

Data-Driven Network LQG Mean Field Games with Heterogeneous Populations via Integral Reinforcement Learning¹

Jean Zhu and Shuang Gao

Polytechnique Montréal, GERAD²

August 21, 2026

¹MTNS 2026. Supported by FRQNT and NSERC (Canada).

²Groupe d'études et de recherche en analyse des décisions

Motivation

Challenge

- ▶ Large-scale systems (i.e. smart power grids, traffic network, EV users) consist of agents making independent decisions and interacting with others in a competitive environment.
- ▶ As the number of agents grows, solutions become computationally intractable.

Mean-Field Games

- ▶ Resolve the scaling problem by approximating interactions through a "Mean Field".
- ▶ Standard MFG solutions require exact knowledge of system dynamics which are often unknown in real-world conditions.

Solution

Learn Nash equilibrium directly from data trajectories.



Outline

Problem Formulation

Model-Based Solution

Data-Driven Solution

 Data-Driven Formulation

 Iterative Algorithm

Numerical Example

Conclusion

Problem Formulation

Consider a large population of agents in $\mathcal{K} \in \mathbb{N}^+$ classes interacting through a network.

Agent dynamics (class k , agent i):

$$dx_{k,i} = (A_k x_{k,i} + B_k u_{k,i}) dt + D_k d\omega_{k,i}$$

$x_{k,i} \in \mathbb{R}^{n_k}$: state $u_{k,i} \in \mathbb{R}^{m_k}$: control input

$\omega_{k,i} \in \mathbb{R}^{d_k}$: Brownian noise

Cost (infinite horizon):

$$J_k(u_{k,i}, u_{-k,i}) = \mathbb{E} \int_0^\infty e^{-\rho t} \left[\|x_{k,i} - \sum_{m=1}^{\mathcal{K}} H_{km} \bar{x}_m^{(c_m)}\|_{Q_k}^2 + \|u_{k,i}\|_{R_k}^2 \right] dt$$

H_{km} : coupling between classes,

$\bar{x}_m^{(c_m)} = \frac{1}{c_m} \sum_{i=1}^{c_m} x_{m,i}$ is the mean of class m ,

$Q_k = Q_k^\top \succeq 0$ and $R_k = R_k^\top \succ 0$.

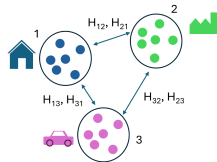


Figure 1: Networked interactions across classes

Objective

Find Nash equilibrium $(u_{1,1}, \dots, u_{\mathcal{K}, c_{\mathcal{K}}})$ associated with costs J_k without knowing system dynamics.

Problem Formulation

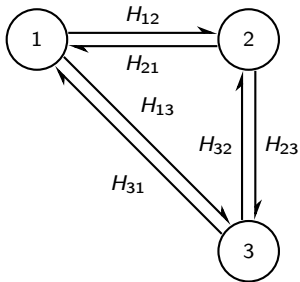
The following matrices are defined:

$$\mathcal{A} = \text{diag}(A_1, \dots, A_K), \quad \mathcal{B} = \text{diag}(B_1, \dots, B_K),$$
$$\mathcal{Q} = \text{diag}(Q_1, \dots, Q_K), \quad \mathcal{R} = \text{diag}(R_1, \dots, R_K),$$

and a block matrix $\mathcal{H} \in \mathbb{R}^{N \times N}$, where $N = \sum_{k=1}^K n_k$, composed of blocks $H_{km} \in \mathbb{R}^{n_k \times n_m}$, encodes the network coupling between classes

$$\mathcal{H} = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1K} \\ H_{21} & H_{22} & \cdots & H_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ H_{K1} & H_{K2} & \cdots & H_{KK} \end{bmatrix}.$$

For the simple 3 class network in the introduction, the $\mathcal{H}$ matrix takes the form



$$\mathcal{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

Model-Based Solution

Proposition 1 (Zhu and Gao, 2026)³

Under technical assumptions, the MFG strategy of agent $a_{k,i}$ is uniquely given by

$$u_{k,i}^* = -R_k^{-1} B_k^\top (P_k x_{k,i} + s_k), \quad (1)$$

where $P_k \succ 0$ and $s_k = \sum_{m=1}^K \Pi_{km} \bar{x}_m$. Defining matrices $\mathcal{P} \in \mathbb{R}^{N \times N}$ and $\Pi \in \mathbb{R}^{N \times N}$ such that

$$\mathcal{P} = \text{diag}(P_1, \dots, P_K) \text{ and } \Pi = \begin{bmatrix} \Pi_{11} & \Pi_{12} & \dots & \Pi_{1K} \\ \Pi_{21} & \Pi_{22} & \dots & \Pi_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \Pi_{K1} & \Pi_{K2} & \dots & \Pi_{KK} \end{bmatrix}$$

Matrices $P_k \in \mathbb{R}^{n_k \times n_k}$ and $\Pi_{km} \in \mathbb{R}^{n_k \times n_m}$ are solutions to

$$\rho \mathcal{P} = \mathcal{Q} + \mathcal{P} \mathcal{A} + \mathcal{A}^\top \mathcal{P} - \mathcal{P} B R^{-1} B^\top \mathcal{P}, \quad (2a)$$

$$\rho \Pi = -\mathcal{Q} \mathcal{H} + \Pi (\mathcal{A} - B R^{-1} B^\top (\mathcal{P} + \Pi)) + (\mathcal{A} - \mathcal{P} B R^{-1} B^\top) \Pi. \quad (2b)$$

The mean field dynamics are given by

$$\dot{\bar{X}} = (\mathcal{A} - B R^{-1} B^\top (\mathcal{P} + \Pi)) \bar{X}, \text{ with } \bar{X} = [\bar{x}_1^\top, \bar{x}_2^\top, \dots, \bar{x}_K^\top]^\top \in \mathbb{R}^N \quad (3)$$

and initial condition $\bar{X}(0) = \bar{X}_0$.

³Following the fixed-point approach (Huang et al., 2007; Huang and Zhou, 2020).

Model-Based Solution: Decoupling

Defining matrix $\Omega = \mathcal{P} + \Pi$ and summing the equations in (2), the coupled equations are reduced to decoupled Algebraic Riccati Equations (AREs) in $\mathcal{P}$ and Ω :

$$\rho\mathcal{P} = \mathcal{Q} + \mathcal{P}\mathcal{A} + \mathcal{A}^\top\mathcal{P} - \mathcal{P}\mathcal{B}\mathcal{R}^{-1}\mathcal{B}^\top\mathcal{P}, \quad (4a)$$

$$\rho\Omega = \mathcal{Q}(I_N - \mathcal{H}) + \Omega\mathcal{A} + \mathcal{A}^\top\Omega - \Omega\mathcal{B}\mathcal{R}^{-1}\mathcal{B}^\top\Omega. \quad (4b)$$

After solving (4), Π can be obtained by $\Pi = \Omega - \mathcal{P}$.

Model-Based Solution: Assumptions

To ensure the existence of unique stabilizing solutions, we introduce

Assumption 1

The pair $(A_k - \frac{1}{2}\rho I_n, B_k)$ is stabilizable and the pair $(A_k - \frac{1}{2}\rho I_n, Q_k^{\frac{1}{2}})$ is observable for all classes.

Assumption 2

The eigenvalues of the Hamiltonian matrix $H_\Omega \in \mathbb{R}^{2N \times 2N}$ are strong (N,N) c-splitting, and the associated N-dimensional stable invariant subspace is a graph subspace.

$$H_\Omega = \begin{bmatrix} \mathcal{A} - \frac{1}{2}\rho I_N & -\mathcal{B}\mathcal{R}^{-1}\mathcal{B}^\top \\ -\mathcal{Q}(I_N - \mathcal{H}) & -\mathcal{A}^\top + \frac{1}{2}\rho I_N \end{bmatrix} \in \mathbb{R}^{2N \times 2N}$$

- Assumption 2 ensures the existence of a unique stabilizing solution Ω to (4b) i.e., the closed-loop matrix $\mathcal{A} - \frac{1}{2}\rho I_N - \mathcal{B}\mathcal{R}^{-1}\mathcal{B}^\top \Omega$ is Hurwitz.

Data-Driven Solution

We adapt the Integral Reinforcement Learning for LQG MFGs developed by Xu et al. (2023) to networked multi-class problems.

Track one representative agent $a_{k,1}$ per class k and define :

$$\mathcal{X} = [x_{1,1}^\top, \dots, x_{K,1}^\top]^\top, \quad \mathcal{U} = [u_{1,1}^\top, \dots, u_{K,1}^\top]^\top.$$

Main steps :

- **Data Collection** : Collect state and input trajectory under control input

$$\mathcal{U}(t) = - \underbrace{\mathcal{L}_\Omega^{(0)}}_{\text{Stabilizing Gain}} \mathcal{X}(t) + \underbrace{I(t)}_{\text{Exploration Noise}} = \alpha(t).$$

- **IRL** : Once enough state and input trajectory data is collected, IRL is used to compute matrices $\mathcal{P}$, Ω and associated gains $\mathcal{L}_\mathcal{P}$ and $\mathcal{L}_\Omega$ model-free.
- **Mean-Field Approximation** : Representative agents are then used to approximate the mean-field trajectory $\bar{X}(t)$.



Data-Driven Solution

We adapt the Integral Reinforcement Learning for LQG MFGs developed by Xu et al. (2023) to networked multi-class problems.

Track one representative agent $a_{k,1}$ per class k and define :

$$\mathcal{X} = [x_{1,1}^\top, \dots, x_{K,1}^\top]^\top, \quad \mathcal{U} = [u_{1,1}^\top, \dots, u_{K,1}^\top]^\top.$$

Main steps :

- **Data Collection** : Collect state and input trajectory under control input

$$\mathcal{U}(t) = - \underbrace{\mathcal{L}_\Omega^{(0)}}_{\text{Stabilizing Gain}} \mathcal{X}(t) + \underbrace{I(t)}_{\text{Exploration Noise}} = \alpha(t).$$

- **IRL** : Once enough state and input trajectory data is collected, IRL is used to compute matrices $\mathcal{P}$, Ω and associated gains $\mathcal{L}_\mathcal{P}$ and $\mathcal{L}_\Omega$ model-free.
- **Mean-Field Approximation** : Representative agents are then used to approximate the mean-field trajectory $\bar{X}(t)$.



Data-Driven Solution

We adapt the Integral Reinforcement Learning for LQG MFGs developed by Xu et al. (2023) to networked multi-class problems.

Track one representative agent $a_{k,1}$ per class k and define :

$$\mathcal{X} = [x_{1,1}^\top, \dots, x_{K,1}^\top]^\top, \quad \mathcal{U} = [u_{1,1}^\top, \dots, u_{K,1}^\top]^\top.$$

Main steps :

- **Data Collection** : Collect state and input trajectory under control input

$$\mathcal{U}(t) = - \underbrace{\mathcal{L}_\Omega^{(0)}}_{\text{Stabilizing Gain}} \mathcal{X}(t) + \underbrace{I(t)}_{\text{Exploration Noise}} = \alpha(t).$$

- **IRL** : Once enough state and input trajectory data is collected, IRL is used to compute matrices $\mathcal{P}$, Ω and associated gains $\mathcal{L}_\mathcal{P}$ and $\mathcal{L}_\Omega$ model-free.
- **Mean-Field Approximation** : Representative agents are then used to approximate the mean-field trajectory $\bar{X}(t)$.



Data-Driven Solution

We adapt the Integral Reinforcement Learning for LQG MFGs developed by Xu et al. (2023) to networked multi-class problems.

Track one representative agent $a_{k,1}$ per class k and define :

$$\mathcal{X} = [x_{1,1}^\top, \dots, x_{K,1}^\top]^\top, \quad \mathcal{U} = [u_{1,1}^\top, \dots, u_{K,1}^\top]^\top.$$

Main steps :

- **Data Collection** : Collect state and input trajectory under control input

$$\mathcal{U}(t) = - \underbrace{\mathcal{L}_\Omega^{(0)}}_{\text{Stabilizing Gain}} \mathcal{X}(t) + \underbrace{I(t)}_{\text{Exploration Noise}} = \alpha(t).$$

- **IRL** : Once enough state and input trajectory data is collected, IRL is used to compute matrices $\mathcal{P}$, Ω and associated gains $\mathcal{L}_\mathcal{P}$ and $\mathcal{L}_\Omega$ model-free.
- **Mean-Field Approximation** : Representative agents are then used to approximate the mean-field trajectory $\bar{X}(t)$.



Data-Driven Solution

We adapt the Integral Reinforcement Learning for LQG MFGs developed by Xu et al. (2023) to networked multi-class problems.

Track one representative agent $a_{k,1}$ per class k and define :

$$\mathcal{X} = [x_{1,1}^\top, \dots, x_{K,1}^\top]^\top, \quad \mathcal{U} = [u_{1,1}^\top, \dots, u_{K,1}^\top]^\top.$$

Main steps :

- **Data Collection** : Collect state and input trajectory under control input

$$\mathcal{U}(t) = - \underbrace{\mathcal{L}_\Omega^{(0)}}_{\text{Stabilizing Gain}} \mathcal{X}(t) + \underbrace{I(t)}_{\text{Exploration Noise}} = \alpha(t).$$

- **IRL** : Once enough state and input trajectory data is collected, IRL is used to compute matrices $\mathcal{P}$, Ω and associated gains $\mathcal{L}_\mathcal{P}$ and $\mathcal{L}_\Omega$ model-free.
- **Mean-Field Approximation** : Representative agents are then used to approximate the mean-field trajectory $\bar{X}(t)$.



Data-Driven Solution

We adapt the Integral Reinforcement Learning for LQG MFGs developed by Xu et al. (2023) to networked multi-class problems.

Track one representative agent $a_{k,1}$ per class k and define :

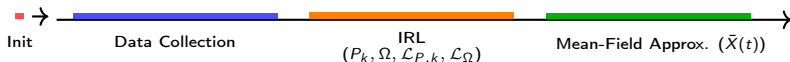
$$\mathcal{X} = [x_{1,1}^\top, \dots, x_{K,1}^\top]^\top, \quad \mathcal{U} = [u_{1,1}^\top, \dots, u_{K,1}^\top]^\top.$$

Main steps :

- **Data Collection** : Collect state and input trajectory under control input

$$\mathcal{U}(t) = - \underbrace{\mathcal{L}_\Omega^{(0)}}_{\text{Stabilizing Gain}} \mathcal{X}(t) + \underbrace{I(t)}_{\text{Exploration Noise}} = \alpha(t).$$

- **IRL** : Once enough state and input trajectory data is collected, IRL is used to compute matrices $\mathcal{P}$, Ω and associated gains $\mathcal{L}_\mathcal{P}$ and $\mathcal{L}_\Omega$ model-free.
- **Mean-Field Approximation** : Representative agents are then used to approximate the mean-field trajectory $\bar{X}(t)$.



Towards a Data-Driven Solution

Define value function ansatzs for P_k and Ω as

$$\Psi_{1,k}(t, x_{k,i}) = e^{-\rho t} x_{k,i}^\top P_k^{(\ell)} x_{k,i} \text{ for all } k, \quad (5a)$$

$$\Psi_2(t, \mathcal{X}) = e^{-\rho t} \mathcal{X}^\top \Omega^{(\ell)} \mathcal{X}. \quad (5b)$$

Applying Itô's formula to (5), they become

$$\begin{aligned} d\Psi_{1,k}(t, x_{k,i}) = & e^{-\rho t} \left[-\rho x_{k,i}^\top P_k^{(\ell)} x_{k,i} + 2(A_k x_{k,i} + B_k \alpha_{k,i})^\top P_k^{(\ell)} x_{k,i} + \text{Tr}(D_k D_k^\top P_k^{(\ell)}) \right] dt \\ & + 2e^{-\rho t} x_{k,i}^\top P_k^{(\ell)} D_k dw_{k,i}, \end{aligned} \quad (6a)$$

$$\begin{aligned} d\Psi_2(t, \mathcal{X}) = & e^{-\rho t} \left[-\rho \mathcal{X}^\top \Omega^{(\ell)} \mathcal{X} + 2(\mathcal{A}\mathcal{X} + \mathcal{B}\alpha)^\top \Omega_k \mathcal{X} + \text{Tr}(\mathcal{D}\mathcal{D}^\top \Omega^{(\ell)}) \right] dt \\ & + 2e^{-\rho t} \mathcal{X}^\top \Omega^{(\ell)} \mathcal{D}_k d\mathcal{W}. \end{aligned} \quad (6b)$$

Note

Both equations depend on matrices A_k and B_k and their block diagonal counterparts $\mathcal{A}$ and $\mathcal{B}$.

Idea

To remove the dependency, the Kleinman iteration is used.

Towards a Data-Driven Solution

Theorem - Kleinman (1968)

Consider a generic Algebraic Riccati Equation (ARE)

$$A^\top P^* + P^* A - P^* B R^{-1} B^\top P^* + Q = 0, \quad (7)$$

where the pair (A, B) is stabilizable, $R \succ 0$, and a unique stabilizing solution P^* exists.

Let $L^{(0)}$ be an initial stabilizing gain, and $P^{(\ell)}$ the symmetric positive definite solution of the Lyapunov equation

$$(A - BL^{(\ell-1)})^\top P^{(\ell)} + P^{(\ell)}(A - BL^{(\ell-1)}) + (L^{(\ell-1)})^\top R L^{(\ell-1)} + Q = 0, \quad (8)$$

where $L^{(\ell)} = R^{-1} B^\top P^{(\ell)}$. Then, the following hold:

1. $A - BL^{(\ell)}$ is Hurwitz,
2. $P^* \preceq P^{(\ell+1)} \preceq P^{(\ell)}$,
3. $\lim_{\ell \rightarrow \infty} L^{(\ell)} = L^*$, $\lim_{\ell \rightarrow \infty} P^{(\ell)} = P^*$.

□

Assumption 3

The matrix $\mathcal{Q}(I_N - \mathcal{H})$ is symmetric.

Under Assumptions 1 and 2, initially stabilizing gains $L_{P,k}^{(0)}$ and $\mathcal{L}_\Omega^{(0)}$ can be selected.

Under Assumption 3, by the Kleinman iteration, the AREs for P_k and Ω can be solved iteratively :

$$\begin{aligned} \rho P_k^{(\ell)} = & P_k^{(\ell)}(A_k - B_k L_{P,k}^{(\ell-1)}) + (A_k - B_k L_{P,k}^{(\ell-1)})^\top P_k^{(\ell)} \\ & + (L_{P,k}^{(\ell-1)})^\top R_k L_{P,k}^{(\ell-1)} + Q_k, \end{aligned} \quad (9)$$

where $L_{P,k}^{(\ell)} = R_k^{-1} B_k^\top P_k^{(\ell)}$ and

$$\begin{aligned} \rho \Omega^{(\ell)} = & \Omega^{(\ell)}(A - B \mathcal{L}_\Omega^{(\ell-1)}) + (A - B \mathcal{L}_\Omega^{(\ell-1)})^\top \Omega^{(\ell)} \\ & + (\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R} \mathcal{L}_\Omega^{(\ell-1)} + \mathcal{Q}(I_N - \mathcal{H}), \end{aligned} \quad (10)$$

with $\mathcal{L}_\Omega^{(\ell)} = \mathcal{R}^{-1} B^\top \Omega^{(\ell)}$.

Assumption 3

The matrix $\mathcal{Q}(I_N - \mathcal{H})$ is symmetric.

Under Assumptions 1 and 2, initially stabilizing gains $L_{P,k}^{(0)}$ and $\mathcal{L}_\Omega^{(0)}$ can be selected.

Under Assumption 3, by the Kleinman iteration, the AREs for P_k and Ω can be solved iteratively :

$$\begin{aligned} \rho P_k^{(\ell)} = & P_k^{(\ell)}(A_k - B_k L_{P,k}^{(\ell-1)}) + (A_k - B_k L_{P,k}^{(\ell-1)})^\top P_k^{(\ell)} \\ & + (L_{P,k}^{(\ell-1)})^\top R_k L_{P,k}^{(\ell-1)} + Q_k, \end{aligned} \quad (9)$$

where $L_{P,k}^{(\ell)} = R_k^{-1} B_k^\top P_k^{(\ell)}$ and

$$\begin{aligned} \rho \Omega^{(\ell)} = & \Omega^{(\ell)}(A - \mathcal{B} \mathcal{L}_\Omega^{(\ell-1)}) + (A - \mathcal{B} \mathcal{L}_\Omega^{(\ell-1)})^\top \Omega^{(\ell)} \\ & + (\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R} \mathcal{L}_\Omega^{(\ell-1)} + \mathcal{Q}(I_N - \mathcal{H}), \end{aligned} \quad (10)$$

with $\mathcal{L}_\Omega^{(\ell)} = \mathcal{R}^{-1} \mathcal{B}^\top \Omega^{(\ell)}$.

The resulting matrices $P_k^{(\ell)}$, $\Omega^{(\ell)}$, $L_{P,k}^{(\ell)}$ and $\mathcal{L}_\Omega^{(\ell)}$ satisfy the following properties:

1. $A_k - \frac{1}{2}\rho I_{n_k} - B_k L_{P,k}^{(\ell)}$ and $\mathcal{A} - \frac{1}{2}\rho I_N - \mathcal{B} \mathcal{L}_\Omega^{(\ell)}$ are Hurwitz,
2. $P_k^* \preceq P_k^{(\ell+1)} \preceq P_k^{(\ell)}$ and $\Omega^* \preceq \Omega^{(\ell+1)} \preceq \Omega^{(\ell)}$,
3. $\lim_{\ell \rightarrow \infty} L_{P,k}^{(\ell)} = L_{P,k}^*$, $\lim_{\ell \rightarrow \infty} P_k^{(\ell)} = P_k^*$,
 $\lim_{\ell \rightarrow \infty} \mathcal{L}_\Omega^{(\ell)} = \mathcal{L}_\Omega^*$, and $\lim_{\ell \rightarrow \infty} \Omega^{(\ell)} = \Omega^*$,

where P_k^* is the unique positive definite solution to the ARE for P_k and Ω^* the unique stabilizing solution to the ARE for Ω .

Substituting in the iterative procedure into the equations (6) for $d\Psi_{1,k}$ and $d\Psi_2$ yields

$$d\Psi_{1,k} = e^{-\rho t} \left[-x_{k,1}^\top ((L_{P,k}^{(\ell-1)})^\top R_k L_{P,k}^{(\ell-1)} - Q_k) x_{k,1} + 2(\alpha_{k,1} + L_{P,k}^{(\ell-1)} x_{k,1})^\top R_k L_{P,k}^{(\ell)} x_{k,1} \right. \\ \left. + \text{Tr}(D_k D_k^\top P_k^{(\ell)}) \right] dt + 2e^{-\rho t} x_{k,1}^\top P_k^{(\ell)} D_k dw_{k,1}, \quad (11a)$$

$$d\Psi_2 = e^{-\rho t} \left[-\mathcal{X}^\top ((\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R} \mathcal{L}_\Omega^{(\ell-1)} - \mathcal{Q}(I_N - \mathcal{H})) \mathcal{X} + 2(\alpha + \mathcal{L}_\Omega^{(\ell-1)} \mathcal{X})^\top \mathcal{R} \mathcal{L}_\Omega^{(\ell)} \mathcal{X} \right. \\ \left. + \text{Tr}(\mathcal{D} \mathcal{D}^\top \Omega^{(\ell)}) \right] dt + 2e^{-\rho t} \mathcal{X}^\top \Omega^{(\ell)} \mathcal{D} dW, \quad (11b)$$

which are now independent of system matrices A_k and input matrix B_k .

Integrating both sides of the equation for $d\Psi_{1,k}$ and $d\Psi_2$ over a short time interval $[t, t + \Delta t]$ and taking its expectation yields

$$\Delta\Psi_{1,k}^t = -\mathcal{I}_{q,k}^t + 2\mathcal{I}_{1,P,k}^t + \mathcal{I}_{2,P,k}^t, \quad (12a)$$

$$\Delta\Psi_2^t = -\mathcal{I}_K^t + 2\mathcal{I}_{1,\Omega}^t + \mathcal{I}_{2,\Omega}^t, \quad (12b)$$

where the terms for (12a) are defined by

$$\Delta\Psi_{1,k}^t = \mathbb{E}\left[e^{-\rho(t+\Delta t)}x_{k,1}^\top(t+\Delta t)P_k^{(\ell)}x_{k,1}(t+\Delta t) - e^{-\rho t}x_{k,1}^\top(t)P_k^{(\ell)}x_{k,1}(t)\right],$$

$$\mathcal{I}_{q,k}^t = \mathbb{E} \int_t^{t+\Delta t} e^{-\rho\tau} \left[x_{k,1}^\top \left((L_{P,k}^{(\ell-1)})^\top R_k L_{P,k}^{(\ell-1)} + Q_k \right) x_{k,1} \right] d\tau,$$

$$\mathcal{I}_{1,P,k}^t = \mathbb{E} \int_t^{t+\Delta t} e^{-\rho\tau} (\alpha_{k,1} + L_{P,k}^{(\ell-1)}x_{k,1})^\top R_k L_{P,k}^{(\ell)}x_{k,1} d\tau,$$

$$\mathcal{I}_{2,P,k}^t = \frac{1}{\rho} (e^{-\rho t} - e^{-\rho(t+\Delta t)}) \text{Tr}(D_k D_k^\top P_k^{(\ell)}).$$

The definition of the terms for (12b) are structurally identical, substituting

$$\hat{P}_k^{(\ell)} \rightarrow \hat{\Omega}^{(\ell)}, \quad x_{k,1} \rightarrow \mathcal{X}, \quad L_{P,k} \rightarrow \mathcal{L}_\Omega, \quad D_k \rightarrow \mathcal{D}, \quad R_k \rightarrow \mathcal{R}, \quad Q_k \rightarrow \mathcal{Q}(I_N - \mathcal{H}).$$

Data-Driven Solution

Equation (12) can be expressed using the Kronecker product representation. The following terms are defined for (12a)

$$\begin{aligned}
 \hat{P}_k^{(\ell)} &= [P_{k,11}^{(\ell)}, 2P_{k,12}^{(\ell)}, \dots, 2P_{k,1n_k}^{(\ell)}, P_{k,22}^{(\ell)}, \dots, P_{k,n_k n_k}^{(\ell)}]^\top \in \mathbb{R}^{\frac{n_k(n_k+1)}{2}}, \\
 \hat{x}_{k,1} &= [x_1^2, x_1 x_2, \dots, x_1 x_{n_k}, x_2^2, \dots, x_{n_k}^2]^\top \in \mathbb{R}^{\frac{n_k(n_k+1)}{2}}, \quad x_j = x_{k,1,j}, \\
 \delta_{P,k}^t &= \mathbb{E} \left[e^{-\rho(t+\Delta t)} \hat{x}_{k,1}(t+\Delta t) - e^{-\rho t} \hat{x}_{k,1}(t) \right] \in \mathbb{R}^{\frac{n_k(n_k+1)}{2}}, \\
 \mathcal{I}_{xx,k}^t &= \mathbb{E} \left[\int_t^{t+\Delta t} e^{-\rho \tau} (x_{k,1} \otimes x_{k,1}) d\tau \right] \in \mathbb{R}^{n_k^2}, \\
 \mathcal{I}_{x\alpha,k}^t &= \mathbb{E} \left[\int_t^{t+\Delta t} e^{-\rho \tau} (x_{k,1} \otimes \alpha_{k,1}) d\tau \right] \in \mathbb{R}^{n_k m_k}, \\
 \delta_\rho^t &= e^{-\rho(t+\Delta t)} - e^{-\rho t} \in \mathbb{R}, \\
 \theta_{P,k}^{(\ell)} &= \frac{1}{\rho} \text{Tr} \left(D_k D_k^\top P_k^{(\ell)} \right) \in \mathbb{R}.
 \end{aligned}$$

The terms for (12b) are structurally identical, substituting

$$\hat{P}_k^{(\ell)} \rightarrow \hat{\Omega}^{(\ell)}, \quad \hat{x}_{k,1} \rightarrow \hat{\mathcal{X}}, \quad x_{k,1} \rightarrow \mathcal{X}, \quad \alpha_{k,1} \rightarrow \alpha, \quad D_k \rightarrow \mathcal{D}, \quad P_k^{(\ell)} \rightarrow \Omega^{(\ell)},$$

and replacing dimensions $n_k \rightarrow N$, $m_k \rightarrow M$.

Data-Driven Solution

The integral equations (12) for a single time step ($[t, t + \Delta t]$), expressed using the Kronecker product representation, become

$$\begin{aligned} 0 = & (\delta_{P,k}^t)^\top \hat{P}_k^{(\ell)} + \delta_\rho^t \theta_{P,k}^{(\ell)} \\ & + (\mathcal{I}_{xx,k}^t)^\top \text{vec}((L_{P,k}^{(\ell-1)})^\top R_k L_{P,k}^{(\ell-1)} + Q_k) \\ & - 2 \left[(\mathcal{I}_{x\alpha,k}^t)^\top (I_{n_k} \otimes R_k) + (\mathcal{I}_{xx,k}^t)^\top (I_{n_k} \otimes (L_{P,k}^{(\ell-1)})^\top R_k) \right] \text{vec}(L_{P,k}^{(\ell)}), \end{aligned} \quad (13a)$$

$$\begin{aligned} 0 = & (\delta_{\Omega,k}^t)^\top \hat{\Omega}^{(\ell)} + \delta_\rho^t \theta_\Omega^{(\ell)} \\ & + (\mathcal{I}_{xx}^t)^\top \text{vec}((\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R} \mathcal{L}_\Omega^{(\ell-1)} + \mathcal{Q}(I_N - \mathcal{H})) \\ & - 2 \left[(\mathcal{I}_{x\alpha}^t)^\top (I_N \otimes \mathcal{R}) + (\mathcal{I}_{xx}^t)^\top (I_N \otimes (\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R}) \right] \text{vec}(\mathcal{L}_\Omega^{(\ell)}). \end{aligned} \quad (13b)$$

The integral equations (12) for a single time step $([t, t + \Delta t])$, expressed using the Kronecker product representation, become

$$\begin{aligned}
 0 = & (\delta_{P,k}^t)^\top \hat{P}_k^{(\ell)} + \delta_\rho^t \theta_{P,k}^{(\ell)} \\
 & + (\mathcal{I}_{xx,k}^t)^\top \text{vec}((L_{P,k}^{(\ell-1)})^\top R_k L_{P,k}^{(\ell-1)} + Q_k) \\
 & - 2 \left[(\mathcal{I}_{x\alpha,k}^t)^\top (I_{n_k} \otimes R_k) + (\mathcal{I}_{xx,k}^t)^\top (I_{n_k} \otimes (L_{P,k}^{(\ell-1)})^\top R_k) \right] \text{vec}(L_{P,k}^{(\ell)}),
 \end{aligned} \tag{13a}$$

$$\begin{aligned}
 0 = & (\delta_{\Omega,k}^t)^\top \hat{\Omega}^{(\ell)} + \delta_\rho^t \theta_\Omega^{(\ell)} \\
 & + (\mathcal{I}_{xx}^t)^\top \text{vec}((\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R} \mathcal{L}_\Omega^{(\ell-1)} + \mathcal{Q}(I_N - \mathcal{H})) \\
 & - 2 \left[(\mathcal{I}_{x\alpha}^t)^\top (I_N \otimes \mathcal{R}) + (\mathcal{I}_{xx}^t)^\top (I_N \otimes (\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R}) \right] \text{vec}(\mathcal{L}_\Omega^{(\ell)}).
 \end{aligned} \tag{13b}$$

Stacking (13) over $l \in \mathbb{N}$ time steps associated with real-time data

► For the equation for P_k

$$0 = \Delta_{1k} \hat{P}_k^{(\ell)} + \Delta_{2k} \text{vec}(L_{P,k}^{(\ell)}) + \Delta_6 \theta_{P,k}^{(\ell)} - \Delta_{4k}, \quad (14)$$

$$\Delta_{1k} = [\delta_{P,k}^{t_1}, \dots, \delta_{P,k}^{t_l}]^\top \in \mathbb{R}^{l \times \frac{n_k(n_k+1)}{2}},$$

$$\begin{aligned} \Delta_{2k} = & -2[\mathcal{I}_{xx,k}^{t_1}, \dots, \mathcal{I}_{xx,k}^{t_l}]^\top (I_{n_k} \otimes (L_{P,k}^{(\ell-1)})^\top R_k) \\ & - 2[\mathcal{I}_{x\alpha,k}^{t_1}, \dots, \mathcal{I}_{x\alpha,k}^{t_l}]^\top (I_{n_k} \otimes R_k) \in \mathbb{R}^{l \times m_k n_k}, \end{aligned}$$

$$\Delta_{4k} = -[\mathcal{I}_{xx,k}^{t_1}, \dots, \mathcal{I}_{xx,k}^{t_l}]^\top \text{vec}((L_{P,k}^{(\ell-1)})^\top R_k L_{P,k}^{(\ell-1)} + Q_k) \in \mathbb{R}^l.$$

$$\Delta_6 = [\delta_\rho^{t_1}, \dots, \delta_\rho^{t_l}]^\top \in \mathbb{R}^l.$$

- Similarly, for the equation for Ω

$$0 = \Delta_1 \hat{\Omega}_k^{(\ell)} + \Delta_3 \text{vec}(\mathcal{L}_\Omega^{(\ell)}) + \Delta_6 \theta_\Omega^{(\ell)} - \Delta_5 \quad (15)$$

$$\Delta_1 = [\delta_\Omega^{t_1}, \dots, \delta_\Omega^{t_I}]^\top \in \mathbb{R}^{I \times \frac{N(N+1)}{2}},$$

$$\begin{aligned} \Delta_3 = & -2[\mathcal{I}_{XX}^{t_1}, \dots, \mathcal{I}_{XX}^{t_I}]^\top (I_N \otimes (\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R}) \\ & - 2[\mathcal{I}_{X\alpha}^{t_1}, \dots, \mathcal{I}_{X\alpha}^{t_I}]^\top (I_N \otimes \mathcal{R}) \in \mathbb{R}^{I \times NM}, \end{aligned}$$

$$\Delta_5 = -[\mathcal{I}_{XX}^{t_1}, \dots, \mathcal{I}_{XX}^{t_I}]^\top \cdot \text{vec}((\mathcal{L}_\Omega^{(\ell-1)})^\top \mathcal{R} \mathcal{L}_\Omega^{(\ell-1)} + \mathcal{Q}(I_N - \mathcal{H})) \in \mathbb{R}^I.$$

Data-Driven Solution

Δ matrices are computed using trajectory **data** and are thus known. The integral equations (14), (15)

$$0 = \Delta_{1k} \hat{P}_k^{(\ell)} + \Delta_{2k} \text{vec}(L_{P,k}^{(\ell)}) + \Delta_6 \theta_{P,k}^{(\ell)} - \Delta_{4k}$$

$$0 = \Delta_1 \hat{\Omega}^{(\ell)} + \Delta_3 \text{vec}(\mathcal{L}_{\Omega}^{(\ell)}) + \Delta_6 \theta_{\Omega}^{(\ell)} - \Delta_5.$$

can then be reformulated as the matrix forms

$$\underbrace{\begin{bmatrix} \Delta_{1k} & \Delta_{2k} & \Delta_6 \end{bmatrix}}_{\Xi_{1,k}} \begin{bmatrix} \hat{P}_k^{(\ell)} \\ \text{vec}(L_{P,k}^{(\ell)}) \\ \theta_{P,k}^{(\ell)} \end{bmatrix} = \Delta_{4k}, \quad (16a)$$

$$\underbrace{\begin{bmatrix} \Delta_1 & \Delta_3 & \Delta_6 \end{bmatrix}}_{\Xi_2} \begin{bmatrix} \hat{\Omega}^{(\ell)} \\ \text{vec}(\mathcal{L}_{\Omega}^{(\ell)}) \\ \theta_{\Omega}^{(\ell)} \end{bmatrix} = \Delta_5. \quad (16b)$$

where $\Xi_{1,k}$, Ξ_2 , Δ_{4k} and Δ_5 are constructed from **data**, and $\hat{P}_k^{(\ell)}$, $\text{vec}(L_{P,k}^{(\ell)})$, $\theta_{P,k}^{(\ell)}$, $\hat{\Omega}^{(\ell)}$, $\text{vec}(\mathcal{L}_{\Omega}^{(\ell)})$, and $\theta_{\Omega}^{(\ell)}$ are the unknowns.

Assumption 4 (Persistence of Excitation)

There exists $L > 0$ such that for $l \geq L$, the matrices

$$\Phi_k = \begin{bmatrix} \mathcal{I}_{xx,k}^{t_1} & \cdots & \mathcal{I}_{xx,k}^{t_l} \\ \mathcal{I}_{x\alpha,k}^{t_1} & \cdots & \mathcal{I}_{x\alpha,k}^{t_l} \\ \delta_\rho^{t_1} & \cdots & \delta_\rho^{t_l} \end{bmatrix} \quad \text{and} \quad \Phi = \begin{bmatrix} \mathcal{I}_{xx}^{t_1} & \cdots & \mathcal{I}_{xx}^{t_l} \\ \mathcal{I}_{x\alpha}^{t_1} & \cdots & \mathcal{I}_{x\alpha}^{t_l} \\ \delta_\rho^{t_1} & \cdots & \delta_\rho^{t_l} \end{bmatrix}$$

satisfy $\text{rank}(\Phi_k) = \frac{n_k(n_k+1)}{2} + m_k n_k + 1$ for all k , and $\text{rank}(\Phi) = \frac{N(N+1)}{2} + NM + 1$.

This ensures (16) has unique solutions, and can be satisfied in practice by injecting exploration noise during learning.

Algorithm 1: Mean Field Games IRL

■ Initialization:

Choose initial stabilizing gains $\mathcal{L}_{\Omega}^{(0)}$ and $L_{P,k}^{(0)}$
Select representative agents $a_{k,1}$ and threshold ϵ

■ Data Collection:

Apply control $\mathcal{U}(t) = -\mathcal{L}_{\Omega}^{(0)}\mathcal{X}(t) + I(t)$
Collect data until **Persistence of Excitation** is met

■ Policy Iteration:

repeat

 Solve for **local parameters** $(P_k, L_{P,k}, \theta_{P,k})$ via Eq. (16a)

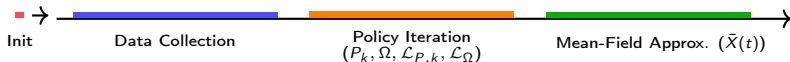
 Solve for **global parameters** $(\Omega, \mathcal{L}_{\Omega}, \theta_{\Omega})$ via Eq. (16b)

$\ell \leftarrow \ell + 1$

until $\|\Delta\mathcal{P}\| \leq \epsilon$ and $\|\Delta\Omega\| \leq \epsilon$

■ Mean-Field Trajectory Approx.:

The mean-field trajectory is approximated by the empirical mean over N_S trajectories of the representative agents with $\mathcal{U}(t) = -\mathcal{L}_{\Omega}\mathcal{X}(t)$ and $\mathbb{E}[\mathcal{X}(0)] = \bar{X}_0$.



Numerical Example

Table 1: Class-specific dynamics and cost parameters for multi-class MFG.

Parameter	Class 1	Class 2	Class 3
A_k	$\begin{bmatrix} 0 & 10 \\ -10 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -5 \end{bmatrix}$	$\begin{bmatrix} 0 & -4 \\ 3 & -6 \end{bmatrix}$
B_k	$\begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \\ 0 & 1.0 \\ 1.0 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.8 \\ 3.0 \end{bmatrix}$
D_k	$0.1I_2$	$0.1I_3$	$0.1I_2$
Q_k	$\text{diag}(20, 10)$	$\text{diag}(10, 15, 20)$	$\text{diag}(30, 20)$
R_k	0.8	$\text{diag}(0.5, 0.7)$	0.6
ρ_k	0.1	0.1	0.1

Numerical Example

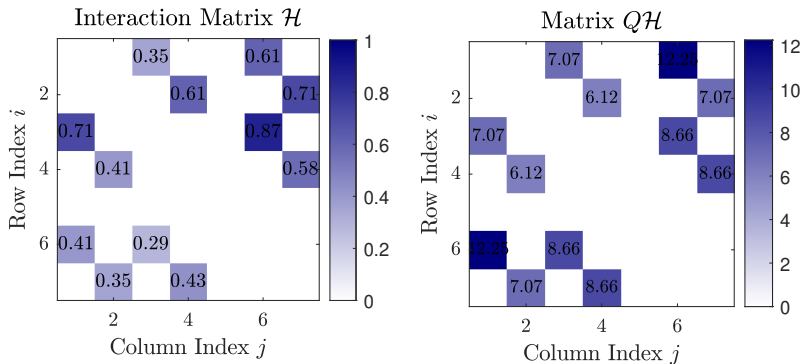


Figure 2: Coupling matrix $\mathcal{H}$ and associated symmetric $Q\mathcal{H}$

Matrix $\mathcal{H}$ is chosen such that $Q\mathcal{H}$ is symmetric and thus satisfies Assumption 3.

Numerical Example

During data collection, representative agents $a_{k,1}$ are simulated 20 times for 20s. Exploration noise consists of a sum of sinusoids ($N = 500$, $A = 25$, $f \in \mathcal{U}[-100, 100]$).

Table 2: Learned (IRL) vs. Ground Truth (MATLAB care()) Results for $(P_k, L_{P,k})$

Class	Parameter	Learned (IRL)	Ground Truth
1	$L_{P,1}$	[3.9628, 2.7432]	[3.9608, 2.7376]
	P_1	$\begin{bmatrix} 2.8162 & 0.3592 \\ 0.3592 & 1.8363 \end{bmatrix}$	$\begin{bmatrix} 2.8102 & 0.35843 \\ 0.35843 & 1.8316 \end{bmatrix}$
2	$L_{P,2}$	$\begin{bmatrix} -0.42082 & -0.67709 & 2.9372 \\ 3.6708 & 5.5758 & 0.56791 \end{bmatrix}$	$\begin{bmatrix} -0.42334 & -0.67943 & 2.9314 \\ 3.6677 & 5.5738 & 0.5616 \end{bmatrix}$
	P_2	$\begin{bmatrix} 13.4200 & 2.6777 & -0.20979 \\ 2.6777 & 4.0741 & -0.33665 \\ -0.20979 & -0.33665 & 1.4692 \end{bmatrix}$	$\begin{bmatrix} 13.4070 & 2.6732 & -0.21167 \\ 2.6732 & 4.0715 & -0.33972 \\ -0.21167 & -0.33972 & 1.4657 \end{bmatrix}$
3	$L_{P,3}$	[-3.6898, 6.0468]	[-3.6908, 6.0446]
	P_3	$\begin{bmatrix} 10.2430 & -3.4719 \\ -3.4719 & 2.1364 \end{bmatrix}$	$\begin{bmatrix} 10.2340 & -3.4672 \\ -3.4672 & 2.1335 \end{bmatrix}$

Numerical Example

Table 3: Comparison of Selected Parameters for $(\Omega, \mathcal{L}_\Omega)$

Parameter	Learned (IRL)	Ground Truth	Error
$\Omega_{1,1}$	2.7121	2.7111	0.0010
$2\Omega_{1,2}$	0.7515	0.7514	0.0001
$2\Omega_{1,3}$	-1.2599	-1.2599	0.0000
$\Omega_{6,6}$	9.5690	9.5677	0.0013
$\mathcal{L}_{\Omega,1,1}$	3.8569	3.8585	0.0016
$\mathcal{L}_{\Omega,1,2}$	2.7079	2.7032	0.0047
$\mathcal{L}_{\Omega,2,3}$	-0.4585	-0.4515	0.0070
$\mathcal{L}_{\Omega,3,6}$	-1.3606	-1.3572	0.0034

Numerical Example

For validation, a finite population of 50 agents per class is simulated, with all initial states located uniformly in $[0.5, 1.5]$.

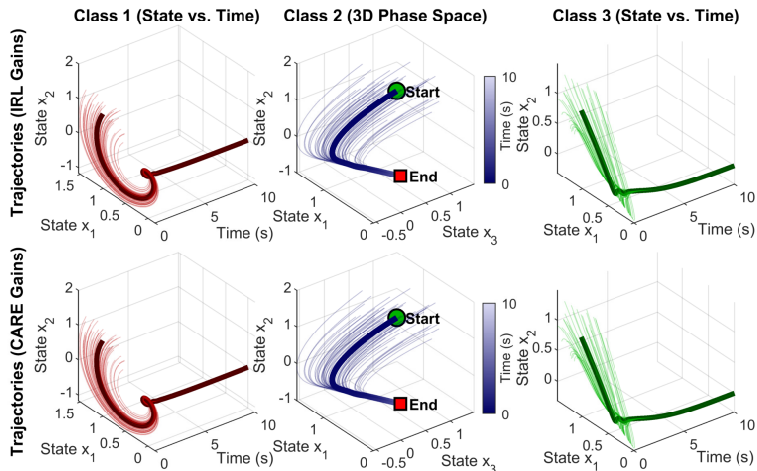


Figure 3: State trajectories (Classes 1 & 3) and 3D phase space representation (Class 2) simulated under data-driven learned controls (top row) versus MATLAB's `care()` gains (bottom row).

Conclusion

- ▶ Solution for LQG mean field games with heterogeneous classes.
- ▶ Data-driven solution with unknown dynamics.

Further investigations :

- ▶ Extend to more general networks, offset, price signal tracking, finite time horizon problems and non-linear dynamics.
- ▶ Applications to energy systems (EV charging, coordination of energy users,...).

References

- Huang, M. and Zhou, M. (2020). Linear quadratic mean field games: Asymptotic solvability and relation to the fixed point approach. *IEEE Transactions on Automatic Control*, 65(4), 1397–1412.
- Huang, M., Caines, P., and Malhamé, R. (2007). Large-population cost-coupled LQG problems with nonuniform agents: Individual-mass behavior and decentralized ϵ -Nash equilibria. *IEEE Transactions on Automatic Control*, 52(9), 1560–1571.
- Jiang, Y. & Jiang, Z.-P. (2012). Computational adaptive optimal control for continuous-time-linear systems with completely unknown dynamics. *Automatica*, 48(10), 2699–2704.
- Kleinman, D. (1968). On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1), 114–115.
- Xu, Z., Chen, J., Wang, B.C., and Shen, T. (2025a). Data-driven mean field equilibrium computation in large-population LQG games. *IEEE Transactions on Control of Network Systems*, 12(4), 2713–2725.
- Xu, Z., Shen, T., and Huang, M. (2023). Model-free policy iteration approach to NCE-based strategy design for linear quadratic Gaussian games. *Automatica*, 155, 111162.
- Zhu, J. and Gao, S. (2026). Data-Driven Network LQG Mean Field Games with Heterogeneous Populations via Integral Reinforcement Learning. *arXiv*, 2602.14339.

Appendix : Mean-Field Trajectory Approximation

The mean-field trajectory is approximated online using the learned $\mathcal{L}_\Omega$. The representative agents are run N_s times with input $\mathcal{U} = \mathcal{L}_\Omega \mathcal{X}$. For the j -th run, the dynamics are

$$d\mathcal{X}^j(t) = (\mathcal{A} - \mathcal{B}\mathcal{L}_\Omega)\mathcal{X}^j(t)dt + \mathcal{D}dw(t), \quad \mathcal{X}^j(0) = \bar{X}_0$$

Taking the empirical mean over N_s sample trajectories

$$\mathcal{X}^{(N_s)}(t) \triangleq \frac{1}{N_s} \sum_{j=1}^{N_s} \mathcal{X}^j(t)$$

and by the law of large number,

$$\lim_{N_s \rightarrow \infty} \mathcal{X}^{(N_s)}(t) = \bar{X}(t), \quad \forall t \geq 0.$$