \Phi(\mu, \pi_i', \pi_{-i}) - \Phi(\mu, \pi_i, \pi_{-i}) - (V_i(\mu, \pi_i', \pi_{-i}) - V_i(\mu, \pi_i, \pi_{-i})) \right| \leq \alpha$$

Change in potential due to unilateral shift

Change in value due to unilateral shift

Closeness parameter

Structural Properties

- For any game, the **maximizer of near-potential function** yields $\alpha-$approximate Nash eq

Given a game can we learn a near potential function Φ which yields $\alpha-$Nash equilibrium with small value of α?

X. Guo, X. Li*, **C. Maheshwari***, S. Sastry, M. Wu. Markov α-Potential Games. IEEE TAC 2025

C. Maheshwari, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024

Proxy MARL Objective

A function $\Phi : S \times \Pi \to \mathbb{R}$ is called a **Markov Near Potential Function (MNPF)** for game G with **closeness parameter** κ if for all $\pi_i, \pi_i' \in \Pi_i, \pi_{-i} \in \Pi_{-i}, i \in I$

$$|\Phi(\mu, \pi_i', \pi_{-i}) - \Phi(\mu, \pi_i, \pi_{-i}) - (V_i(\mu, \pi_i', \pi_{-i}) - V_i(\mu, \pi_i, \pi_{-i}))| \leq \alpha$$

Change in potential due to unilateral shift

Change in value due to unilateral shift

Closeness parameter

Converting it to an optimization problem

$$\min_{\Phi} \max_{i} \max_{\pi_i, \pi_i', \pi_{-i}} |(\Phi(\mu, \pi_i, \pi_{-i}) - \Phi(\mu, \pi_i', \pi_{-i})) - (V_i(\mu, \pi_i, \pi_{-i}) - V_i(\mu, \pi_i', \pi_{-i}))|$$

Key insight: To approximate Nash we only need to approximate potential near a suitable "reference policy"

(Alphabetical, *Corresponding authors)X. Guo, X. Li*, **C. Maheshwari***, M. Wu, S. Sastry. Sastry. *Markov $\alpha-$potential Games*. IEEE TAC 2025

Proxy MARL Objective

- Consider a potential function candidate Φ

$\pi^{*,\Phi}$ is a Nash-equilibrium of cooperative game with value function as Φ

$\pi_i^{\dagger,\Phi}$ is a best-response of player i when opponents play $\pi_{-i}^{*,\Phi}$

Proxy Objective

$$F_i(\Phi) = \underbrace{\Phi(\pi^{*,\Phi}) - \Phi(\pi_i^{\dagger,\Phi}, \pi_{-i}^{*,\Phi})}_{\text{Change in potential due to unilateral shift}} - \underbrace{V_i(\pi^{*,\Phi}) - V_i(\pi_i^{\dagger,\Phi}, \pi_{-i}^{*,\Phi})}_{\text{Change in value due to unilateral shift}}$$

Proposition

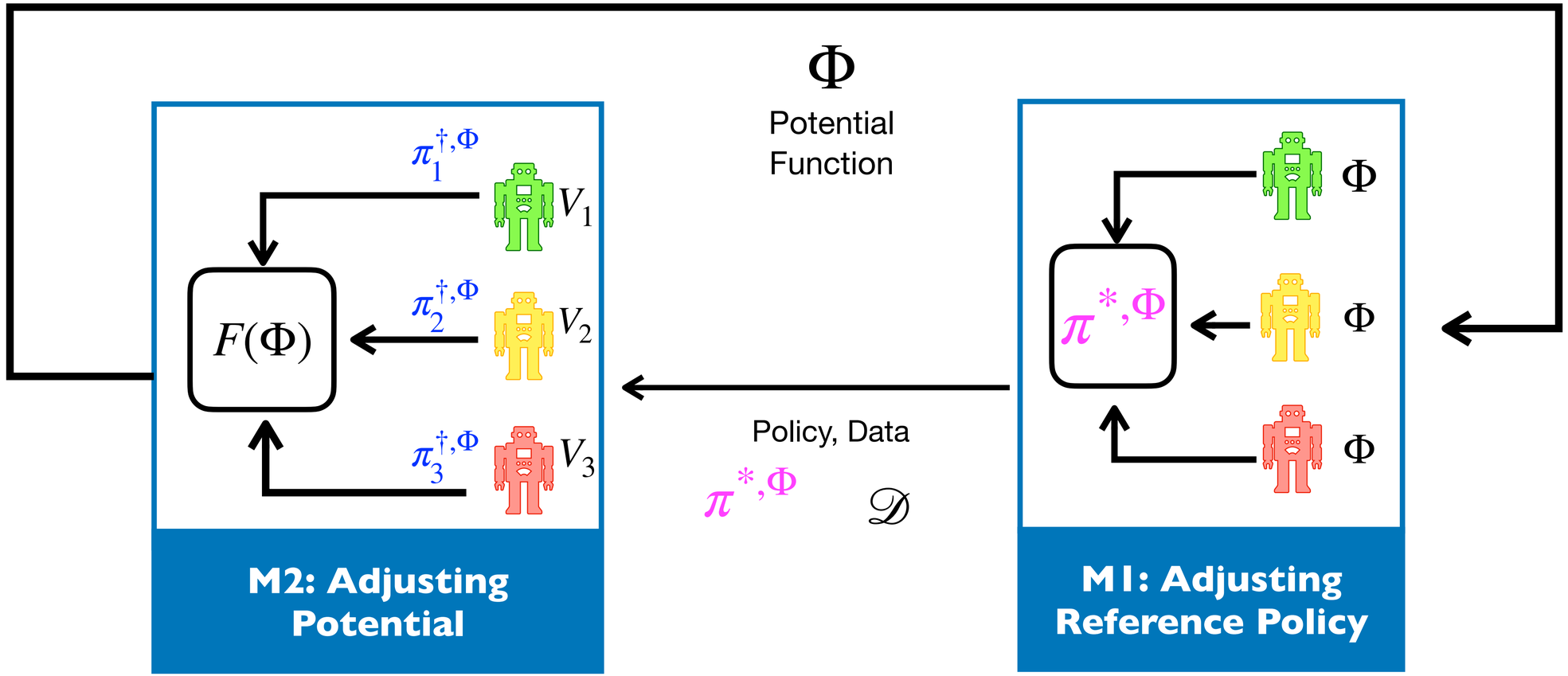
Suppose that, for some Φ, $F_i(\Phi) \leq \alpha$ for every player i. Then $\pi^{*,\Phi}$ is a an $\alpha-$approximate Nash eq.

A. Kalanther, S. Bharvirkar, S. Sastry, **C. Maheshwari.** Near-Potential Policy Optimization for General-sum Multi-agent Reinforcement Learning
Working paper: https://arxiv.org/abs/2603.06977

Overview of MARL Training Pipeline

$$F_i(\Phi) = \Phi(\pi^{*,\Phi}) - \Phi(\pi_i^{\dagger,\Phi}, \pi_{-i}^{*,\Phi}) - V_i(\pi^{*,\Phi}) - V_i(\pi_i^{\dagger,\Phi}, \pi_{-i}^{*,\Phi})$$

- Our MARL training pipeline is based on solving $\min_{\Phi} \max_{i \in \mathcal{N}} F_i(\Phi)$



MARL Training Pipeline

$$F_i(\Phi) = \Phi(\pi^{*,\Phi}) - \Phi(\pi_i^{\dagger,\Phi}, \pi_{-i}^{*,\Phi}) - V_i(\pi^{*,\Phi}) - V_i(\pi_i^{\dagger,\Phi}, \pi_{-i}^{*,\Phi})$$

- Our MARL training pipeline is based on solving $\min_{\Phi} \max_{i \in \mathcal{N}} F_i(\Phi)$

- Since this is a non-smooth optimization, we consider the **log-sum exp smoothing**

$$\min_{\Phi} \max_{i \in \mathcal{N}} F_i(\Phi) \quad \Longrightarrow \quad \min_{\Phi} F_\beta(\Phi) = \frac{1}{\beta} \log\Big(\sum_{i \in \mathcal{N}} \exp(\beta F_i(\Phi)) \Big)$$

$$\text{As } \beta \uparrow \infty, F_\beta(\Phi) \to \max_{i \in \mathcal{N}} F_i(\Phi)$$

- For tractability, we consider potential function Φ_w **that is parametrized by** $w \in W$ (e.g. neural network, linear parameterization, etc)

MARL Training Pipeline: Gradient Estimator

- We develop a **gradient-based scheme** to solve $\min_{w \in W} F_\beta(\Phi_w)$

- Computing the **(first-order)gradient is challenging** due to the structure of $F_\beta(\Phi_w)$

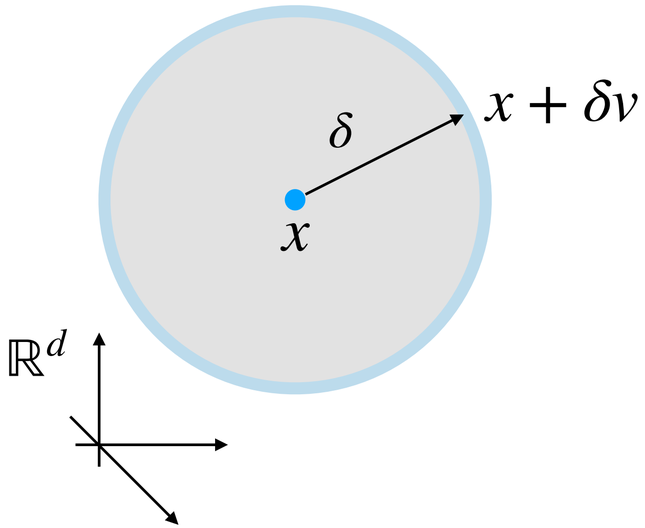
$$F_i(\Phi_w) = \Phi_w(\pi^{*,\Phi_w}) - \Phi_w(\pi_i^{\dagger,\Phi_w}, \pi_{-i}^{*,\Phi_w}) - V_i(\pi^{*,\Phi_w}) - V_i(\pi_i^{\dagger,\Phi_w}, \pi_{-i}^{*,\Phi_w})$$

Dependence on w is not only due to Φ_w, but also due to policies π^{*,Φ_w} and

Approach: We use **(randomized) zeroth-order gradient estimator**

Randomized Zeroth-order Gradient Estimator

$$L : \mathbb{R}^d \to \mathbb{R}$$



Gradient estimator

$$\hat{\nabla} L(x; v) = \frac{d}{2\delta} \left(L(x + \delta u) - L(x - \delta u) \right) u,$$

where $v \sim \mathcal{S}^{d-1}$

$$\mathbb{E}[\hat{\nabla} L(x; u)] = \frac{L(x + \delta) - L(x - \delta)}{2\delta}$$

1-D case

Properties

1. Biased Estimator

Smoothed-out function

$$\mathbb{E}[\hat{\nabla} L(x; u)] = \nabla L^{(\delta)}(x) = \nabla \mathbb{E}_{u \in \text{Unif}(B^d)}[L(x + \delta u)]$$

2. Bias-Variance trade-off

Bias $\propto \delta$ **Variance** $\propto \frac{1}{\delta^2}$

MARL Training Pipeline: Gradient Estimator

- We develop a gradient-based scheme to solve $\min_{w \in W} F_\beta(\Phi_w)$

- Computing the (first-order)gradient is challenging due to the structure of $F_\beta(\Phi_w)$

$$F_i(\Phi_w) = \Phi_w(\pi^{*,\Phi_w}) - \Phi_w(\pi_i^{\dagger,\Phi_w}, \pi_{-i}^{*,\Phi_w}) - V_i(\pi^{*,\Phi_w}) - V_i(\pi_i^{\dagger,\Phi_w}, \pi_{-i}^{*,\Phi_w})$$

Dependence on w is not only due to Φ_w, but also due to policies π^{*,Φ_w} and $\pi_i^{\dagger,\Phi_w}$

Approach: We use (randomized) zeroth-order gradient estimator

$$\hat{\nabla}_w F_\beta(w;\delta) = \frac{\dim(w)}{2\delta}\left(F_\beta(\Phi_{\hat{w}}) - F_\beta(\Phi_{\check{w}})\right)u,$$

where $u \sim \text{Unif}(\mathcal{S}^{\dim(w)})$, $\quad \hat{w} = w + \delta u, \quad \check{w} = w - \delta u$

MARL Training Pipeline: Evaluating Gradient

$$\hat{\nabla}_w F_\beta(w;\delta) = \frac{\dim(w)}{2\delta}\left(F_\beta(\Phi_{\hat{w}}) - F_\beta(\Phi_{\check{w}})\right)u,$$

- To evaluate $F_\beta(\Phi_w)$, we need to compute estimate of $\pi^{*,\Phi}$ and $\pi^{\dagger,\Phi}$
  - Computing $\pi^{*,\Phi}$: Any cooperative game solver that converges to Nash equilibrium (e.g. MAPPO, HAPPO etc)
  - Computing $\pi^{\dagger,\Phi}$: Any reinforcement learning solver (e.g. PPO, GRPO etc)

MARL Training Pipeline

Algorithm 1 Near-Potential Policy Optimization

Input: The zeroth-order step size parameter $\delta \geq 0$, gradient descent learning rate $\eta \geq 0$, and smoothness parameter $\beta \geq 0$.
Initialize π^{Φ}, $\pi_i^{\dagger}$ for all agents $i \in [n]$, and $w \in \mathcal{W}$
for each iteration **do**
  Sample $u \sim \mathcal{S}(\mathbb{R}^{\dim(w)})$
  Set $\hat{w} = w + \delta u$ and $\check{w} = w - \delta u$
  $\check{\pi}^{\Phi} \leftarrow \textsc{CoopGameSolver}(\pi^{\Phi}, \check{w}, K_1)$
  $\pi^{\Phi} \leftarrow \textsc{CoopGameSolver}(\pi^{\Phi}, \hat{w}, K_1)$
  for each agent $i \in \mathcal{N}$ **do**
    $\check{\pi}_i^{\dagger} \leftarrow \textsc{RLSolver}(\pi_i^{\dagger}, \check{\pi}_{-i}^{\Phi}, r_i, K_2)$
    $\pi_i^{J} \leftarrow \textsc{RLSolver}(\pi_i^{J}, \pi_{-i}^{\Phi}, r_i, K_2)$
  end for
  for each agent $i \in \mathcal{N}$ **do**
    Compute $F_i(\Phi_{\hat{w}})$ and $F_i(\Phi_{\check{w}})$
  end for
  Set $F_\beta(\Phi_{\hat{w}}) = \frac{1}{\beta}\log(\sum_i \exp(\beta F_i(\Phi_{\hat{w}})))$
  Set $F_\beta(\Phi_{\check{w}}) = \frac{1}{\beta}\log(\sum_i \exp(\beta F_i(\Phi_{\check{w}})))$
  $w \leftarrow w - \eta \frac{\dim(w)}{2\delta}\left(F_\beta(\Phi_{\hat{w}}) - F_\beta(\Phi_{\check{w}})\right) u$
end for
return π^{Φ}

Sampling points for zeroth-order estimator

Cooperative game solver (e.g. MAPPO / HAPPO) to evaluate $\pi^{*,\Phi}$

RL solver (e.g. PPO, GRPO etc) to evaluate $\pi^{\dagger,\Phi}$

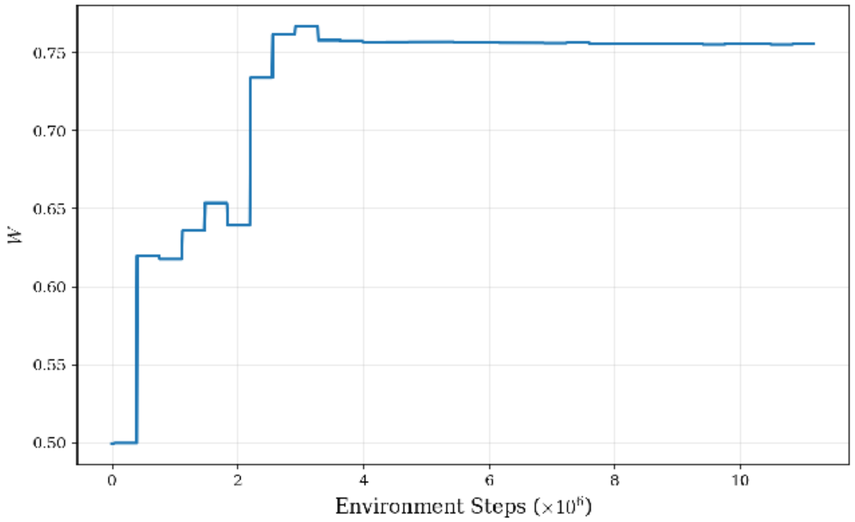
Zeroth order gradient evaluation

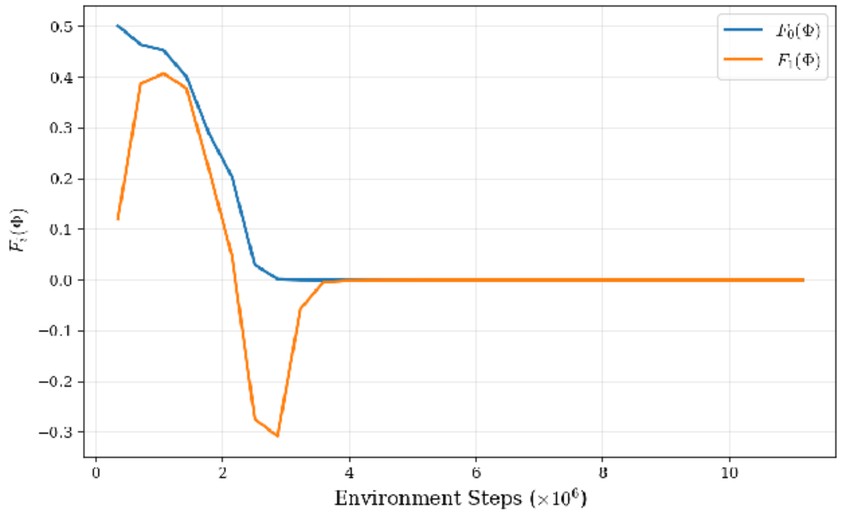
Numerical Evaluation

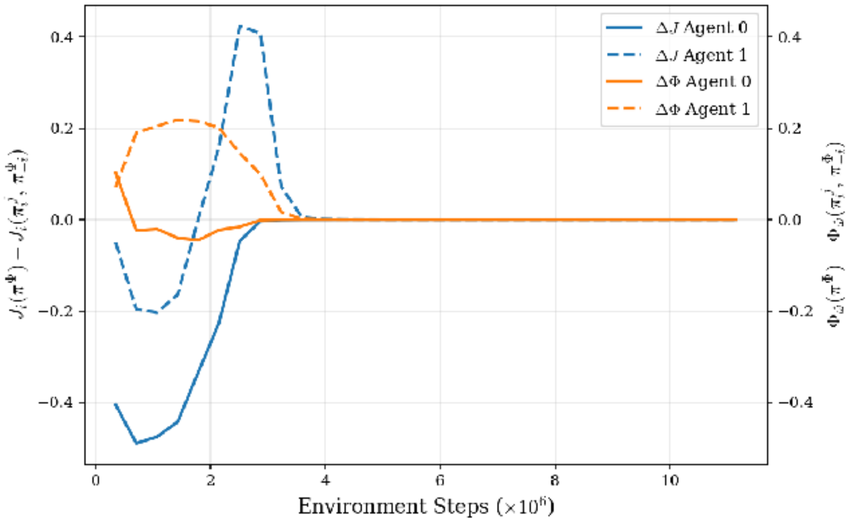
$$\mathcal{G} = \begin{array}{c|cc} & B_1 & B_2 \\ \hline A_1 & (1,\,1) & (1,\,1/2) \\ A_2 & (1/2,\,7/4) & (0,\,2) \end{array}$$

$$\Phi_w = wV_1 + (1-w)V_2 \qquad \Phi_w = \begin{array}{c|cc} & B_1 & B_2 \\ \hline A_1 & 1 & (1+w)/2 \\ A_2 & (7-5w)/4 & 2(1-w) \end{array}$$

- Note that this is not a potential game







Numerical Evaluation

$\alpha \cdot$

	B_1	B_2
A_1	$(1, -1)$	$(-1, 1)$
A_2	$(-1, 1)$	$(1, -1)$

$+$

$(1-\alpha) \cdot$

	B_1	B_2
A_1	$(1, 1)$	$(1, 1/2)$
A_2	$(1/2, 7/4)$	$(0, 2)$

- Here, we consider a parametrization of potential function as $\Phi_w(x, y) = -(x - w_1)^2 - (y - w_2)^2$

Regret values (lower the better)

α	NePPO	IPPO	MAPPO
0.0	**0.000**	**0.000**	0.500
0.2	**0.000**	**0.000**	0.800
0.4	**0.036**	DNC	1.100
0.6	**0.052**	DNC	1.400
0.8	**0.034**	DNC	1.700
1.0	**0.000**	0.011	N/A

Zero-sum game

Numerical Evaluation

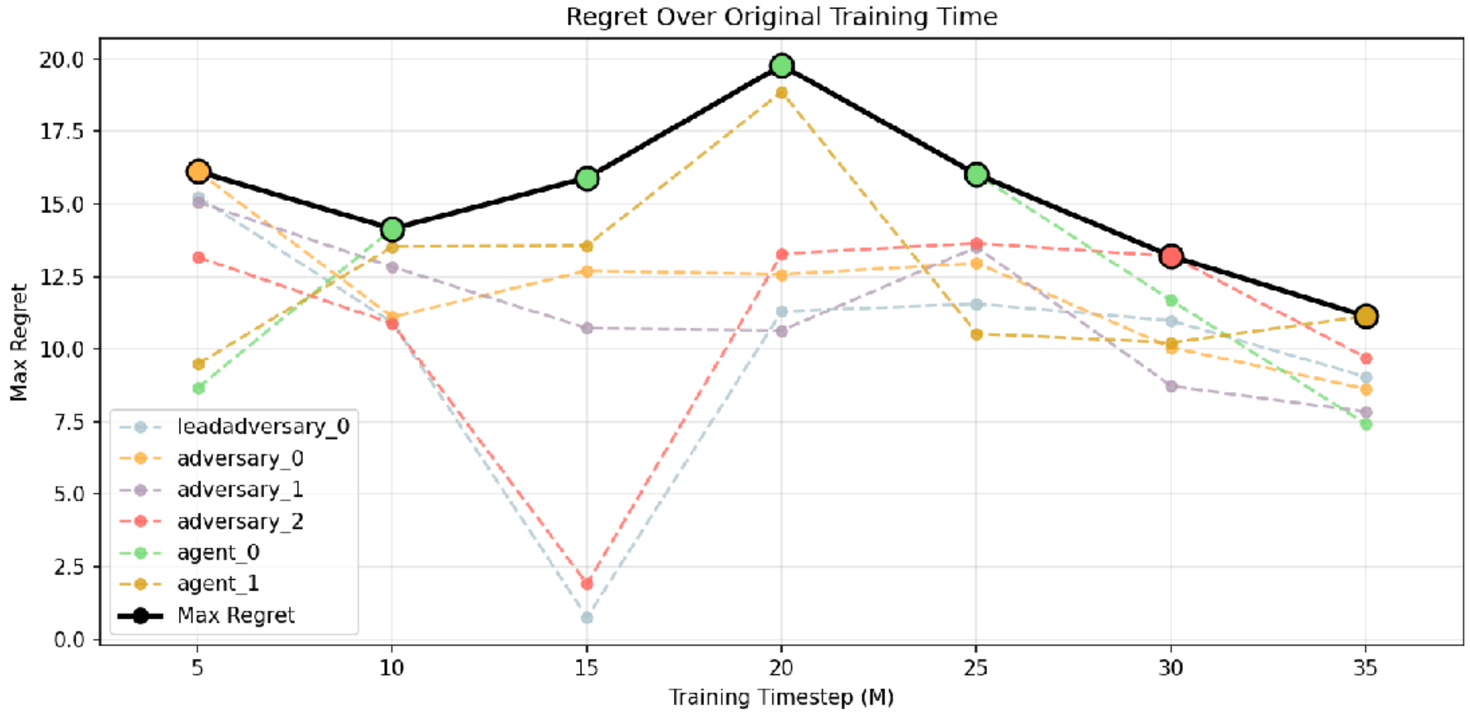
- Consider the simple_world_comm environment from MPE benchmark

$$\Phi_w(\pi) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k \phi_w(s_k, a_k) \right]$$

one-layer NN

$$\phi_w(s_t, a_t) = \sum_{i \in \mathcal{N}} \sigma_i(W s_t + b)\, r_i(s_t, a_t)$$

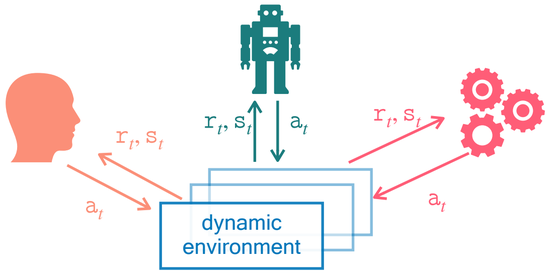
Regret values (lower the better)	NePPO	IPPO	MAPPO
Simple World Comm	**11.14**	23.90	51.78



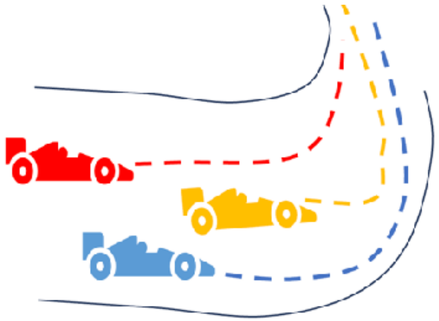
Today's Talk

Challenge — Strategic interaction in dynamic multi-agent environment

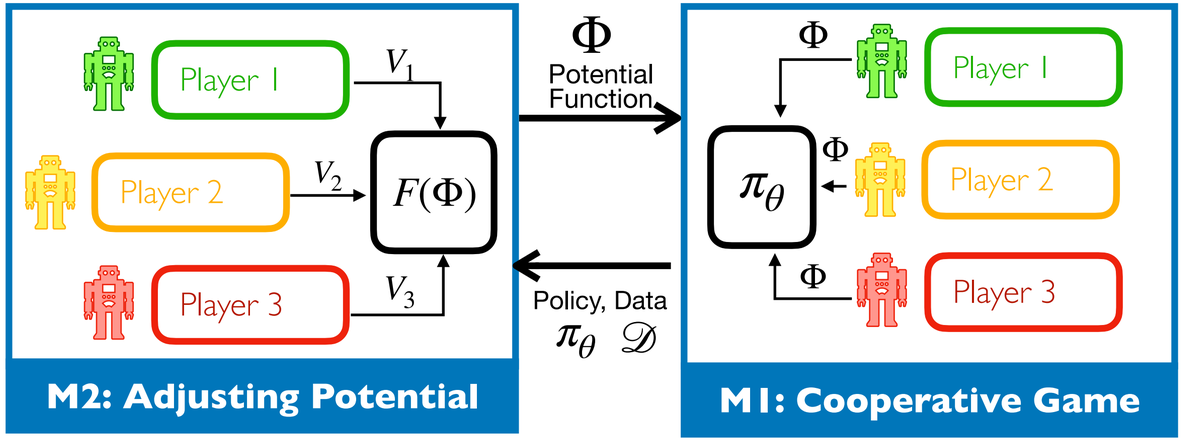
Interaction between decentralized RL agents



Designing strategies in multi-car racing



Deep Multi-agent RL for general-sum games



Approach — Markov Near-Potential Functions

References

- **C. Maheshwari**, M. Wu, S. Sastry. Convergence of Decentralized Actor-Critic Algorithm in General-sum Markov Games. IEEE LCSS 2024
- **C. Maheshwari**, M. Wu, D. Pai, S. Sastry. Independent and Decentralized Learning in Markov Potential Games. IEEE TAC 2025
- X. Guo, X. Li*, **C. Maheshwari***, S. Sastry, M. Wu. Markov α-Potential Games. IEEE TAC 2025
- D. Kalaria, **C. Maheshwari,** S. Sastry. Real-Time Algorithms for Game-Theoretic Motion Planning and Control in Autonomous Racing using Near-Potential Function. L4DC 2025
- A. Kalanther, S. Bharvirkar, S. Sastry, **C. Maheshwari.** Near-Potential Policy Optimization for General-sum Multi-agent Reinforcement Learning 2026. Working paper: https://arxiv.org/abs/2603.06977

Thanks to Collaborators

- Shankar Sastry
- Manxi Wu
- Xin Guo
- Dvij Kalaria
- Addison Kalanther
- Sanika Bharvirkar
- Druv Pai
- Xinyu Li

Website / Email

Email
chinmay_maheshwari@jhu.edu