COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Ditech

A Hybrid Framework of Reinforcement Learning and Physics-Informed Deep Learning for Spatiotemporal Mean Field Games

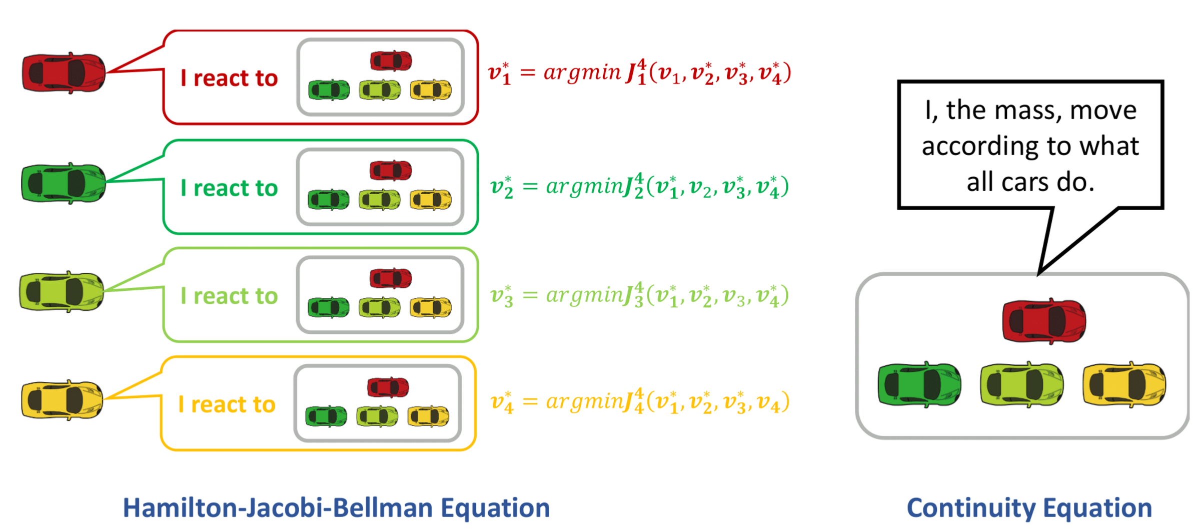
Xu Chen, Shuo Liu, Sharon Di
Columbia University

Introduction

How to model autonomous vehicle (AV) control strategy and traffic flow?

Assumptions:

- AVs observe global traffic information
- AVs plan velocity controls by anticipating others' behaviors in a time horizon
- AVs utilize their predefined driving costs in a non-cooperative way



Contributions:

- Model AVs non-cooperative driving behaviors by mean field game
- Solve MFG and quantify equilibrium control performance

N-car Differential Game

- dynamic

$$\dot{x}_i(t) = v_i(t), \quad x_i(0) = x_{i,0}, \quad i = 1, 2, \ldots, N,$$

position speed

- driving cost

$$J_i^N(v_i, v_{-i}) = \underbrace{\int_0^T \underbrace{f_i^N(v_i(t), x_i(t), x_{-i}(t))}_{\text{cost function}}\, dt}_{\text{running cost}} + \underbrace{V_T(x_i(T))}_{\text{terminal cost}},$$

- admissible set

$$\mathcal{A} = \{v(\cdot): \ 0 \le v(t) \le u_{\max}, \forall t \in [0, T]\}$$

- Nash equilibrium

$$J_i^N(v_i^*, v_{-i}^*) \le J_i^N(v_i, v_{-i}^*), \quad \forall v_i \in \mathcal{A}, \quad i = 1, \ldots, N.$$

Mean Field Game (MFG)

Mean field limit ($N \to \infty$)

$x_1(t), \cdots x_N(t) \to \rho(x, t)$
positions density

$v_1(t), \cdots v_N(t) \to u(x, t)$
speeds velocity

Optimal cost: minimizes

$$V(x, t) = \min_{v:[t,T] \to [0, u_{\max}]} \left[\int_t^T f(v(s), \rho(x(s), s))\, ds + V_T(x(T)) \right],$$

$$\text{s.t. } \dot{x}(s) = v(s), \quad x(t) = x,$$

MFG system

$$[\text{MFG}] \begin{cases} (\text{CE}) & \rho_t + (\rho u)_x = 0, \\ (\text{HJB}) & V_t + f^*(V_x, \rho) = 0, \\ & u = f_p^*(V_x, \rho). \end{cases}$$

Cost Function

- MFG-Nonseparable

$$f_{\text{NonSep}}(u, \rho) = \underbrace{\frac{1}{2}\left(\frac{u}{u_{\max}}\right)^2}_{\text{kinetic energy}} - \underbrace{\frac{u}{u_{\max}}}_{\text{efficiency}} + \underbrace{\frac{u\rho}{u_{\max}\rho_{\text{jam}}}}_{\text{safety}}$$

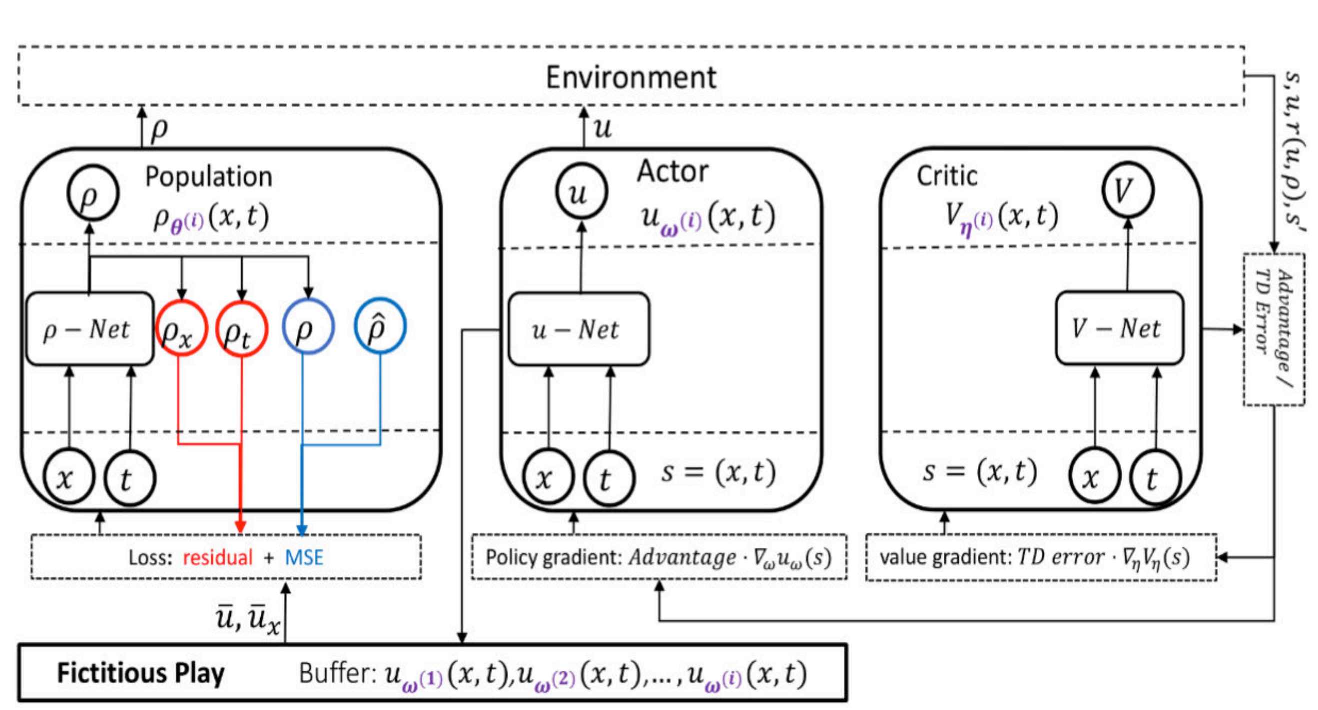
- MFG-Separable

$$f_{\text{Sep}}(u, \rho) = \underbrace{\frac{1}{2}\left(\frac{u}{u_{\max}}\right)^2}_{\text{kinetic energy}} - \underbrace{\frac{u}{u_{\max}}}_{\text{efficiency}} + \underbrace{\frac{\rho}{\rho_{\text{jam}}}}_{\text{safety}}$$

- MFG-LWR

$$f_{\text{LWR}}(u, \rho) = \frac{1}{2}(U(\rho) - u)^2$$

Framework



Algorithm

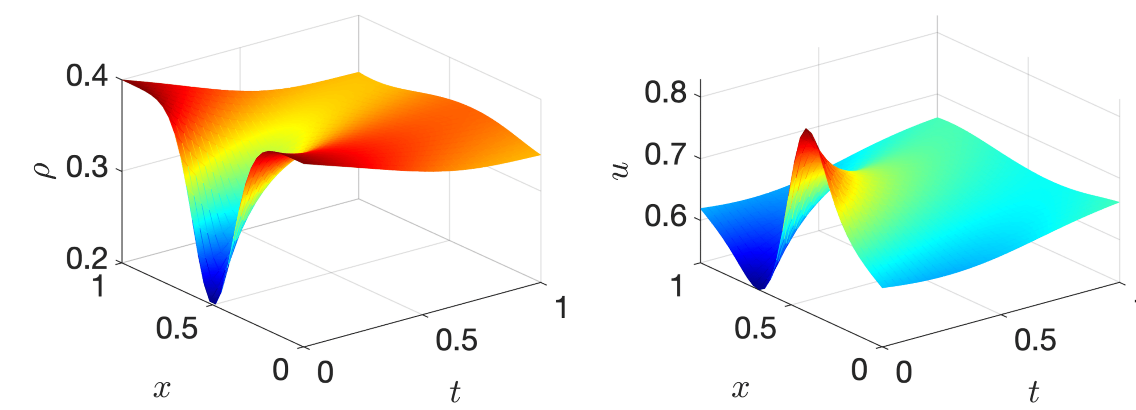
Algorithm 1 MFG-RL-PIDL

1: Initialization: Population network ρ-Net: $\rho_{\theta^{(0)}}(s)$; Actor network u-Net: $u_{\omega^{(0)}}(s)$ and critic network V-Net: $V_{\eta^{(0)}}(s)$.
2: **for** $i \leftarrow 0$ to I **do**
3: Sample a batch of states **s** from state space $\mathcal{X} \times \mathcal{T}$;
4: **for** each state s_l in **s** **do** —*RL - the representative agent*
5: Select u according $u_{\omega^{(i)}}(s_l)$;
6: Obtain ρ according $\rho_{\theta^{(i)}}(s_l)$;
7: Execute u and observe reward $r(u, \rho)$;
8: Update state $s_l \to s_l'$;
9: Obtain value function: $V_{\eta^{(i)}}(s), V_{\eta^{(i)}}(s')$.
10: **end for**
11: Calculate the advantage (Equation 15);
12: Store the actor network $u_{\omega^{(i)}}(s)$ into buffer. —*FP*
13: Compute $\bar{u}$ (Equation 13);
14: Obtain MSE_o (Equation 11); —*PIDL - Population*
15: Obtain residual (Equation 14 and 16);
16: Update ρ-Net, u-Net and V-Net and obtain $\rho_{\theta^{(i+1)}}(s)$, $u_{\omega^{(i+1)}}(s)$ and $V_{\eta^{(i+1)}}(s)$;
17: Check convergence (Equation 17).
18: **end for**
19: Output u, ρ

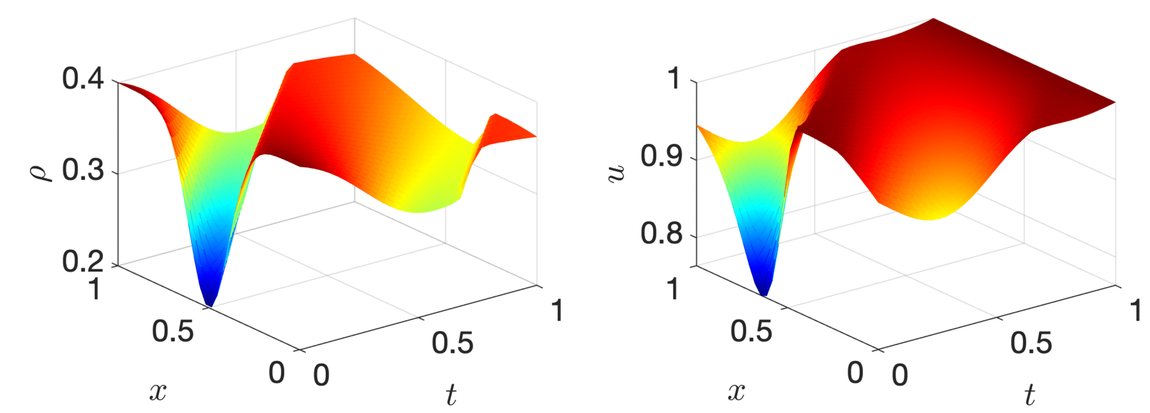
Numerical Results

MFE

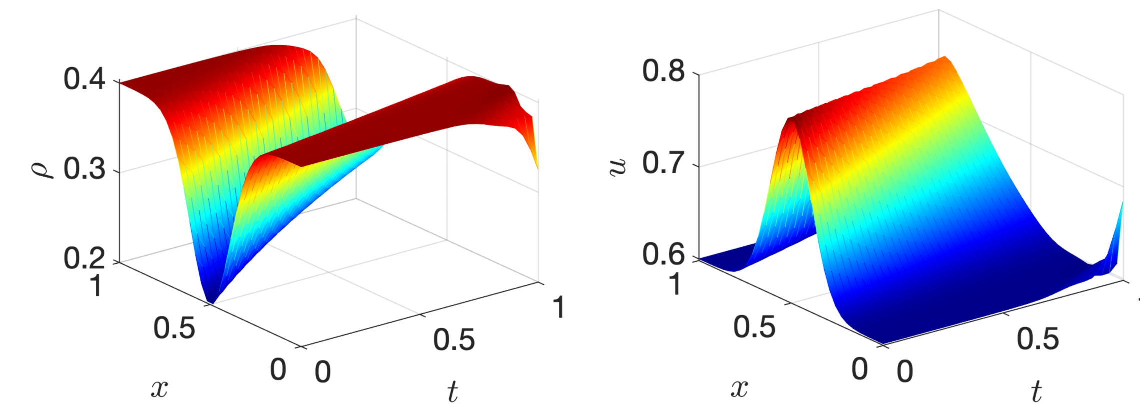
- MFG-Nonseparable



- MFG-Separable



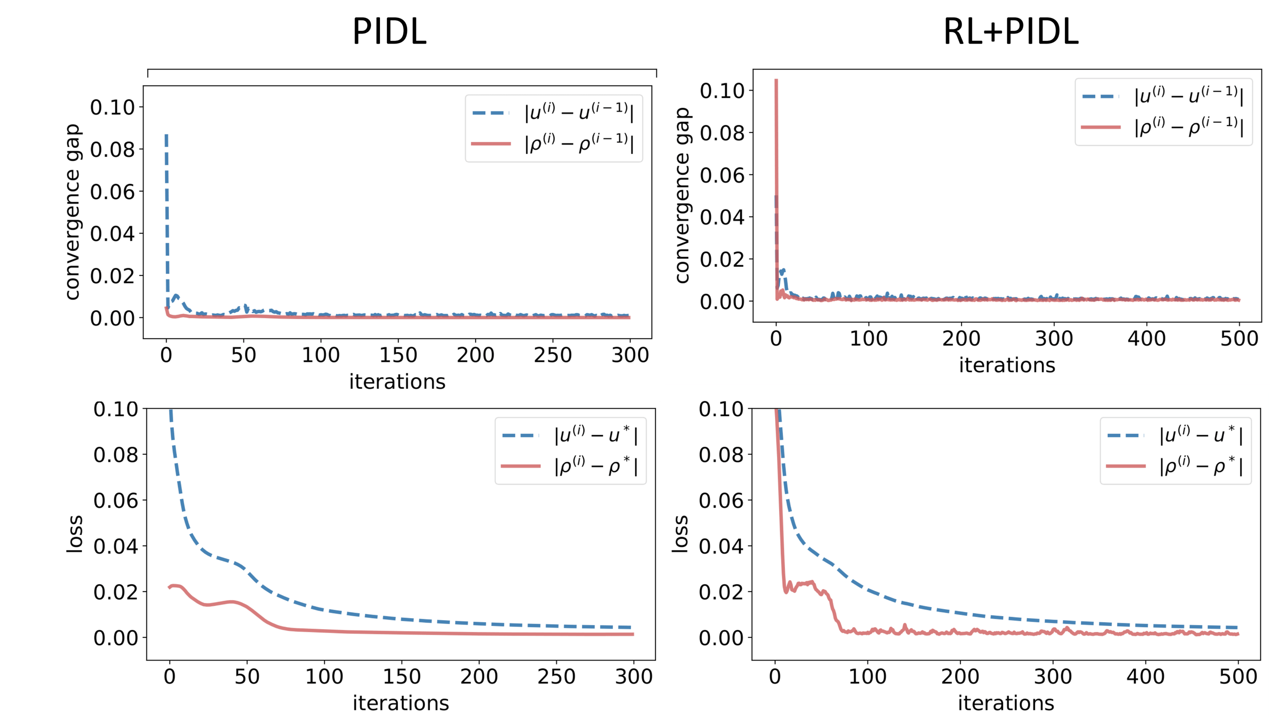
- MFG-LWR



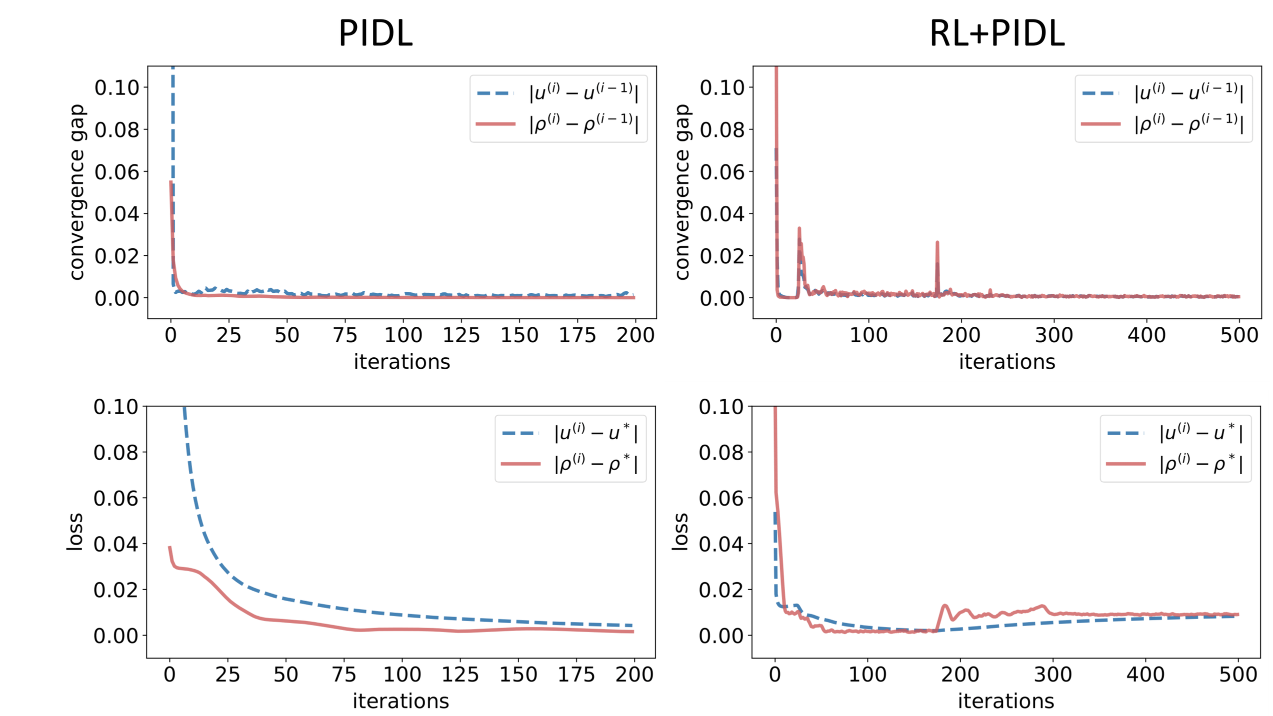
Numerical Results

Convergence

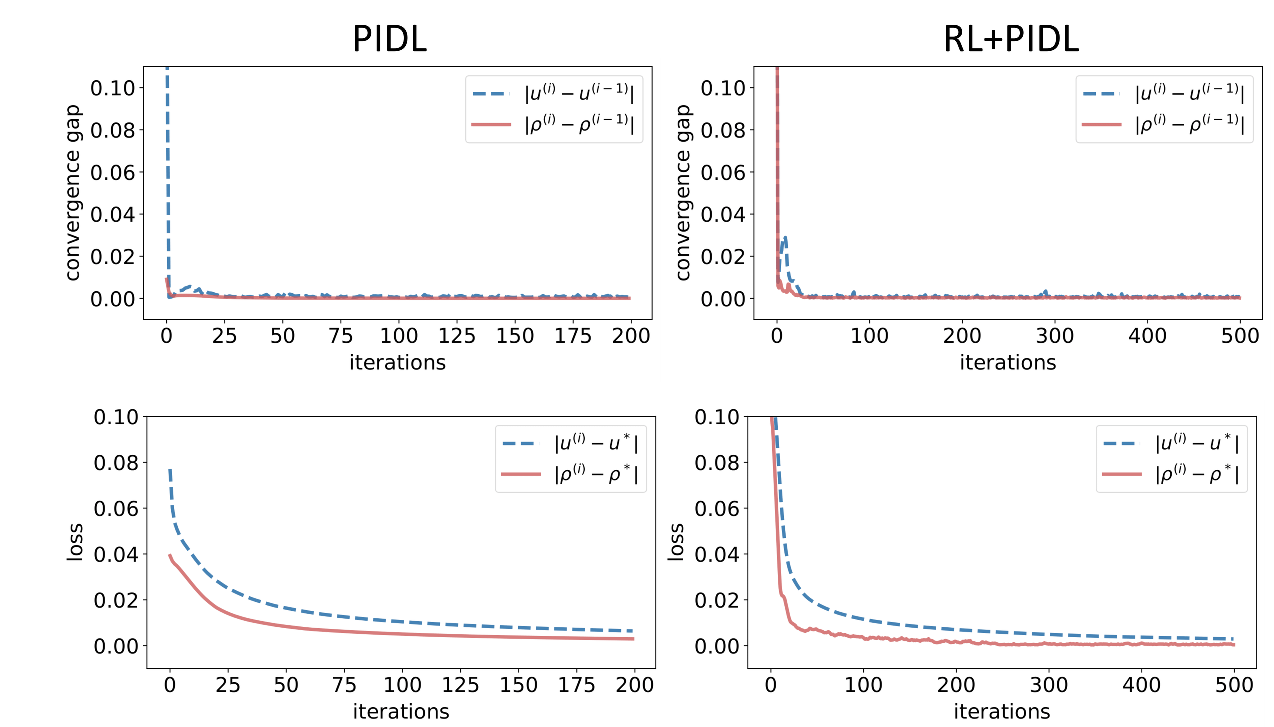
- MFG-Nonseparable



- MFG-Separable



- MFG-LWR



- Exploitability