Physics-Informed Neural Operator for Coupled Forward-Backward Partial Differential Equations

Xu Chen [1] **Yongjie Fu** [1] **Shuo Liu** [1] **Xuan Di** [1]

Abstract

This paper proposes a physics-informed neural operator (PINO) framework to solve a system of coupled forward-backward partial differential equations (PDEs) arising from mean field games (MFGs). The MFG system incorporates a forward PDE to model the propagation of population dynamics and a backward PDE for a representative agent's optimal control. The PINO is developed to tackle the forward PDE efficiently, particularly when the initial population density varies. A learning algorithm is devised and its performance is evaluated on one application domain, which is autonomous driving velocity control. The PINO exhibits both memory efficiency and generalization capabilities, compared to physics-informed neural networks (PINN).

1. INTRODUCTION

In contrast to numerical solvers for partial differential equations (PDEs), recent years have seen a growing trend of using neural networks (NN) to approximate PDE solutions because of its grid-free scheme (Raissi et al., 2019; Di et al., 2023), which, however, requires a large amount of data samples to train. Physics informed neural networks (PINN) have demonstrated their data-efficiency in training physics uninformed neural networks (PUNN) to solve PDEs (Shi et al., 2021). However, one shortcoming is that, a new NN has to be re-trained every time when the input initial conditions vary. It thus lacks generalization capability to a family of PDEs differing only in initial conditions. To tackle such a challenge, Fourier neural operator (FNO) is developed to take in various initial conditions to train an NN that could predict PDE outputs with different conditions (Li et al., 2020; Thodi et al., 2023). The rationale is to propagate information from initial or other boundary conditions by projecting them into a high dimensional space using Fourier transformation. Physics information can be further incorporated into FNO, which is physics informed neural operator (PINO) (Li et al., 2021). PINO utilizes physics loss to train the neural operator, which could further reduce the required training data size by leveraging the knowledge of PDE formulations.

In contrast to the classical PDE systems, here we focus on a more difficult class of forward-backward PDE systems, which arise from the concept of game theory, in particular, mean field games (MFG). MFGs are micro-macro games aiming to model the strategic interaction among a large amount of self-interested agents who make dynamic decisions (corresponding to the backward PDE), while a population distribution is propagated to represent the state of interacting individual agents (corresponding to the forward PDE) (Lasry & Lions, 2007; Huang et al., 2006; Cardaliaguet, 2010; 2015). The equilibrium of MFG, so called mean field equilibria (MFE), is characterized by two PDEs, they are,

1. *Agent dynamic*: individuals' dynamics using optimal control, i.e, a backward Hamilton-Jacobi-Bellman (HJB) equation, solved backwards using dynamic programming given the terminal state;
2. *Mass dynamic*: system evolution arising from each individual's choices, i.e, a forward Fokker-Planck-Komogorov (FPK) equation, solved forward provided the initial state, representing agents' anticipation of other agents' choices and future system dynamics.

MFE is challenging to solve due to the coupled forward and backward structure of these two PDE systems. Therefore, researchers seek various machine learning methods, including reinforcement learning (RL) (Kizilkale & Caines, 2013; Yin et al., 2014; Yang et al., 2018; Elie et al., 2020; Perrin et al., 2021; Mguni et al., 2018; Subramanian et al., 2022; Chen et al.) and PINN (Ruthotto et al., 2020; Carmona & Laurière, 2021; Germain et al., 2022; Chen et al., 2023). Among them, PINN is one method that has shown its efficiency in solving MFEs with continuous states and actions.

Motivated by the MFG framework, this paper aims to tackle a more challenging task, namely, solving a set of coupled forward-backward PDE systems with arbitrary initial condi-

[1]Columbia University, New York, United States. Correspondence to: Xuan Di <sharon.di@columbia.edu>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023.

tions. To achieve this goal, we train a PINO to approximate the solution to a family of the coupled PDE system with various initial conditions.

The rest of this paper is organized as: Section 2 presents preliminaries about the coupled PDE system and PINO. Section 3 proposes a PINO learning framework for coupled PDEs. Section 4 presents the solution approach. Section 5 demonstrates numerical experiments. Section 6 concludes.

2. Preliminary

2.1. Spatiotemporal Mean Field Games (ST-MFG)

Spatiotemporal MFG (ST-MFG) models a class of MFGs defined in a spatiotemporal domain over a finite horizon. An ST-MFG is a system of partial differential equations comprising the forward FPK and backward HJB equations. Mathematically, we have the following PDEs for ST-MFG:

$$\begin{aligned}
&[\text{ST-MFG}]\forall(x,t)\in\mathcal{X}\times\mathcal{T} \\
&(\text{FPK})\ \rho_t+(\rho\cdot u)_x=0, && (1)\\
&\rho(x,0)\equiv\boldsymbol{\rho}_0, && (2)\\
&(\text{HJB})\ V_t+\min_u\{f(u,\rho)+uV_x\}=0, && (3)\\
&V(x,T)\equiv\boldsymbol{V}_T. && (4)
\end{aligned}$$

We now explain the details in the PDE system.

Definition 2.1. ST-MFG. A population of agents navigate a space domain $\mathcal{X}$ with a finite planning horizon $\mathcal{T}=[0,T]\,, T\in[0,\infty)$. At time $t\in\mathcal{T}$, a generic agent selects continuous-time-space decision $u(x,t)$ at position $x\in\mathcal{X}$. The decision triggers the evolution of population density $\rho(x,t)$ over the spatiotemporal domain. The generic agent aims to minimize the total cost arising from the population density, indicating a congestion effect.

FPK (Equ. 1 and 2). $\rho(x,t),\forall(x,t)\in\mathcal{X}\times\mathcal{T}$ is the population density of all agents in the system (i.e., mean-field state). ρ_t,ρ_x are partial derivatives of $\rho(x,t)$ with respect to t,x, respectively. $\boldsymbol{\rho}_0$ denotes the initial population density over the space domain $\mathcal{X}$. The FPK equation captures the population dynamics starting from the initial density.

HJB (Equ. 3 and 4). The HJB equation depicts the optimal control of a generic agent. We specify each element in the optimal control problem as: **State** (x,t) is the agent's position at time t. $(x,t)\in\mathcal{X}\times\mathcal{T}$. **Action** $u(x,t)$ is the velocity of the agent at position x at time t. The optimal velocity evolves as time progresses. **Cost** $f(u,\rho)$ is the congestion cost depending on agents' action u and population density ρ. **Value function** $V(x,t)$ is the minimum cost of the generic agent starting from position x at time t. V_t,V_x are partial derivatives of $V(x,t)$ with respect to t,x, respectively. $\boldsymbol{V}_T$ denotes the terminal cost.

Definition 2.2. Mean Field Equilibrium (MFE). In an ST-MFG, $(u^*(x,t),\rho^*(x,t)),\forall(x,t)\in\mathcal{X}\times\mathcal{T}$ is called an MFE if following conditions hold: (1) $\rho^*_t+(\rho^*\cdot u^*)_x=0$; (2) $V^*_t+u^*V^*_x+f(u^*,\rho^*)=0$; (3) $u^*=argmin_p\{f(p,\rho)+pV_x\}$. In this work, we adopt a neural operator to find MFE.

2.2. Physics-Informed Neural Operator (PINO)

The physics-informed neural operator (PINO) is proposed to solve PDEs with high computational speed (Li et al., 2021). The PINO utilizes physics in training a Fourier neural operator (Li et al., 2020), also referred to as FNO. Compared to traditional physics-informed neural networks (PINNs), PINO has the ability to propagate information from initial or other boundary conditions by projecting boundary conditions into a high dimensional space using Fourier transformation in the FNO. This enables the development of a scalable learning framework for solving ST-MFG with various initial population densities, eliminating the need to retrain multiple PINNs

3. Learning ST-MFG via PINO

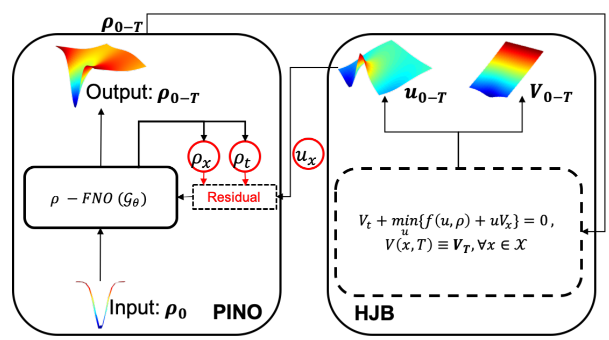


Figure 1. Scalable Learning Framework for ST-MFG

In this section, we introduce a scalable learning framework where a PINO effectively captures population dynamics, which are determined by various initial densities and the optimal control in an ST-MFG. Fig. 1 illustrates the workflow of our proposed framework. The PINO module utilizes a Fourier neural operator ρ-FNO to represent population density $\boldsymbol{\rho}_{0-T}$. The ρ-FNO is updated using a residual defined by the physical rule (FPK) that captures the relationship between population evolution and velocity control. The optimal velocity $\boldsymbol{u}_{0-T}$ is computed by the HJB equation given the population density $\boldsymbol{\rho}_{0-T}$. These two modules internally depend on each other. We now introduce them separately.

3.1. PINO module for FPK equation

Fig. 2 illustrates the architecture of PINO. In the context of our problem, we feed the neural operator with input fields $\boldsymbol{\rho}_0$ over the space domain $\mathcal{X}$, which represents the initial density in ST-MFG. The operator processes this input and produces the output, which corresponds to the population density over a spatiotemporal domain $\boldsymbol{\rho}_{0-T}$. The ρ-FNO consists of N Fourier layers. Each Fourier layer transforms information through two processes: the Fourier transformation $\mathcal{F} \to l \to \mathcal{F}^{-1}$ and the linear transformation l'. The output resulting from these two transformations is subsequently fed into a nonlinear activation function σ to produce the input for the next Fourier layer.

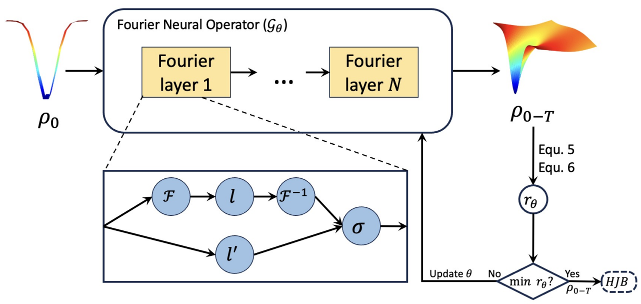


Figure 2. PINO Architecture

The training of ρ-FNO is guided by the residual (marked in red) determined by physical rules of population dynamics. Mathematically, the residual r_θ is calculated as:

$$r_\theta = \frac{\sum_{\boldsymbol{\rho}_0 \in \boldsymbol{\rho}^{\mathcal{D}}} L_{FPK}(\boldsymbol{\rho}_0, \mathcal{G}_\theta(\boldsymbol{\rho}_0))}{|\boldsymbol{\rho}^{\mathcal{D}}|}, \tag{5}$$

where, the set $\boldsymbol{\rho}^{\mathcal{D}}$ contains various initial densities. $L_{FPK}(\boldsymbol{\rho}_0, \mathcal{G}_\theta)$ is the physics loss, which is calculated as:

$$L_{FPK} = \alpha ||\mathcal{G}_\theta|_{t=0} - \boldsymbol{\rho}_0||^2 + \beta || \frac{\partial \mathcal{G}_\theta}{\partial t} + \frac{\partial(\mathcal{G}_\theta \cdot u)}{\partial x}||^2. \tag{6}$$

The first term in the physics loss quantifies the difference between the output of the operator at time 0 and the initial density. The second term evaluates the physical discrepancy based on Equ. 1. The weight parameters α and β are used to adjust the relative importance of these terms. The optimal control u is obtained from the agent control module.

3.2. HJB module

We solve the HJB equation (Equ. 3) to determine the optimal velocity given the population dynamics. Numerical methods commonly employed for solving the HJB equation include backward induction (Perrin et al., 2021), the Newton method (Huang et al., 2020), and variational inequality (Huang et al., 2021). Learning-based methods, such as RL (Guo et al., 2019; Perrin et al., 2020) and PIDL (Chen et al., 2023), can also be used for solving the HJB equation. In this work, we adopt backward induction since the dynamics of the agents and the cost functions are known in the MFG system.

4. Solution Approach

In this section, we develop a learning algorithm (Alg. 1) based on the proposed scalable learning framework. In Alg. 1, we first initialize the neural operator $\mathcal{G}_{\theta^{(0)}}$, parameterized by $\theta^{(0)}$. During the ith iteration of the training process, we first sample a batch of initial population densities $\boldsymbol{\rho}_0$. We use each $\boldsymbol{\rho}_0$ to generate the population density over the entire spatiotemporal domain $\boldsymbol{\rho}_{0-T}$. Given $\boldsymbol{\rho}_{0-T}$, we perform backward induction (Lines 6-9) to calculate the optimal speed $\boldsymbol{u}_{0-T}^{(i)}$. We then update the parameter θ of the neural operator according to the residual. We check the following convergence conditions for the population density $\boldsymbol{\rho}_{0-T}$ obtained by $\mathcal{G}_\theta(\rho_0)$:

$$\frac{\sum_{\boldsymbol{\rho}_0 \in \boldsymbol{\rho}^{\mathcal{D}}} |\mathcal{G}_{\theta^{(i)}}(\rho_0) - \mathcal{G}_{\theta^{(i-1)}}(\rho_0)|}{|\boldsymbol{\rho}^{\mathcal{D}}|} < \epsilon_\rho \tag{7}$$

The training process moves on to the next iteration till the convergence condition holds.

Algorithm 1 PINO-MFG

1: Initialization: ρ-FNO: $\mathcal{G}_{\theta^{(0)}}$;
2: **for** $i \leftarrow 0$ to I **do**
3:   Sample a batch of initial population densities $\boldsymbol{\rho}_0$ from the set $\boldsymbol{\rho}^{\mathcal{D}}$ of density distribution;
4:   Generate $\boldsymbol{\rho}_{0-T}^{(i)}$ using the neural operator $\mathcal{G}_{\theta^{(i)}}(\boldsymbol{\rho}_0)$ corresponding to each $\boldsymbol{\rho}_0$ in the batch;
5:   **for** each $\boldsymbol{\rho}_{0-T}^{(i)}$ generated by ρ-FNO **do**
6:     **for** $t \leftarrow T$ to 1 **do**
7:       $u_{t-1}^{(i)} \leftarrow argmin_u\{f(u, \rho) + uV_x(t, x)\}$;
8:       Update value function using $u_{t-1}^{(i)}$ according to the HJB equation.
9:     **end for**
10:     Obtain $\boldsymbol{u}_{0-T}^{(i)}$ and calculate $\boldsymbol{u}_x^{(i)}$;
11:   **end for**
12:   Obtain residual $r_{\theta^{(i)}}$ according to Equ. 5 and 6;
13:   Update the neural operator and obtain $\mathcal{G}_{\theta^{(i+1)}}$;
14:   Check convergence (Equ. 7).
15: **end for**
16: Output $\boldsymbol{\rho}_{0-T}$

We also try two PINN-based algorithms (Chen et al., 2023) for ST-MFGs. One is a joint RL and PIDL algorithm, which iteratively solves HJB using the actor-critic method, and FPK using a PINN. The other is a pure PIDL algorithm that iteratively solves HJB and FPK using two PINNs. We refer these baseline algorithms as “RL-PIDL” and “Pure-PIDL”, respectively. Note that PINN has difficulty propagating information from initial conditions (Li et al., 2021), RL-PIDL and Pure-PIDL have to assign and retrain new NNs to handle various initial population densities. In contrast, our proposed PINO framework does not encounter the memory and efficiency issues faced by the baselines.

5. Experiment

In this section, we apply the proposed algorithm to autonomous driving velocity control problem. We first introduce the MFG system for autonomous driving: A population of autonomous vehicles (AVs) navigate a ring road (Fig. 3). At time t, a generic AV selects $u(x,t)$ at position x. The speed choice triggers the evolution of population density over the ring road. The generic AV aims to minimize the total cost. The length of ring road is 1. It means positions $x = 0$ and $x = 1$ are the same. AVs move along the ring road until time T. We assume AVs have no preference for their locations at time T, i.e. $V(x,T) = 0, \forall x \in [0,1]$.

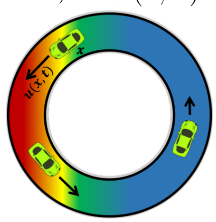


Figure 3. Autonomous Driving

Fig. 4 illustrates the initial conditions $\rho(x,0)$ sampled from the set $\boldsymbol{\rho}^{\mathcal{D}}$. The initial density $\rho(x,0)$ represents the distribution of the AV population over the ring road at time 0. The set $\boldsymbol{\rho}^{\mathcal{D}}$ comprises density curves with a bell shape: $\hat{\rho}(x) = a - (a-b)e^{-\frac{(x-0.5)^2}{\delta^2}}, a < b$ and an inverted bell shape: $\hat{\rho}(x) = a - (a-b)e^{-\frac{(x-0.5)^2}{\delta^2}}, a > b$. These density curves are parameterized by a, b, and δ, where $a, b \sim \text{Uniform}[0.2, 0.6]$ and $\delta \sim \text{Uniform}[0.1, 0.2]$.

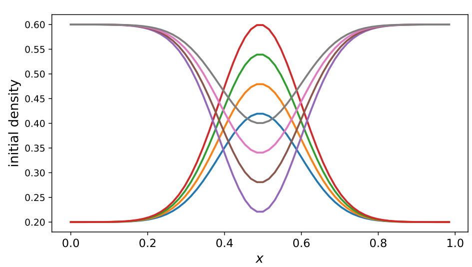
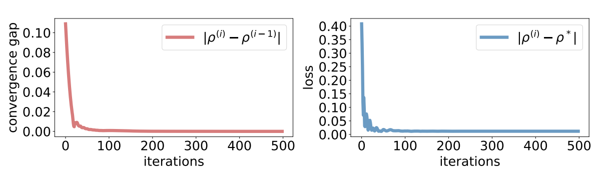


Figure 4. Sample Initial Conditions $\rho(x,0)$

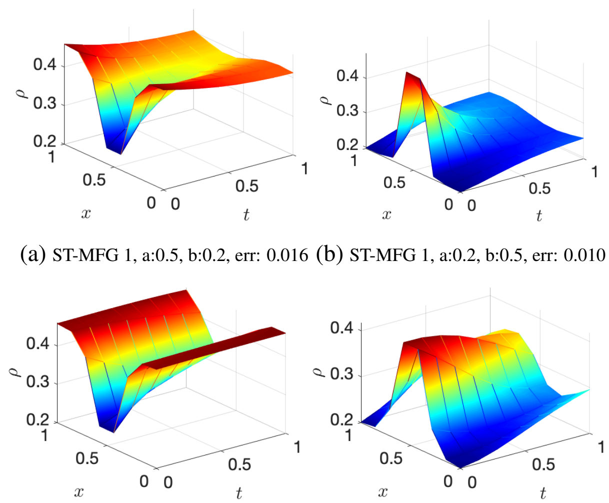
Fig. 5 demonstrates the performance of the algorithm in solving ST-MFG. The x-axis represents the iteration index during training. Fig. 5a displays the convergence gap, calculated as $|\rho^{(i)} - \rho^{(i-1)}|$. Fig. 5b displays the 1-Wasserstein distance (W1-distance), which measures the closeness (Lauriere et al., 2022) between our results and the MFE (mean field equilibrium) obtained by numerical methods, represented as $|\rho^{(i)} - \rho^{*}|$. Our proposed algorithm converges to the MFE after 200 iterations.



(a) Convergence gap (b) W1-distance

Figure 5. Algorithm Performance.

Fig. 6 demonstrates the population density ρ^* at MFE in two different ST-MFGs, which are referred as ST-MFG 1 and 2. The cost function in ST-MFG 1 is $r(u,\rho) = \frac{1}{2}u^2 - u + u\rho$. In ST-MFG 1, the population of AVs incurs a penalty if they select the same velocity control, whereas in ST-MFG 2, they do not. The x-axis represents position x, and the y-axis represents time t. Fig. 6a and 6c have an initial density with an inverted bell shape. Fig. 6b and 6d have an initial density with a bell shape. The cost function in ST-MFG 2 is $r(u,\rho) = \frac{1}{2}(1-\rho-u)^2$. Compared to the equilibrium ρ^* in ST-MFG 2, the population density in ST-MFG 1 quickly dissipates and no wave forms.



(a) ST-MFG 1, a:0.5, b:0.2, err: 0.016 (b) ST-MFG 1, a:0.2, b:0.5, err: 0.010

(c) ST-MFG 2, a:0.5, b:0.2, err: 0.018 (d) ST-MFG 2, a:0.2, b:0.5, err: 0.013

Figure 6. Population Density ρ^* at MFE.

In Table 1, we make a comparison of different learning methods. The computational time refers to the total training time (unit: s) required for solving ST-MFGs with various initial conditions. The PINO-based algorithm demonstrates a reduced need for neural networks (NNs) and shorter training time compared to PINN-based algorithms.

		PINO	RL-PIDL	Pure-PIDL
Memory (Number of NNs)		1	48	32
Time (s)	ST-MFG 1	48.57	1544.96	687.44
	ST-MFG 2	46.43	1384.48	313.36

Table 1. Comparison of Existing Learning Methods

6. Conclusion

This work presents a scalable learning framework for solving coupled forward-backward PDE systems using a physics-informed neural operator (PINO). PINO allows for efficient training of the forward PDE with varying initial conditions. Compared to traditional physics-informed neural networks (PINNs), our proposed framework overcomes memory and efficiency limitations. We also demonstrate the efficiency of this method on a numerical example motivated by optimal autonomous driving control. The PINO-based framework offers a memory, data efficient approach for solving complex PDE systems with generalizability to similar PDE systems differing only in boundary conditions.

Broader impact

This work is motivated by the computational challenge faced by the mean field game (MFG). MFGs have gained increasing popularity in recent years in finance, economics, and engineering, due to its power to model the strategic interactions among a large number of agents in multi-agent systems. The equilibria associated with MFGs, aka, mean field equilibra (MFE), are challenging to solve due to its coupled forward and backward PDE structure. That is why computational methods based on machine learning have gained momentum. The PINO-based learning method developed in this study empowers the generalization of the trained neural networks to various initial conditions, which holds the potential to solve large-scale MFGs, in particular graph-based applications, including but not limited to autonomous vehicle driving on road networks, pedestrian or crowd dynamics, vehicle fleet network management, internet packet routing, social opinion dynamics, and epidemiology.

References

Cardaliaguet, P. Notes on mean field games. Technical report, Stanford University, 2010.

Cardaliaguet, P. *Weak solutions for first order mean field games with local coupling*, pp. 111–158. Springer, Cham, 2015.

Carmona, R. and Laurière, M. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games i: The ergodic case. *SIAM Journal on Numerical Analysis*, 59(3):1455–1485, 2021.

Chen, X., Liu, S., and Di, X. Learning dual mean field games on graphs. In *Proceedings of the 26th European Conference on Artificial Intelligence*, ECAI 2023.

Chen, X., Liu, S., and Di, X. A hybrid framework of reinforcement learning and physics-informed deep learning for spatiotemporal mean field games. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, 2023.

Di, X., Shi, R., Mo, Z., and Fu, Y. Physics-informed deep learning for traffic state estimation: A survey and the outlook. *Algorithms*, 16(6):305, 2023.

Elie, R., Perolat, J., Laurière, M., Geist, M., and Pietquin, O. On the convergence of model free learning in mean field games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7143–7150, New York, NY, USA, 04 2020. AAAI.

Germain, M., Mikael, J., and Warin, X. Numerical resolution of mckean-vlasov fbsdes using neural networks. *Methodology and Computing in Applied Probability*, 24 (4):1–30, 2022.

Guo, X., Hu, A., Xu, R., and Zhang, J. Learning mean-field games. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Huang, K., Di, X., Du, Q., and Chen, X. A game-theoretic framework for autonomous vehicles velocity control: Bridging microscopic differential games and macroscopic mean field games. *Discrete and Continuous Dynamical Systems - B*, 25(12):4869–4903, 2020. ISSN 1531-3492.

Huang, K., Chen, X., Di, X., and Du, Q. Dynamic driving and routing games for autonomous vehicles on networks: A mean field game approach. *Transportation Research Part C: Emerging Technologies*, 128:103189, 2021.

Huang, M., Malhamé, R. P., and Caines, P. E. Large population stochastic dynamic games: closed-loop Mckean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information & Systems*, 6(3): 221–252, 2006.

Kizilkale, A. C. and Caines, P. E. Mean field stochastic adaptive control. *IEEE Transactions on Automatic Control*, 58(4):905–920, 2013. doi: 10.1109/TAC.2012.2228032.

Lasry, J.-M. and Lions, P.-L. Mean field games. *Japanese journal of mathematics*, 2(1):229–260, 2007.

Lauriere, M., Perrin, S., Girgin, S., Muller, P., Jain, A., Cabannes, T., Piliouras, G., Perolat, J., Elie, R., Pietquin, O., and Geist, M. Scalable deep reinforcement learning algorithms for mean field games. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12078–12095. PMLR, 2022.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., and Anandkumar, A. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2111.03794*, 2021.

Mguni, D., Jennings, J., and de Cote, E. M. Decentralised learning in systems with many, many strategic agents. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Perrin, S., Perolat, J., Laurière, M., Geist, M., Elie, R., and Pietquin, O. Fictitious play for mean field games: Continuous time analysis and applications. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, 2020.

Perrin, S., Laurière, M., Pérolat, J., Geist, M., Élie, R., and Pietquin, O. Mean field games flock! the reinforcement learning way. In *International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Ruthotto, L., Osher, S. J., Li, W., Nurbekyan, L., and Fung, S. W. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

Shi, R., Mo, Z., and Di, X. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 540–547, 2021.

Subramanian, S., Taylor, M., Crowley, M., and Poupart, P. Decentralized mean field games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Thodi, B. T., Ambadipudi, S. V. R., and Jabari, S. E. Learning-based solutions to nonlinear hyperbolic pdes: Empirical insights on generalization errors. *arXiv preprint arXiv:2302.08144*, 2023.

Yang, J., Ye, X., Trivedi, R., Xu, H., and Zha, H. Deep mean field games for learning optimal behavior policy of large populations. In *International Conference on Learning Representations*, 2018.

Yin, H., Mehta, P. G., Meyn, S. P., and Shanbhag, U. V. Learning in mean-field games. *IEEE Transactions on Automatic Control*, 59(3):629–644, 2014. doi: 10.1109/TAC.2013.2287733.