NEURAL PARAMETER CALIBRATION FOR FINITE-STATE MEAN FIELD GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Mean field games efficiently approximate a very large population of strategic agents. While these games can aid the understanding of complex systems, their deployment in real-world settings is challenged by the specification of their parameters: mean field games (MFGs) often involve hidden preferences, constraints, and interactions that can rarely be theoretically derived or directly observed. To address this gap, we present a neural network-based framework for learning parametric, finite-state MFGs from observed population dynamics. To do so, we formulate the parameter calibration as an inverse problem and use implicit differentiation to backpropagate through the games' equilibrium. The resulting approach is fully differentiable and enables us to estimate flexible trajectory-wise parameter paths, including state- and time-dependent specifications without requiring observations of the individual agents' actions or rewards. We provide a proof for the exactness of the gradient computation in a discrete-time formulation. We validate our framework through numerical experiments across four systems of increasing complexity, ranging from synthetic linear-quadratic benchmarks to real-world urban mobility datasets.

1 INTRODUCTION

Mean field games (MFGs) offer a powerful mathematical framework for analyzing the strategic behavior of massive populations of interacting agents. By reducing the intractable complexity of a large number of individual interactions to a tractable problem involving a single representative agent and a macroscopic population distribution, MFGs have enabled important progress in modelling large-scale systems. Originally introduced independently by Lasry and Lions Lasry & Lions (2007) and Huang, Malhamé, and Caines Huang et al. (2006), this paradigm has been successfully applied across diverse domains, including urban traffic routing Chevalier et al. (2015); Festa & Göttlich (2018); Huang et al. (2019); Tanaka et al. (2021), epidemic spread Hubert & Turinici (2018); Doncel et al. (2022); Lee et al. (2021); Aurell et al. (2022b); Buckley & Vladimirsky (2025); Liu et al. (2025); Liu & Dayanıklı (2026), financial markets Cardaliaguet & Lehalle (2018); Carmona (2020); Carmona & Laurière (2023), and cybersecurity Kolokoltsov & Bensoussan (2016); Qian et al. (2020); Shen et al. (2025). While a broad part of the literature traditionally considers MFGs in continuous state-action spaces Bensoussan et al. (2013); Carmona & Delarue (2018), there has been growing interest in discrete-time, discrete-space settings Anahtarci et al. (2021); Yongacoglu et al. (2022); Yardim et al. (2023) and finite-state, continuous-time MFGs, which are highly suited for discrete decision-making processes Gomes et al. (2013).

Despite their theoretical elegance and advances, the deployment of MFGs in real-world scenarios remains bottlenecked by the challenge of parameter specification. Constructing an accurate MFG requires defining complex structural parameters, such as hidden preference costs, transition constraints, and interaction rates. In real-world environments, these parameters are abstract and rarely directly observable, barring the data-driven adoption of MFGs. While purely descriptive computational models or finite-agent simulations can often be calibrated from historical data Wang & Meng (2021); Hirano & Izumi (2022); Gaskin et al. (2023); Giampiccolo et al. (2024), they inherently lack the causal, strategic, and forward-looking structure that MFGs provide. Consequently, a rigorous methodology to calibrate the structural MFG parameters from observed population data forms a valuable addition to the current modelling landscape.

Contributions. We introduce a neural network-based framework for learning parametric MFGs entirely from macroscopic trajectory data. An overview of the method is presented in Fig. 1. We formulate the parameter calibration as an inverse problem: by treating the fixed-point equilibrium solver

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

as an implicit neural network component, our method uses implicit differentiation to backpropagate through the MFG equilibrium. This allows us to learn highly flexible, time-dependent representations of the game's underlying parameters without observing individual agent actions or rewards. Our main contributions are summarized as follows:

- We propose a novel, fully differentiable approach that estimates trajectory-wise parameter paths obtained by evaluating a **time- and mean-field-dependent neural map** on **observed population-density trajectories**, using implicit differentiation through the equilibrium fixed point.
- We provide **theoretical foundations** supporting the method under suitable conditions, including the exactness of the adjoint gradient computation, the consistency of a computational decoupling strategy, and a uniform generalization bound.
- We validate the framework on **four systems of increasing complexity**, demonstrating that it **(1)** accurately calibrates MFG parameters from both synthetic and real data, **(2)** outperforms a mean-field dynamics baseline lacking game-theoretic structure, and **(3)** maintains robustness to noise and dimensionality scaling in synthetic benchmarks.

Because the proposed calibration method only considers the mean field observations, it is not guaranteed to solve the inverse problem. Instead, it is our aim to accurately predict future mean field flows under varying initial conditions, which is key for many applications.

Related Work. Our approach intersects with several active areas of research. Inverse Reinforcement Learning (IRL) aims to recover reward functions from agent behavior Chen et al. (2022; 2021); Anahtarci et al. (2024; 2025b); Alkır & Saldi (2025); Anahtarci et al. (2025a). It is important to clarify that, unlike standard IRL, our method is not model-free: we assume the model is known up to some parameterization. However, because these parameters can depend on both time and the mean field distribution, learning them remains a challenging task. In the context of MFGs, recent advances in deep learning have primarily focused on solving the forward problem—computing equilibria for known parameters using neural networks or operator learning Cohen et al. (2024); Hofgard et al. (2026)—or exploring neural ordinary differential equations or model-based reinforcement learning for forward MFGs Thöni et al. (2026); Huang et al. (2024). While classical parameter estimation has been studied for standard partial differential equations and multi-agent simulations Gaskin et al. (2023), the calibration of MFGs from macroscopic density data remains largely unexplored. We contribute to filling this gap by considering the continuous-time, finite-state case.

Paper Outline. The remainder of this paper is structured as follows. Sec. 2 introduces the background of finite-state MFGs. Sec. 3 details our parameter calibration methodology and its theoretical foundations. Sec. 4 presents the test cases we study. Sec. 5 provides the experimental evaluation across synthetic and real-world datasets. We conclude in Sec. 6.

Code Availability. All code for the data generation and implementation will be made publicly available. Experimental details are also provided in the appendices.

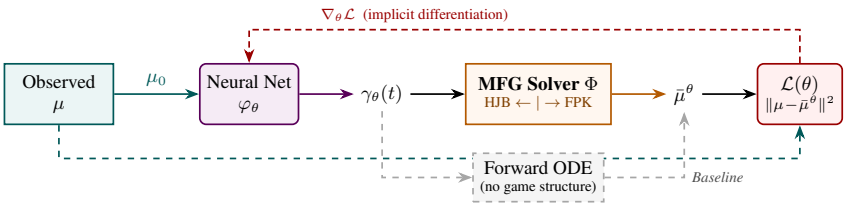


Figure 1: **Method overview.** The MFG solver Φ (coupled HJB and FPK equations) is parameterized by the network φ_θ, which predicts $\gamma_\theta(t)$ based on the initial value of the observed mean field flow μ_0. Gradients flow back via implicit differentiation through the fixed point (dashed red path). The dashed gray path shows the forward-only baseline that omits the solver process as it does not incorporate any game structure.

2 Background

General Notation. Throughout this work, we use the following notation. Let $\mathbb{N} = \{0, 1, \dots\}$ denote the set of natural numbers and $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. For any $k \in \mathbb{N}$, we define $[k] = \{1, \dots, k\}$.

Let $\mathbb{R}$ denote the set of real numbers. For any $d \in \mathbb{N}^*$, $\mathbb{R}^d$ denotes the set of real-valued vectors of dimension d, and $\mathbb{R}^{d\times d}$ the set of real-valued square matrices of size d. Let $\mathbb{M}^d = \{A \in \mathbb{R}^{d\times d} \mid A[x,y] \geq 0$ for any distinct x, y, and for any $x \in [d]$, $A[x,x] = -\sum_{y\in[d],\, y\neq x} A[x,y]\}$ denote the set of transition rate matrices (infinitesimal generators).

Finite-state Mean Field Games. We consider MFGs with a finite-state space as introduced by Gomes et al. (2013). Specifically, we let $[d] := \{1, \ldots, d\}$ denote the state space. We consider a continuous-time horizon $t \in [0, T]$ with $T > 0$. Let $\mathcal{A}$ be a Polish action space and $\mathcal{P}([d])$ the set of probability measures on $[d]$ (the $(d-1)$-simplex). The choice of $\mathcal{A}$ is model-dependent. A control is a function $\alpha : [0, T] \times [d] \mapsto \mathcal{A}$. We restrict our attention to admissible controls, i.e., Borel-measurable feedback maps $\alpha : [0, T] \times [d] \to \mathcal{A}$ such that, for every population flow μ, the induced transition rates define a valid generator in $\mathbb{M}^d$ and the cost $J^{\gamma,\mu_0}(\alpha; \mu)$ is finite. We denote by $\gamma : [0, T] \times \mathcal{P}([d]) \to \mathbb{R}^n$ the dynamic parameters of the game, belonging to a space of continuous functions Γ. In practice, we may consider more restrictive parameter classes, such as constant parameters $\gamma \in \mathbb{R}^n$ or time-only functions $\gamma : [0, T] \to \mathbb{R}^n$, which are special cases of this general formulation. The game is defined by: a transition rate matrix $Q : \mathbb{R}^n \times \mathcal{A} \times \mathcal{P}([d]) \to \mathbb{M}^d$, a running cost $f : \mathbb{R}^n \times [d] \times \mathcal{A} \times \mathcal{P}([d]) \to \mathbb{R}$, and a terminal cost $g : \mathbb{R}^n \times [d] \times \mathcal{P}([d]) \to \mathbb{R}$. For a specific parameter function $\gamma \in \Gamma$ and values $(t, x, a, \mu) \in [0, T] \times [d] \times \mathcal{A} \times \mathcal{P}([d])$, we define the evaluated parametric functions as:

$$Q^\gamma(t, a, \mu) := Q(\gamma(t, \mu), a, \mu),\ f^\gamma(t, x, a, \mu) := f(\gamma(t, \mu), x, a, \mu),\ g^\gamma(x, \mu) := g(\gamma(T, \mu), x, \mu). \quad (2.1)$$

We denote by $\mathcal{M}^\gamma := (d, \mathcal{A}, T, Q^\gamma, f^\gamma, g^\gamma)$ a parametric MFG, and by $(\mathcal{M}^\gamma)_{\gamma\in\Gamma}$ the family.

Controls, State Transitions and Markovian Dynamics. Analogous to general MFG theory Lasry & Lions (2007); Huang et al. (2006), finite-state MFGs consider a representative player interacting with a given distribution of agents $\mu : [0, T] \to \mathcal{P}([d])$. The representative agent influences their transition between states. This process is governed by the transition matrix $Q^\gamma(t, \alpha_t, \mu_t) \in \mathbb{M}^d$, which depends on the parameter function γ, the player's control α, and the distribution over states μ_t. For each $x \neq y \in [d]$, each matrix entry $Q^\gamma(t, \alpha_t, \mu_t)[x, y]$ describes the transition rate from state x to y at time t. For the parameter function γ and given an initial distribution $\mu_0 \in \mathcal{P}([d])$, the representative agent controls their state $X_t^{\mu_0,\alpha,\gamma}$ following:

$$\mathbb{P}\big(X_{t+h}^{\gamma,\mu_0,\alpha} = y \mid X_t^{\gamma,\mu_0,\alpha} = x\big) = h\, Q^\gamma(t, \alpha_t, \mu_t)[x, y] + o(h), \quad h \to 0^+, \quad (2.2)$$

where $X_0^{\gamma,\mu_0,\alpha} \sim \mu_0$. The state process $(X_t^{\gamma,\mu_0,\alpha})_{t\in[0,T]}$ is a continuous-time Markov chain with initial law μ_0, transition rates Q^γ and control α. It implicitly depends on $\mu = (\mu_t)_{t\in[0,T]}$.

The Expected Cost and Mean Field Equilibrium. Throughout the game, the representative player aims to minimize their expected cost, a nonlinear function that depends on both the player's control as well as the population distribution $\mu_t \in \mathcal{P}([d])$ Kolokoltsov (2010). The total expected cost J is the sum of a running cost f and a terminal cost g. If the representative player uses control α, its expected cost is

$$J^{\gamma,\mu_0}(\alpha; \mu) = \mathbb{E}\Big[\int_0^T f^\gamma\left(s, X_s^{\gamma,\mu_0,\alpha}, \alpha_s(X_s^{\gamma,\mu_0,\alpha}), \mu_s\right) ds + g^\gamma(X_T^{\gamma,\mu_0,\alpha}, \mu_T)\Big]. \quad (2.3)$$

Because the representative player cannot directly influence the population distribution μ, the player solves a standard optimal control problem, parameterized by μ (and γ).

At the Nash equilibrium, the distribution of the population and the representative player are the same.
Definition 2.1. *For $\gamma \in \Gamma$, a Nash equilibrium of the parametric MFG $\mathcal{M}^\gamma$, associated with an initial distribution $\mu_0 \in \mathcal{P}([d])$, is a pair $(\bar{\alpha}^{\gamma,\mu_0}, \bar{\mu}^{\gamma,\mu_0})$ such that: **(1)** $\bar{\alpha}^{\gamma,\mu_0}$ minimizes the cost functional $J^{\gamma,\mu_0}(\cdot\,; \bar{\mu}^{\gamma,\mu_0})$ and **(2)** for every $t \in [0, T]$ it holds that $\bar{\mu}_t^{\gamma,\mu_0} = Law(X_t^{\gamma,\mu_0,\bar{\alpha}})$.*

The second requirement, known as the consistency condition Caines et al. (2017), is one of the features distinguishing the MFG equilibrium from a mean field control problem, in which a single decision maker controls the mean field.

Characterizing the Mean Field Equilibrium: the Forward-Backward System. To characterize the equilibrium introduced in Definition 2.1, the MFG $\mathcal{M}^\gamma$ is described by a coupled system consisting of a Hamilton–Jacobi–Bellman equation (HJB) and a Fokker–Planck–Kolmogorov equation (FPK). The HJB equation encodes the optimization problem of the representative player, runs backward in time, and describes the value function $\bar{u}^{\gamma,\mu_0}$. The FPK equation encodes the consistency condition,

runs forward in time, and governs the evolution of the population distribution μ^{γ,μ_0}. More precisely, the following system of equations characterizes the equilibrium (see (Carmona & Delarue, 2018, Section 7.2) for details):

$$\begin{cases} -\partial_t \bar{u}_t^{\gamma,\mu_0}(x) = H^\gamma\big(t, x, \bar{\mu}_t^{\gamma,\mu_0}, \bar{u}_t^{\gamma,\mu_0}, \bar{\alpha}_t^{\gamma,\mu_0}(x)\big) & (i) \\ \partial_t \bar{\mu}_t^{\gamma,\mu_0}(x) = \sum_{y\in[d]} \bar{\mu}_t^{\gamma,\mu_0}(y)\, Q^\gamma\big(t, \bar{\alpha}_t^{\gamma,\mu_0}(y), \bar{\mu}_t^{\gamma,\mu_0}\big)[y,x] & (ii) \\ \bar{\mu}_{|t=0}^{\gamma,\mu_0} = \mu_0, \quad \bar{u}_T^{\gamma,\mu_0}(x) = g^\gamma(x, \bar{\mu}_T^{\gamma,\mu_0}), \quad x \in [d], & (iii) \end{cases} \tag{2.4}$$

where the equilibrium (feedback) control is $\bar{\alpha}_t^{\gamma,\mu_0}(x) = \arg\min_{a\in\mathcal{A}} H^\gamma\big(t, x, \bar{\mu}_t^{\gamma,\mu_0}, \bar{u}_t^{\gamma,\mu_0}(\cdot), a\big)$, and the discrete-gradient form of the Hamiltonian H^γ is defined for $p \in \mathbb{R}^d$ by:

$$H^\gamma(t, x, \mu, p, a) = f^\gamma(t, x, a, \mu) + \sum_{y\neq x} Q^\gamma(t, a, \mu)[x, y]\,(p_y - p_x). \tag{2.5}$$

This is because $Q^\gamma(t, a, \mu) \in \mathbb{M}^d$ has rows summing to zero, such that $\sum_{y\in[d]} Q^\gamma(t, a, \mu)[x, y]\, p_y = \sum_{y\neq x} Q^\gamma(t, a, \mu)[x, y]\,(p_y - p_x)$. Intuitively, the Hamiltonian combines the instantaneous cost f^γ with the expected variation of the value function induced by the controlled transitions Q^γ. In the sequel, we may omit γ or μ_0 from the superscript when the context is clear.

Objectives. From here, our goal is to propose a method that achieves three main objectives: **(1) calibrate** the parameterized MFG model $\mathcal{M}_\gamma$ by finding a suitable γ so that the mean-field flow matches the observations, **(2) highlight the forward-looking** quality of the mean field game by outperforming an approach based solely on mean-field dynamics, and **(3) maintain robustness** against noisy data, (reasonably) out-of-distribution samples, and novel scenarios.

3 Method

We describe our algorithm for finding the mean field equilibrium that reflects the learned parameters γ in Sec. 3.1. A theoretical analysis considering the contractivity of our approach, uniqueness of solutions, and associated approximation error is provided in App. B.

3.1 Estimating MFG Parameters

General Idea. We estimate the parameters γ_θ to minimize the discrepancy between observed mean field trajectories and the predicted mean field trajectories $\bar{\mu}^{\gamma_\theta}$, where $\bar{\mu}^{\gamma_\theta}$ is the mean-field trajectory at equilibrium of the MFG parametrized by γ_θ and starting with the same initial condition as the observation.

More formally, let $\mathcal{D} = \{(\mu_t^i)_{t\in[0,T]}\}_{1\leq i\leq N_{\text{traj}}}$ be a set of mean field trajectories generated from an unknown ground-truth parameter γ^* and possibly from different initial distributions $(\mu_0^i)_{1\leq i\leq N_{\text{traj}}}$. These trajectories may be inexact observations of Nash equilibrium flows due to noise in the data. γ^* belongs to the subset of parameters $\Gamma \subset \mathbb{R}^p$. The calibration problem associated with the MFG model $\{\mathcal{M}_\gamma : \gamma \in \Gamma\}$ consists of finding $\hat{\gamma} \in \arg\min_{\gamma\in\Gamma} \sum_i \|\Phi(\gamma, \mu_0^i) - \mu^i\|_2$, where Φ maps a parameter and an initial distribution to the corresponding mean field trajectory. To calibrate towards γ^*, we train a neural network φ_θ. In general, the network $\varphi_\theta : [0, T] \times \mathcal{P}([d]) \to \mathbb{R}^n$ returns time- and distribution-dependent parameter estimates; simpler architectures (constant or time-only) are obtained by restricting the input; see Sec. 5.

Learning the Parameter Estimates. We approximate γ^* by a neural network φ_θ with parameters θ. We train the parameters to minimize a squared L_2 discrepancy between observed mean field trajectories and generated mean field trajectories after solving the MFG corresponding to our current parameter estimate. In practice, we use a minibatch and a random time sub-interval in $[0, T]$ to improve generalization and avoid getting stuck in local loss minima. For an interval $I \subset [0, T]$, we use the continuous-time notation $\|\mu - \mu'\|^2_{L^2(I)} := \int_I \sum_{x\in[d]} (\mu_t(x) - \mu'_t(x))^2\, dt$, which is approximated by a discrete-time sum in the implementation. The average loss over a minibatch of size M is

$$\mathcal{L}(\theta) = \tfrac{1}{M} \sum_{i=1}^M \left\|\mu^i - \bar{\mu}^{i,\theta}\right\|^2_{L^2([t_i, t_i+\delta])}, \tag{3.1}$$

where $\bar{\mu}^{i,\theta} = \bar{\mu}^{\gamma_\theta, \mu_0^i}$ represents the Nash equilibrium trajectory with γ_θ, and $\delta \in (0, T]$ is the sub-trajectory length ($\delta = T$ recovers the full-trajectory loss). In practice, we use the Adam optimizer Kingma & Ba (2014). The full training and parameter calibration procedure is summarized in Algo. 1.

Picard Solver. Consider a fixed initial distribution μ_0 and parameter γ. To find the Nash equilibrium, we iteratively update the control, value function, and population distribution Carlini & Silva (2014): at iteration k, we have the mean field μ^k and we solve the backward equation to compute $u^{(k+1)}$; from this, we compute the new control $\alpha^{(k+1)}$; finally we solve the forward dynamics using this new control and obtain $\mu^{(k+1)}$.

We stress that this depends on γ. We denote a single full cycle of this forward-backward update as an operator Ξ, such that $\mu^{(k+1),\gamma} = \Xi(\mu^{(k),\gamma}, \gamma)$. The mean field is expected to converge towards a fixed point $\bar{\mu}^\gamma$ satisfying $\bar{\mu}^\gamma = \Xi(\bar{\mu}^\gamma, \gamma)$, which corresponds to the Nash equilibrium for parameter γ. The result depends on the initial distribution μ_0 at time 0, and we denote the fixed-point solver that computes this equilibrium as Φ, mapping the parameters and initial condition to the equilibrium trajectory: $\bar{\mu}^{\gamma,\mu_0} = \Phi(\gamma, \mu_0)$. See App. A for details on Picard iteration for MFGs.

Incorporating the Neural Network. We embed the neural network dynamically within the Picard solver to find a coupled fixed point $\bar{\mu}^{\gamma,\mu_0} = \Phi(\varphi_\theta(\cdot, \bar{\mu}^{\gamma,\mu_0}), \mu_0)$ where $\gamma = \varphi_\theta(\cdot, \bar{\mu}^{\gamma,\mu_0})$. Because the neural network is evaluated exclusively on the *observed* distributions rather than the solver's intermediate distributions, the resulting parameter $\gamma^i_\theta(t) := \varphi_\theta(t, \mu^i_t)$ fed into the Picard solver is a function of time only. This approach completely decouples the parameter prediction from the solver's internal iterations, rendering the computational graph significantly more tractable.

Implicit Differentiation. Even with the parameter prediction decoupled, computing the parameter gradients $\nabla_\theta \mathcal{L}(\theta)$ naively requires unrolling the Picard iteration and backpropagating through the entire sequence of solver steps. To avoid the massive memory overhead and gradient instabilities associated with unrolling, we use implicit differentiation through the fixed point Bai et al. (2019); Blondel et al. (2022). Since the converged trajectory $\bar{\mu}^{i,\theta}$ is a fixed point of the iterative step operator Ξ, i.e., $\bar{\mu}^{i,\theta} = \Xi(\bar{\mu}^{i,\theta}, \gamma^i_\theta)$, we can apply the Implicit Function Theorem to the residual $\Psi(\bar{\mu}, \gamma) = \bar{\mu} - \Xi(\bar{\mu}, \gamma) = 0$. This yields the exact Jacobian $\frac{\partial \bar{\mu}}{\partial \gamma} = (I - \frac{\partial \Xi}{\partial \bar{\mu}})^{-1} \frac{\partial \Xi}{\partial \gamma}$. This allows us to compute the vector-Jacobian products required for the backward pass by solving a single adjoint linear system at the equilibrium state, bypassing the need to store or unroll the forward solver iterations (see next subsection for justification). This idea generalizes to other ODE systems (see App. F for an extension to a mean field control example).

Algorithm 1 Finding a data-driven optimal control in a finite-state MFG.

Require: $K \in \mathbb{N}$ epochs; $M \in \mathbb{N}$ minibatch size; data $\mathcal{D} \subseteq C([0,T]; \mathcal{P}([d]))$; Picard solver Φ; learning rate ρ; sub-trajectory length $\delta \in (0, T]$

1: Initialize the neural network φ_θ with parameters $\theta = \theta_1$
2: **for** $k = 1, \dots, K$ **do**
3: Sample a minibatch of M trajectories $\mu^i \in \mathcal{D}$ and M initial times $t_i \in [0, T - \delta]$
4: **for** $i = 1, \dots, M$ **do**
5: $\bar{\mu}^{i,\theta_k} \leftarrow \Phi\left(\varphi_{\theta_k}(t, \mu^i_t), \mu^i_0\right)$ ▷ *Solve MFG over the full time horizon* $[0, T]$
6: $L_i \leftarrow \|\mu^i - \bar{\mu}^{i,\theta_k}\|^2_{L^2([t_i, t_i+\delta])}$ ▷ *Calculate loss on the sub-trajectory*
7: $\mathcal{L}(\theta_k) \leftarrow \frac{1}{M} \sum_{i=1}^M L_i$ ▷ *Compute average loss*
8: $\theta_{k+1} \leftarrow \theta_k - \rho \nabla_\theta \mathcal{L}(\theta_k)$ ▷ *Implicit differentiation update*
9: **return** trained neural network parameters θ_{K+1}

Theoretical Analyses While a theoretical analysis is not the main focus of this work, we investigate the key mechanisms guaranteeing the soundness of our method. In particular, we consider the contraction and differentiability of the Picard operator, establish the correctness of the adjoint gradient computation from Algo. 1, and illustrate the uniform generalization bound of the method. All formal statements, assumptions and proofs are provided in App. B.

4 Test Cases

We consider the following examples to provide numerical evidence for the accuracy and generalization of our approach. We provide a brief overview here, and refer to App. E for the full game descriptions, synthetic data generation and experimental details.

Linear-Quadratic. We consider a linear-quadratic (LQ) MFG that is adapted from Cohen et al. (2024, Section 7.1) and can be regarded as a crowd motion model that is purely based on the individual

player's control. The model has a flexible number of states d, bounded continuous transition-rate controls $a_{xy} \in [1, 3]$ for $y \neq x$, and one learnable parameter ($\gamma \equiv b \in \mathbb{R}$). We instantiate the game using synthetically generated data under different conditions of synthetic noise drawn from a Dirichlet distribution.

Cybersecurity. The cybersecurity model considers a botnet-like population of computers that may become victims of a cyberattack. It is originally introduced by Kolokoltsov & Bensoussan (2016), and has subsequently been revisited by Carmona & Delarue (2018, Section 7.2.3), solved numerically by Laurière (2021, Section 7.2) and solved approximately via the master equation by Cohen et al. (2024, Section 7.4). We focus on the implementation found in Carmona & Delarue (2018). The game has $d = 4$ states, corresponding to protected/unprotected and susceptible/infected computers, a discrete action space $\mathcal{A} = \{0, 1\}$, and eight learnable parameters ($\gamma \in \mathbb{R}^8$). Just like the linear-quadratic example, the game is instantiated using synthetically generated data under different conditions of synthetic noise drawn from a Dirichlet distribution.

Susceptible-Infected-Recovered. The Susceptible-Infected-Recovered (SIR) model is well-studied in the MFG literature Kermack & McKendrick (1927). Related works include, but are not limited to, dynamics related to vaccination Hubert & Turinici (2018); Doncel et al. (2022), social distancing Lee et al. (2021); Aurell et al. (2022b); Buckley & Vladimirsky (2025), both vaccination and social distancing Liu et al. (2025); Liu & Dayanıklı (2026) and optimal interventions via governing institutions Aurell et al. (2022a). In this section, we consider a simplified variant of the Stackelberg MFG proposed by Aurell et al. (2022a), allowing a population of homogeneous players to control their own interaction rate. The game has $d = 3$ states ($[d] = \{S, I, R\}$), a continuous action space $\mathcal{A} = [0, 1]$ and four learnable parameters ($\gamma \in \mathbb{R}^4$). We analyze a publicly available influenza monitoring dataset from the *National Respiratory and Enteric Virus Surveillance System (NREVSS)* U.S. Centers for Disease Control and Prevention (2026), which describes the weekly positive rates of influenza tests per U.S. state as reported to the *U.S. Centers for Disease Control and Prevention* (CDC). We filter the data by U.S. state and year, and use the percentage of positive Influenza A tests as an observation proxy for the infected coordinate $\mu(I)$, rather than as a fully observed population distribution. We capture the yearly epidemic peaks, where each state and year constitutes one observed trajectory ($N = 30$ weeks). We approximately fit our 30-week window such that each flu season starts around $t = 0$. The filtering is approximate on purpose to generate different initial states μ_0. We drop all trajectories that contain more than 15 missing values, yielding 279 observed trajectories that we chronologically split into training and testing sets.

Urban Mobility. As a last example, we consider a bike-sharing model where we track the cumulative arrivals and departures at six stations. The example allows for the infusion of strategic behavior in the players' commute by penalizing both the deviation from natural bike demand patterns and arriving at congested stations. Therefore, the game has $d = 6$ states, continuous nonnegative outgoing transition-rate controls, and learnable base-rate and cost parameters. We use the publicly available *Citi Bike trip-level* dataset from New York City Citi Bike (2026). Each record contains a trip's start time, end time, origin station, and destination station. We select a cluster of $K = 5$ stations in Midtown Manhattan and introduce a sixth "external" state representing the rest of the network. By tracking cumulative arrivals and departures at each station in 15-minute bins over the interval $[06{:}00, 22{:}00]$, we reconstruct the empirical mean field distribution $\hat{\mu}(t) \in \mathcal{P}([d])$. Each weekday constitutes one observed trajectory ($N = 64$ time steps), providing approximately 20 trajectories per month that we split into training and testing sets.

5 Experiments & Results

We evaluate the objectives that we introduced in Sec. 2 through numerical experiments based on the examples from Sec. 4. The synthetic experiments are closest to the i.i.d. setting used in the theoretical generalization result, whereas the influenza and bike-sharing experiments use real data with chronological structure and should be interpreted as empirical evaluations beyond that formal guarantee. All experiments have been run $n = 5$ times with different random seeds. We report the mean $\pm$ standard deviation over all runs.

Objective 1: MFG Calibration. We assess the calibration quality by evaluating the predicted mean field flow μ_t^θ and L_2 performance on the LQ and Cybersecurity MFGs with synthetic data. The true parameters $\gamma_i \in [\gamma_{i,\min}, \gamma_{i,\max}]$ take three functional forms: constant, time-dependent and mean field-dependent (see App. E). Fig. 2 shows the predictions of the calibrated MFG on two data

points from the test set, together with the L_2 error on the predicted mean field flow (train and test) and parameters for the LQ example. From the figure, it follows that the predicted mean field flows closely match the observations as the training converges and the L_2 loss on the parameter estimates is minimized. Results for the cybersecurity example are in App. G.

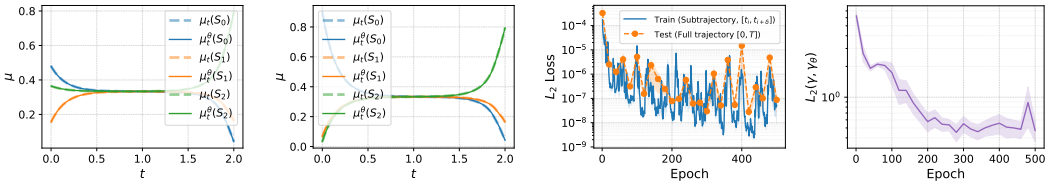


Figure 2: LQ MFG, from left to right: Predicted mean field flow on two random samples from the test set, L_2 loss on mean fields (train and test), and L_2 error on predicted parameters.

Objective 2: Comparing MFGs and Forward Models. To isolate and demonstrate the forward-looking power of the MFG formulation, we compare our model to a mean field dynamics baseline that describes the population flow via exogenous transition rates. The MFG model contains the forward dynamics baseline as a special case when the strategic costs and terminal cost vanish and the HJB solution is selected as $u \equiv 0$, in which case $a^*_{xy} = q^{\text{base}}_{xy}$. We evaluate the two models under the following three testing regimes on the urban mobility example. More elaborate experimental descriptions and results are provided in App. H.

Mean field dynamics model (baseline). In this model, agents follow exogenous, time-dependent transition rates without any strategic behavior. The population distribution evolves according to the Kolmogorov forward equation: $\dot{\mu}_i = \sum_{j \neq i} \mu_j \, q_{ji}(\gamma) - \mu_i \sum_{j \neq i} q_{ij}(\gamma)$,where the rates $q_{ij}(t) = \text{softplus}(\gamma_{ij}(t))$ are parameterized by a neural network $\varphi_\theta : [0, T] \to \mathbb{R}^{d(d-1)}$. Depending on the test regime (see below), φ_θ is either a constant vector, a single affine layer, or a multi-layer perceptron. The network is trained by minimizing the L_2 discrepancy between the predicted and observed trajectories, analogously to the procedure described in Sec. 3.

Mean field game model. In the MFG formulation, each commuter optimizes their destination choice to minimize a cost that penalizes both deviating from the natural demand pattern and arriving at congested stations. The running cost is:

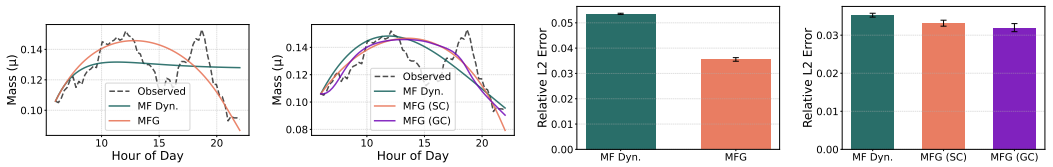
$$f(x, a, \mu) = b \sum_{y \neq x} (a_{xy} - q^{\text{base}}_{xy})^2 + c \sum_{y \neq x} \mu(y) \, a_{xy}, \tag{5.1}$$

where $q^{\text{base}}_{xy}(t) = \text{softplus}(\gamma_{xy}(t))$ are the same base rates as in the baseline model, $b > 0$ controls the cost of deviating from the natural demand, and $c \geq 0$ weights the destination congestion penalty. The terminal cost is $g(x) = w_g(x)$, where $w_g \in \mathbb{R}^d$ is a learnable vector initialized to zero. We consider three test cases of increasing complexity and compare all model predictions with the observed mean field mass:

- **Test 1: Constant Base Rates.** We restrict the base transition rates to be entirely constant over time: $\varphi_\theta(t) = \gamma_0$ (a learned but time-independent vector). A standard ODE with constant rates decays toward a stationary distribution and cannot recreate the complex, double-peaked morning and evening commute patterns. The MFG, however, learns a non-trivial terminal cost $w_g(x)$ representing commuters' end-of-day location preferences (Fig. 3a).
- **Test 2: Linear Base Rates.** We allow the base rates to be affine functions of time: $\varphi_\theta(t) = Wt + b_{\text{bias}}$, where $W \in \mathbb{R}^{d(d-1) \times 1}$ and $b_{\text{bias}} \in \mathbb{R}^{d(d-1)}$. This enables the baseline to capture monotonic trends over the day but prevents it from fitting highly non-linear dynamics (Fig. 3b).
- **Test 3: Restricted Dynamics with General Cost.** We consider a practitioner who relies on a rigid phenomenological model for the baseline dynamics (linear base rates, as in Test 2) and does not wish to alter it. Instead, we augment the MFG with a flexible cost network $\psi_\phi : [0, T] \times \mathcal{P}([d]) \to \mathbb{R}^{d(d-1)}$ (a 2-layer MLP with 64 hidden units and $\tanh$ activations), which replaces the scalar congestion cost $c \cdot \mu(y)$ in equation 5.1 with a general, potentially negative, cost vector $\gamma_{\text{cost}}(t, \mu_t) := \psi_\phi(t, \mu_t) \in \mathbb{R}^{d(d-1)}$. The running cost becomes: $f(x, a, \mu, t) = b \sum_{y \neq x} (a_{xy} - q^{\text{base}}_{xy})^2 - \sum_{y \neq x} [\gamma_{\text{cost}}]_{xy}(t, \mu) \, a_{xy}$,and the corresponding optimal control is: $a^*_{xy} = \max\left(0, \; q^{\text{base}}_{xy} + \frac{[\gamma_{\text{cost}}]_{xy} - (u(y) - u(x))}{2b}\right)$.

The sign of the linear cost term is reversed compared to equation 5.1, and the network ψ_ϕ can output negative values, enabling the model to learn both congestion penalties and incentives (Fig. 3b).

We show the predicted mean field mass for a single station (Station 0) for the three test cases in Fig. 3. Although neither model perfectly matches the observations (because it comes from real data and



(a) (b) (c) (d)

Figure 3: Predicted mean field mass for Station 0 under three test cases: (a) constant base rates, (b) linear base rates with scalar costs (SC) and linear base rates with general costs (GC). Results are shown for both the mean field dynamics baseline (MF Dyn.) and the MFG throughout. Relative errors with respect to the observed mean field mass compare the models with (c) constant base rates and (d) linear base rates with scalar and general costs.

not synthetic MFG-based data), the MFG benefits from the inclusion of the terminal cost g_w that is propagated backward in time by the HJB. This causes agents to strategically adjust their flows throughout the day and outperform the MF baseline on the in-distribution weekday test shown in this experiment (Fig. 3c, d).

Objective 3: Noise Robustness, Out-of-Distribution Forecasts and Novel Scenarios. Fig. 4 presents an analysis of robustness to noise in the observed trajectories μ_t for the LQ model. As the noise level increases, the mean-field L_2 error worsens, while the parameter estimates remain qualitatively close to the ground truth in the displayed time- and mean-field-dependent examples.

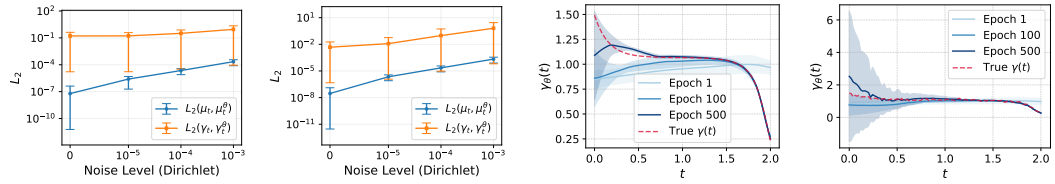


Figure 4: Noise robustness for the parameter estimates γ_θ and the mean field predictions μ_θ for the LQ model. From left to right: time-dependent and mean field-dependent, predictions for the mean-field dependent γ_t at $\sigma = 0$ and $\sigma = 10^{-3}$.

To further demonstrate the noise robustness of the mean field flow predictions, we assess the calibration accuracy in a real-world system by modelling Influenza activity in the United States. We train the calibration method on historical weekly observations and evaluate it on held-out flu seasons. The resulting out-of-distribution prediction of the SIR dynamics is associated with an error of $L_2(\mu_t(I), \mu_t^\theta(I)) = 1.45 \times 10^{-1} \pm 1.11 \times 10^{-1}$, and representative predictions for 2023–2025 are shown in Fig. 5. Because test positivity is not the same object as disease prevalence and only the infected coordinate is observed, this experiment should be interpreted as calibration to an influenza activity proxy. The susceptible and recovered trajectories shown in Fig. 5 are model-implied latent coordinates, not independently observed CDC quantities. Even though the predicted curves do not perfectly match the observed fraction of positive tests, the model captures important characteristics such as the magnitude and duration of the infection peak.

Lastly, we evaluate our model on its zero-shot adaptability to novel scenarios. In doing so, we highlight a fundamental advantage of MFGs: in contrast to a forward baseline model, game-theoretic models have the ability to evaluate counterfactual scenarios without retraining. We illustrate this by revisiting the baseline comparison in the urban mobility MFG, while simulating an announced station closure: the city announces that a given station will be closed for maintenance for a fixed period of time $t \in [t_s, t_e]$ and wishes to predict the behavior of users.

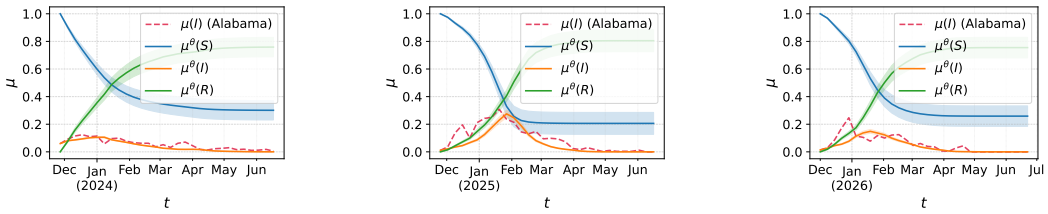


Figure 5: Evolution of the mean field distribution trajectories μ_t^θ for the U.S. state Alabama in for representative held-out seasons from 2023 to 2025. The predictions are compared to the observed proxy $\mu_t(I)$. The predicted mean field flow describing the number of susceptible and recovered agents is derived from the SIR definition given $\mu_t^\theta(I)$.

Since the baseline model has no value function or cost structure, the only way to encode the closure is to mechanically set the transition rates into Station 1 to zero during the closure window. Crucially, the rates remain unchanged before the closure at t_s, as the model has no mechanism to anticipate the disruption. In contrast, the closure can be encoded in the MFG model as a state-dependent running cost penalty. Because the HJB equation propagates the new cost function from terminal time T to time 0, the value function $u(t, x)$ at times $t < t_s$ already reflects the anticipated penalty.

Fig. 6 compares the two calibrated models in this counterfactual closure scenario. The forward dynamics baseline follows its learned historical pattern until the closure begins, at which point the imposed closure constraint changes the trajectory. The MFG model, however, gradually drains the affected station before the closure starts, as the closure penalty is propagated backward through the HJB equation. This simulation illustrates how the game-theoretic model can produce anticipatory responses to announced interventions without retraining.

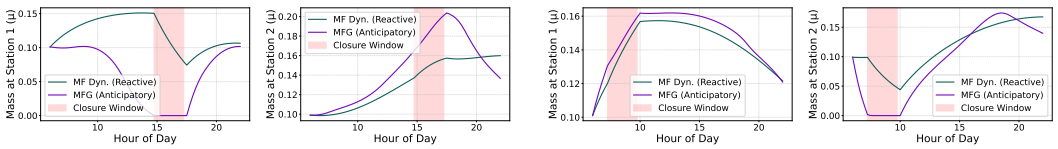


Figure 6: Counterfactual station-closure simulation, from left to right: station 1 (scenario 1), station 2 (scenario 1), station 1 (scenario 2), station 2 (scenario 2). The reactive MF dynamics baseline follows its historical routing pattern unchanged until the closure instant, then flatlines. The anticipatory MFG model propagates the penalty backward through the HJB equation, causing agents to reroute away from the station *before* the closure begins. We refer to App. I for all other trajectories.

6 Conclusion

We introduced a novel, fully differentiable framework for learning trajectory-wise parameter paths induced by time- and mean-field-dependent neural parameterizations of finite-state mean field games directly from macroscopic population trajectories, supported by exact adjoint gradient computations and uniform generalization bounds under discrete-time contraction and smoothness assumptions. We validated the structural advantages of our approach across four domains, illustrating anticipatory responses in counterfactual station-closure simulations.

Limitations. The current methodology is limited by its reliance on *a priori* knowledge of the game's structural form and can struggle to generalize in out-of-distribution scenarios (such as the transition from weekday commutes to weekend leisure riding) when the underlying user utilities fundamentally change. Future work will focus on extending this framework to continuous state-action spaces, relaxing the i.i.d. and smoothness assumptions used in the theoretical analysis, relaxing the requirement for fully observable population densities, and integrating structural discovery mechanisms to improve robustness against unobserved distributional shifts.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of mathematical proofs and implementation of the methods. We have not used generative AI tools for developing theoretical models or conceptual frameworks; formulating mathematical claims; providing critical ingredients for proving mathematical claims; proposing or refining hypotheses; designing or providing feedback on the research methodology or experiments; and the interpretation of results. The rest of the required disclosure tasks, including the generation of synthetic datasets, the assistance with translation, dataset cleaning and reformatting and the support of qualitative and thematic data analysis, are not applicable to this work. Additionally, we used generative AI tools for creating, editing and checking the Python implementation of our methods, and for checking the spelling and grammar of the manuscript. We have reviewed all AI-assisted work. In particular, for the AI-assistance in the writing of the mathematical proofs, we only let the generative AI tool generate a coarse-grained proof sketch, which we further developed ourselves. As a result, all proofs in this work have been human-written and verified by 3 authors. Regarding the implementation of our methods, we used AI tools to assist in drafting and editing Python code. All code has been reviewed and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This research is entirely theoretical and computational, using publicly available data that does not involve any identifiable human participants. Consequently, the authors do not feel that their paper submission raises questions regarding the Code of Ethics.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of the work presented in the paper, we provide a source code as supplementary material to this paper. The code has been run with the 5 fixed seeds. In addition, we provide an explanation of the assumptions and proofs of the theoretical claims made in the paper in App. B. Lastly, we provide all experimental details in App. E and J.

REFERENCES

Şevket Kaan Alkır and Naci Saldi. Inverse Reinforcement Learning for Mean-field Games with Average Reward Criterion. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pp. 7272–7277. IEEE, 2025. doi:10.1109/CDC57313.2025.11312818.

Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi. Learning in Discrete-time Average-cost Mean-field Games. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 3048–3053. IEEE, 2021. doi:10.1109/CDC45484.2021.9682954.

Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi. Maximum Causal Entropy Inverse Reinforcement Learning for Mean-Field Games. *arXiv preprint*, 2024. doi:10.48550/arXiv.2401.06566.

Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi. Kernel Based Maximum Entropy Inverse Reinforcement Learning for Mean-Field Games. *arXiv preprint*, 2025a. doi:10.48550/arXiv.2507.14529.

Berkay Anahtarci, Can Deha Kariksiz, and Naci Saldi. Maximum Causal Entropy IRL in Mean-Field Games and GNEP Framework for Forward RL. *Journal of Machine Learning Research*, 26(121): 1–40, 2025b. doi:10.48550/arXiv.2401.06566.

Alexander Aurell, René Carmona, Gökçe Dayanıklı, and Mathieu Laurière. Optimal Incentives to Mitigate Epidemics: A Stackelberg Mean Field Game Approach. *SIAM Journal on Control and Optimization*, 60(2):S294–S322, 2022a. doi:10.1137/20M1377862.

Alexander Aurell, René. Carmona, Gökçe Dayanıklı, and Mathieu Laurière. Finite state graphon games with applications to epidemics. *Dyn Games Appl*, 12(1):49–81, 2022b. ISSN 2153-0785 (Print) 2153-0785. doi:10.1007/s13235-021-00410-2.

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep Equilibrium Models. In *Advances in Neural Information Processing Systems*, volume 32, 2019. doi:10.48550/arXiv.1909.01377.

Alain Bensoussan, Jens Frehse, and Phillip Yam. *Mean Field Games and Mean Field Type Control Theory*. Springer Briefs in Mathematics, 2013. doi:10.1007/978-1-4614-8508-7.

Alain Bensoussan, K.C. Joseph Sung, Sheung Chi Phillip Yam, and Siu-Pang Yung. Linear-Quadratic Mean Field Games. *Journal of Optimization Theory and Applications*, 169(2):496–529, 2016. doi:10.1007/s10957-015-0819-4.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and Modular Implicit Differentiation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi:10.48550/arXiv.2105.15183.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Finnegan Buckley and Alexander Vladimirsky. Behavioral patterns and mean-field games in epidemiological models. *arXiv preprint*, 2025. doi:10.48550/arXiv.2512.20547.

Peter E. Caines, Minyi Huang, and Roland P. Malhamé. *Mean Field Games*, pp. 1–28. Springer International Publishing, Cham, 2017. ISBN 978-3-319-27335-8. doi:10.1007/978-3-319-27335-8_7-1.

Pierre Cardaliaguet and Charles-Albert Lehalle. Mean field game of controls and an application to trade crowding. *Mathematics and Financial Economics*, 12(3):335–363, 2018. ISSN 1862-9660. doi:10.1007/s11579-017-0206-z.

Elisabetta Carlini and Francisco J. Silva. A Fully Discrete Semi-Lagrangian Scheme for a First Order Mean Field Game Problem. *SIAM Journal on Numerical Analysis*, 52(1):45–67, 2014. doi:10.1137/120902987.

Rene Carmona. Applications of Mean Field Games in Financial Engineering and Economic Theory. *arXiv preprint*, 2020. doi:10.48550/arXiv.2012.05237.

René Carmona and François Delarue. *Probabilistic theory of mean field games with applications I*. Probability Theory and Stochastic Modelling. Springer Cham, 2018. ISBN 978-3-319-58920-6. doi:10.1007/978-3-319-58920-6.

René Carmona and Mathieu Laurière. *Deep Learning for Mean Field Games and Mean Field Control with Applications to Finance*, pp. 369–392. Cambridge University Press, 2023. doi:10.1017/9781009028943.021.

Alekos Cecchin and Markus Fischer. Probabilistic approach to finite state mean field games. *Applied Mathematics & Optimization*, 81:253–300, 2020. doi:10.1007/s00245-018-9488-7.

Yang Chen, Libo Zhang, Jiamou Liu, and Michael Witbrock. Adversarial Inverse Reinforcement Learning for Mean Field Games. *arXiv preprint*, 2021. doi:10.48550/arXiv.2104.14654.

Yang Chen, Libo Zhang, Jiamou Liu, and Shuyue Hu. Individual-Level Inverse Reinforcement Learning for Mean Field Games. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 253–262, 2022. doi:10.48550/arXiv.2202.06401.

Geoffroy Chevalier, Jerome Le Ny, and Roland Malhamé. *A micro-macro traffic model based on Mean-Field Games*, pp. 1983–1988. IEEE, 2015. doi:10.1109/ACC.2015.7171024.

Citi Bike. Citi Bike Trip Histories. https://citibikenyc.com/system-data, 2026. Licensed under the NYCBS Data Use Policy. Data retrieved on May 4, 2026.

Asaf Cohen, Mathieu Laurière, and Ethan Zell. Deep backward and Galerkin methods for the finite state master equation. *J. Mach. Learn. Res.*, 25(1), January 2024. ISSN 1532-4435. doi:10.48550/arXiv.2403.04975.

Josu Doncel, Nicolas Gast, and Bruno Gaujal. A Mean Field Game Analysis of SIR Dynamics with Vaccination. *Probability in the Engineering and Informational Sciences*, 36(2):482–499, 2022. doi:10.1017/S0269964820000522.

Adriano Festa and Simone Göttlich. A mean field game approach for multi-lane traffic management. *IFAC-PapersOnLine*, 51(32):793–798, 2018. doi:10.48550/arXiv.1711.04116.

Thomas Gaskin, Grigorios A. Pavliotis, and Mark Girolami. Neural parameter calibration for large-scale multiagent models. *Proceedings of the National Academy of Sciences*, 120(7):e2216415120, 2023. doi:10.1073/pnas.2216415120.

Stefano Giampiccolo, Federico Reali, Anna Fochesato, Giovanni Iacca, and Luca Marchetti. Robust parameter estimation and identifiability analysis with hybrid neural ordinary differential equations in computational biology. *npj Systems Biology and Applications*, 10(1):139, 2024. ISSN 2056-7189. doi:10.1038/s41540-024-00460-3.

Diogo A. Gomes, Joana Mohr, and Rafael Rigao Souza. Continuous Time Finite State Mean Field Games. *Applied Mathematics & Optimization*, 68(1):99–143, 2013. doi:10.1007/s00245-013-9202-8.

Masanori Hirano and Kiyoshi Izumi. Efficient Parameter Tuning for Multi-agent Simulation Using Deep Reinforcement Learning. In *2022 13th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter)*, pp. 130–137, 2022. doi:10.1109/IIAI-AAI-Winter58034.2022.00035.

William Hofgard, Asaf Cohen, and Mathieu Laurière. Operator Learning for Families of Finite-State Mean-Field Games. *arXiv preprint*, 2026. doi:10.48550/arXiv.2602.13169.

Jiawei Huang, Niao He, and Andreas Krause. Model-Based RL for Mean-Field Games is not Statistically Harder than Single-Agent RL. In *Forty-first International Conference on Machine Learning*, 2024. doi:10.48550/arXiv.2402.05724.

Kuang Huang, Xuan Di, Qiang Du, and Xi Chen. Stabilizing Traffic via Autonomous Vehicles: A Continuum Mean Field Game Approach. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3269–3274, 2019. doi:10.1109/ITSC.2019.8917021.

Minyi Huang, Roland P. Malhamé, and Peter E. Caines. 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. doi:10.4310/CIS.2006.v6.n3.a5.

Emma Hubert and Gabriel Turinici. Nash-MFG equilibrium in a SIR model with time dependent newborn vaccination. *Ricerche di matematica*, 67(1):227–246, 2018. doi:10.1007/s11587-018-0365-0.

William Ogilvy Kermack and Anderson G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772):700–721, 1927. doi:10.1098/rspa.1927.0118.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014. doi:10.48550/arXiv.1412.6980.

Vassili N. Kolokoltsov. *Nonlinear Markov Processes and Kinetic Equations*. Cambridge Tracts in Mathematics. Cambridge University Press, 2010. doi:10.1017/CBO9780511760303.

Vassili N. Kolokoltsov and Alain Bensoussan. Mean-Field-Game Model for Botnet Defense in Cyber-Security. *Applied Mathematics & Optimization*, 74(3):669–692, 2016. ISSN 1432-0606. doi:10.1007/s00245-016-9389-6.

Jean-Michel Lasry and Pierre-Louis Lions. Mean Field Games. *Japanese Journal of Mathematics*, 2: 229–260, 03 2007. doi:10.1007/s11537-007-0657-8.

Mathieu Laurière. Numerical Methods for Mean Field Games and Mean Field Type Control. In *Proc. Sympos. Appl. Math.*, volume 78, pp. 221–282. Amer. Math. Soc., Providence, RI, 2021. doi:10.48550/arXiv.2106.06231.

Wonjun Lee, Siting Liu, Hamidou Tembine, Wuchen Li, and Stanley Osher. Controlling Propagation of Epidemics via Mean-Field Control. *SIAM Journal on Applied Mathematics*, 81(1):190–207, 2021. doi:10.1137/20M1342690.

Huaning Liu and Gökçe Dayanıklı. Modeling Epidemic Spread with Strategic Vaccination and Socialization: a Mean Field Game Analysis. *arXiv preprint*, 2026. doi:10.48550/arXiv.2604.22946.

Huaning Liu, Junke Yang, Soren L. Larsen, Pamela P. Martinez, and Gökçe Dayanıklı. Incorporating Authority Perception, Economic Status, and Behavioral Response in Infectious Disease Control. *arXiv preprint*, 2025. doi:10.48550/arXiv.2512.23188.

Sarah Perrin, Julien Perolat, Mathieu Lauriere, Matthieu Geist, Romuald Elie, and Olivier Pietquin. Fictitious Play for Mean Field Games: Continuous Time Analysis and Applications. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13199–13213. Curran Associates, Inc., 2020. doi:10.48550/arXiv.2007.03458.

Chengshan Qian, Xue Li, Ning Sun, and Yuqing Tian. Data Security Defense and Algorithm for Edge Computing Based on Mean Field Game. *Journal of Cybersecurity*, 2(2):97, 2020. doi:10.32604/jcs.2020.010548.

Shigen Shen, Chenpeng Cai, Yizhou Shen, Xiaoping Wu, Wenlong Ke, and Shui Yu. Joint mean-field game and multiagent asynchronous advantage actor-critic for edge intelligence-based iot malware propagation defense. *IEEE Transactions on Dependable and Secure Computing*, 22(4):3824–3838, 2025. doi:10.1109/TDSC.2025.3542104.

Takashi Tanaka, Ehsan Nekouei, Ali Reza Pedram, and Karl Henrik Johansson. Linearly Solvable Mean-Field Traffic Routing Games. *IEEE Transactions on Automatic Control*, 66(2):880–887, 2021. doi:10.1109/TAC.2020.2986195.

Anna C. M. Thöni, Yoram Bachrach, and Tal Kachman. Neural Mean-Field Games: Extending Mean-Field Game Theory with Neural Stochastic Differential Equations. *arXiv preprint*, 2026. doi:10.48550/arXiv.2504.13228.

U.S. Centers for Disease Control and Prevention. Percent positivity of Respiratory Viruses as reported on the National Respiratory and Enteric Virus Surveillance System (NREVSS) Dashboard. https://www.cdc.gov/nrevss/php/dashboard/index.html/, 2026. The data falls under the Public Domain (U.S. Government). Data retrieved on May 4, 2026.

Shimin Wang and Xiangyu Meng. Adaptive Consensus and Parameter Estimation of Multiagent Systems With an Uncertain Leader. *IEEE Transactions on Automatic Control*, 66(9):4393–4400, 2021. doi:10.1109/TAC.2020.3046215.

Batuhan Yardim, Semih Cayci, Matthieu Geist, and Niao He. Policy Mirror Ascent for Efficient and Independent Learning in Mean Field Games. In *International Conference on Machine Learning*, pp. 39722–39754. PMLR, 2023. doi:10.48550/arXiv.2212.14449.

Bora Yongacoglu, Gürdal Arslan, and Serdar Yüksel. Independent learning and subjectivity in mean-field games. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 2845–2850. IEEE, 2022. doi:10.1109/CDC51059.2022.9992399.

A Picard Iteration for Solving MFGs

Picard iteration is a fixed-point numerical method for solving the forward-backward system associated with MFGs. In this section, we describe the implementation of the solver called in line 5 of Algo. 1. For a parameter path $\gamma = (\gamma_t)_{t\in[0,T]}$ and initial distribution μ_0, we denote by $\bar{\mu}^{\gamma,\mu_0} = \Phi(\gamma, \mu_0) \in \mathcal{P}([d])^{N+1}$ the discrete equilibrium trajectory returned by the solver; the associated value function and control are computed internally.

We aim to solve the system of PDEs described in equation 2.4 on the time interval $[0, T]$.

Let $t_i = i\Delta t$, $i = 0, \dots, N$, with $\Delta t = T/N$. Given a Picard iterate $\mu^{(k)} = (\mu_i^{(k)})_{i=0}^N$, the backward step sets

$$u_N^{(k+1)}(x) = g^\gamma(x, \mu_N^{(k)}),$$

and, for $i = N-1, \dots, 0$,

$$u_i^{(k+1)}(x) = u_{i+1}^{(k+1)}(x) + \Delta t\, H^\gamma\left(t_i, x, \mu_i^{(k)}, u_{i+1}^{(k+1)}(\cdot), \alpha_i^{H,k+1}(x)\right),$$

where

$$\alpha_i^{H,k+1}(x) \in \arg\min_{a\in\mathcal{A}} H^\gamma\left(t_i, x, \mu_i^{(k)}, u_{i+1}^{(k+1)}(\cdot), a\right).$$

The forward step then sets $\tilde{\mu}_0^{(k+1)} = \mu_0$ and, for $i = 0, \dots, N-1$,

$$\tilde{\mu}_{i+1}^{(k+1)}(x) = \tilde{\mu}_i^{(k+1)}(x) + \Delta t \sum_{y\in[d]} \tilde{\mu}_i^{(k+1)}(y)\, Q^\gamma\left(t_i, \alpha_i^{F,k+1}(y), \tilde{\mu}_i^{(k+1)}\right)[y, x],$$

with

$$\alpha_i^{F,k+1}(x) \in \arg\min_{a\in\mathcal{A}} H^\gamma\left(t_i, x, \tilde{\mu}_i^{(k+1)}, u_i^{(k+1)}(\cdot), a\right).$$

Using fixed-point iteration, we produce approximations of the value function $\{u_i^{\gamma,\mu_0}(x)\}_{i=0}^N$ and mean field flow $\{\mu_i^{\gamma,\mu_0}(x)\}_{i=0}^N$ for $x \in [d]$ along the discrete time grid. To improve stability across a broad range of interaction strengths, we use a damping coefficient $\lambda \in [0, 1)$ and set $\lambda = 0.5$ in our experiments, blending each forward FPK update with the previous distribution iterate. The full fixed-point algorithm is summarized in Alg. 2.

For ease of notation, we omit the dependence on γ and μ_0 in Algorithm 2. The algorithm approximates the solution of a single MFG $\mathcal{M}^\gamma$ for a fixed initial law μ_0; in the calibration procedure, it is called separately for each sampled trajectory.

B Theoretical Analyses

We provide theoretical results supporting the algorithmic choices made in Sec. 3.1. The existence and uniqueness of Nash equilibria in finite-state MFGs have been established under various structural conditions, including monotonicity and smallness of the time horizon; see e.g. Gomes et al. (2013) and Cecchin & Fischer (2020). We take these as a starting point and work throughout in the discrete-time setting used by our numerical implementation.

Algorithm 2 Picard iteration for approximating the Nash equilibrium of finite-state MFGs

Require: Parameter path γ, initial law μ_0, initial trajectory $\mu^{(0)}$, damping $\lambda \in [0, 1)$, tolerance ε, maximum iterations K

$k \leftarrow 0$, $\delta \leftarrow \infty$
while $\delta > \varepsilon$ and $k < K$ **do**
  $u^{(k+1)} \leftarrow \mathrm{BackwardHJB}(\mu^{(k)}, \gamma)$
  $\tilde{\mu}^{(k+1)} \leftarrow \mathrm{ForwardFPK}(u^{(k+1)}, \gamma, \mu_0)$
  $\mu^{(k+1)} \leftarrow (1 - \lambda)\tilde{\mu}^{(k+1)} + \lambda\mu^{(k)}$
  $\delta \leftarrow \|\mu^{(k+1)} - \mu^{(k)}\|_\infty$
  $k \leftarrow k + 1$
return $\mu^{(k)}$

Discrete-time setup. We discretize the time horizon $[0, T]$ into N steps of size $\Delta t = T/N$, with time points $t_k = k\,\Delta t$ for $k = 0, \dots, N$. A discrete trajectory is a sequence $\mu = (\mu_0, \mu_1, \dots, \mu_N)$ with $\mu_k \in \mathcal{P}([d])$ for each k. We identify μ with an element of $\mathbb{R}^{(N+1)\times d}$ and equip this space with the norm $\|\mu\|_\infty := \max_{0\le k\le N} \max_{x\in[d]} |\mu_k(x)|$. For discrete trajectories, we use

$$\|\mu\|_{2,N}^2 := \Delta t \sum_{k=0}^{N} \sum_{x\in[d]} |\mu_k(x)|^2.$$

Throughout this appendix, Γ denotes the finite-dimensional set of discretized parameter paths after vectorization, e.g. $\gamma = (\gamma_0, \dots, \gamma_N) \in \mathbb{R}^{(N+1)n}$. Whenever derivatives are used, we assume that Ξ admits a C^1 extension to an open neighborhood of the relevant subset of $\mathbb{R}^{(N+1)d} \times \mathbb{R}^{(N+1)n}$; equivalently, derivatives on the simplex may be understood in local or tangent coordinates.

The discrete Picard operator $\Xi : \mathcal{P}([d])^{N+1} \times \Gamma \to \mathcal{P}([d])^{N+1}$ performs one full forward-backward cycle: given a trajectory μ and parameters γ, it (1) solves the backward HJB equation on the discrete grid to obtain the value function, (2) extracts the optimal control at each time step, and (3) propagates forward the FPK equation to produce a new trajectory $\Xi(\mu, \gamma)$. We denote by $\Phi(\gamma, \mu_0)$ the fixed-point solver that returns the equilibrium trajectory $\bar{\mu}^\gamma$ satisfying $\bar{\mu}^\gamma = \Xi(\bar{\mu}^\gamma, \gamma)$.

Assumption B.1 (Picard Contractivity). *For each $\gamma \in \Gamma$, the Picard operator $\Xi(\cdot, \gamma) : \mathcal{P}([d])^{N+1} \to \mathcal{P}([d])^{N+1}$ is a contraction in the $\|\cdot\|_\infty$ norm with constant $\kappa \in [0, 1)$:*

$$\|\Xi(\mu, \gamma) - \Xi(\mu', \gamma)\|_\infty \le \kappa \|\mu - \mu'\|_\infty, \qquad \forall\, \mu, \mu' \in \mathcal{P}([d])^{N+1}.$$

Remark B.1 (On damping). *Assumption B.1 is a standard smallness condition, typically satisfied when the time horizon T, the Lipschitz constants of f and Q in μ, or the number of states d are sufficiently small (Gomes et al., 2013). In practice, one often replaces Ξ by a damped operator $\Xi_\alpha(\mu, \gamma) := (1 - \alpha)\,\Xi(\mu, \gamma) + \alpha\,\mu$ for some $\alpha \in (0, 1)$, which improves numerical stability.*

In our experiments, damping is used as an empirical stabilization device and, when using large enough damping coefficient, the damped Picard iteration converges within most of the tested configurations. We provide an empirical evaluation of the diagnostics of the Picard solver for the LQ game in App. D, showing that the solver converges within the tested regime. If Assumption B.1 fails, fictitious play offers an alternative route to computing the equilibrium: existence, uniqueness, and algorithmic convergence hold under the weaker Lasry–Lions monotonicity condition rather than contraction (Perrin et al., 2020).

Under Assumption B.1, the Banach fixed-point theorem guarantees that for each $\gamma \in \Gamma$, the iteration $\mu^{(k+1)} = \Xi(\mu^{(k)}, \gamma)$ converges to a unique fixed point $\bar{\mu}^\gamma = \Xi(\bar{\mu}^\gamma, \gamma)$ at geometric rate κ^k.

Differentiability of the equilibrium map. The equilibrium trajectory $\bar{\mu}^\gamma$ is an implicit function of γ defined by the fixed-point equation $\bar{\mu}^\gamma - \Xi(\bar{\mu}^\gamma, \gamma) = 0$. In the discrete-time setting, Ξ is a composition of standard smooth operations (e.g., softplus, matrix-vector products). We formalize the required regularity as follows.

Assumption B.2 (Smoothness). *The Picard map Ξ admits a continuously differentiable extension, in both variables, to an open neighborhood of the trajectories and parameter paths considered in $\mathbb{R}^{(N+1)d} \times \mathbb{R}^{(N+1)n}$.*

In the numerical examples, some implementations use clipping, projection, or discrete optimization steps to enforce admissibility and numerical stability. These operations may introduce nonsmooth points, but the numerical results show that the algorithm still works in these situations. The above smoothness assumption should be understood as a sufficient condition for the theoretical analysis.

Under Assumptions B.1 and B.2, we denote the Jacobian matrices of Ξ at the fixed point $(\bar{\mu}^\gamma, \gamma)$ by

$$\mathcal{J}_\mu := \left.\frac{\partial \Xi}{\partial \mu}\right|_{(\bar{\mu}^\gamma,\gamma)} \in \mathbb{R}^{(N+1)d\times(N+1)d}, \qquad \mathcal{J}_\gamma := \left.\frac{\partial \Xi}{\partial \gamma}\right|_{(\bar{\mu}^\gamma,\gamma)},$$

where we identify $\mu \in \mathbb{R}^{(N+1)\times d}$ with its vectorization in $\mathbb{R}^{(N+1)d}$. Since $\Xi(\cdot,\gamma)$ is κ-Lipschitz (Assumption B.1), its Jacobian satisfies $\|\mathcal{J}_\mu\|_\infty \le \kappa < 1$, where $\|\cdot\|_\infty$ denotes the operator norm induced by the ℓ^∞ vector norm. In particular, the spectral radius satisfies $\rho(\mathcal{J}_\mu) \le \|\mathcal{J}_\mu\|_\infty < 1$, so the matrix $(I - \mathcal{J}_\mu)$ is invertible. The Implicit Function Theorem then yields:

$$\frac{\partial \bar{\mu}^\gamma}{\partial \gamma} = (I - \mathcal{J}_\mu)^{-1}\,\mathcal{J}_\gamma. \tag{B.1}$$

Efficient gradient computation via the adjoint method. While equation B.1 provides an explicit formula for the sensitivity of the equilibrium to the parameters, computing it directly would require forming and inverting the $(N+1)d \times (N+1)d$ matrix $(I-\mathcal{J}_\mu)$, which is prohibitively expensive. In Algo. 1, what is actually needed is not the full Jacobian $\partial\bar{\mu}^\gamma/\partial\gamma$ itself, but rather the vector-Jacobian product $\nabla_\gamma \mathcal{L} = (\partial\bar{\mu}^\gamma/\partial\gamma)^\top \nabla_\mu \ell$. This product can be computed efficiently by solving the adjoint linear system $(I - \mathcal{J}_\mu^\top)\, w = \nabla_\mu \ell$ for w, and then evaluating $\mathcal{J}_\gamma^\top w$. Crucially, this adjoint system can be solved via a fixed-point iteration that only requires matrix-vector products with $\mathcal{J}_\mu^\top$ (computed cheaply by automatic differentiation), without ever forming or storing $\mathcal{J}_\mu$ explicitly.

The following two propositions formalize this approach: Proposition B.1 establishes the convergence of the adjoint iteration and its equivalence with the exact implicit gradient from equation B.1, while Prop. B.2 establishes the Lipschitz continuity of the equilibrium map Φ in γ.

Proposition B.1 (Correctness of Adjoint Gradient Computation)**.** *Let Assumptions B.1 and B.2 hold. Fix an observed trajectory $\mu^{\mathrm{obs}} \in \mathbb{R}^{(N+1)\times d}$ and define the per-sample loss $\ell : \mathbb{R}^{(N+1)\times d} \to \mathbb{R}$ by $\ell(\mu) := \|\mu - \mu^{\mathrm{obs}}\|_{2,N}^2$, where μ^{obs} is held fixed. Define the composite loss $\mathcal{L}(\gamma) := \ell(\bar{\mu}^\gamma) = \|\bar{\mu}^\gamma - \mu^{\mathrm{obs}}\|_{2,N}^2$, where $\bar{\mu}^\gamma = \Phi(\gamma, \mu_0)$ is the equilibrium trajectory.*

Consider the adjoint sequence initialized at $w^{(0)} = 0 \in \mathbb{R}^{(N+1)d}$ and defined by

$$w^{(k+1)} = \mathcal{J}_\mu^\top\, w^{(k)} + \nabla_\mu \ell\big|_{\bar{\mu}^\gamma}. \tag{B.2}$$

Then:

(i) The sequence $(w^{(k)})_{k\ge 0}$ converges to a unique limit $w^* \in \mathbb{R}^{(N+1)d}$.

(ii) The gradient of the composite loss with respect to γ satisfies

$$\nabla_\gamma \mathcal{L} = \mathcal{J}_\gamma^\top\, w^*, \tag{B.3}$$

which equals the exact implicit gradient from equation B.1 composed with $\nabla_\mu \ell$.

Proof. Part (i). Let $A := \mathcal{J}_\mu \in \mathbb{R}^{(N+1)d\times(N+1)d}$ and $v := \nabla_\mu \ell\big|_{\bar{\mu}^\gamma} \in \mathbb{R}^{(N+1)d}$. The iteration equation B.2 reads $w^{(k+1)} = A^\top w^{(k)} + v$. Its unique fixed point, if it exists, is $w^* = (I - A^\top)^{-1} v$.

By Assumption B.1, $\|A\|_\infty \le \kappa < 1$. Since the spectral radius satisfies $\rho(A) \le \|A\|_\infty < 1$ for any induced matrix norm, and A and $A^\top$ share the same eigenvalues, we have $\rho(A^\top) = \rho(A) < 1$. Consequently, $(A^\top)^k \to 0$ as $k \to \infty$, which implies both that $(I - A^\top)$ is invertible (with Neumann series $(I - A^\top)^{-1} = \sum_{j=0}^\infty (A^\top)^j$) and that the iteration converges. Indeed, $w^{(k)} - w^* = (A^\top)^k (w^{(0)} - w^*) \to 0$.

Part (ii). Since $\mathcal{L}(\gamma) = \ell(\bar{\mu}^\gamma)$ and $\bar{\mu}^\gamma = \Phi(\gamma, \mu_0)$, the chain rule gives

$$\nabla_\gamma \mathcal{L} = \left(\frac{\partial \bar{\mu}^\gamma}{\partial \gamma}\right)^\top \nabla_\mu \ell\big|_{\bar{\mu}^\gamma}.$$

Substituting the implicit derivative equation B.1 and transposing:

$$\nabla_\gamma \mathcal{L} = \left((I - \mathcal{J}_\mu)^{-1}\mathcal{J}_\gamma\right)^\top v = \mathcal{J}_\gamma^\top \left(I - \mathcal{J}_\mu^\top\right)^{-1} v = \mathcal{J}_\gamma^\top w^*,$$

where the last equality uses $w^* = (I - A^\top)^{-1}v$ from Part (i). This is precisely equation B.3. □

Remark B.2 (Practical approximation). *Proposition B.1 assumes that the Picard solver has converged to the exact fixed point $\bar{\mu}^\gamma$. In practice, the solver is terminated after K iterations, yielding an approximate equilibrium $\mu^{(K)}$. By the contraction property (Assumption B.1), $\|\mu^{(K)} - \bar{\mu}^\gamma\|_\infty \le \kappa^K \|\mu^{(0)} - \bar{\mu}^\gamma\|_\infty$, so the fixed-point approximation error decays geometrically. Quantifying the corresponding finite-iteration error in the implicit gradient would require additional regularity of the Jacobians and of the adjoint solve; all theoretical statements below are stated for the exact fixed point.*

Proposition B.2 (Lipschitz Continuity of the Equilibrium Map). *Let Assumption B.1 hold, and suppose $\Xi(\mu, \gamma)$ is $L_{\Xi,\gamma}$-Lipschitz in γ uniformly over $\mu \in \mathcal{P}([d])^{N+1}$:*

$$\|\Xi(\mu, \gamma_1) - \Xi(\mu, \gamma_2)\|_\infty \le L_{\Xi,\gamma} \|\gamma_1 - \gamma_2\|_\infty, \qquad \forall\, \mu \in \mathcal{P}([d])^{N+1}.$$

Then, for any initial distribution μ_0, the equilibrium map $\Phi(\cdot, \mu_0)$ of the Picard solver is well-defined and Lipschitz in γ: for any two parameter sequences γ_1, γ_2,

$$\|\Phi(\gamma_1, \mu_0) - \Phi(\gamma_2, \mu_0)\|_\infty \le \frac{L_{\Xi,\gamma}}{1 - \kappa} \|\gamma_1 - \gamma_2\|_\infty. \tag{B.4}$$

Proof. By Assumption B.1 and the Banach fixed-point theorem, for each γ the fixed point $\bar{\mu}^\gamma = \Xi(\bar{\mu}^\gamma, \gamma)$ is unique, and $\Phi(\gamma, \mu_0) := \bar{\mu}^\gamma$.

Write $\mu_j := \Phi(\gamma_j, \mu_0) = \bar{\mu}^{\gamma_j}$ for $j = 1, 2$, so $\mu_j = \Xi(\mu_j, \gamma_j)$. By the triangle inequality, the κ-contraction property of $\Xi(\cdot, \gamma_1)$, and the $L_{\Xi,\gamma}$-Lipschitz property of Ξ in γ:

$$\begin{aligned}
\|\mu_1 - \mu_2\|_\infty &= \|\Xi(\mu_1, \gamma_1) - \Xi(\mu_2, \gamma_2)\|_\infty \\
&\le \|\Xi(\mu_1, \gamma_1) - \Xi(\mu_2, \gamma_1)\|_\infty + \|\Xi(\mu_2, \gamma_1) - \Xi(\mu_2, \gamma_2)\|_\infty \\
&\le \kappa \|\mu_1 - \mu_2\|_\infty + L_{\Xi,\gamma} \|\gamma_1 - \gamma_2\|_\infty.
\end{aligned}$$

Since $\kappa < 1$, rearranging gives $\|\mu_1 - \mu_2\|_\infty \le \frac{L_{\Xi,\gamma}}{1-\kappa} \|\gamma_1 - \gamma_2\|_\infty$. □

Proposition B.2 establishes that the equilibrium map Φ is Lipschitz in γ with constant $L_{\Xi,\gamma}/(1 - \kappa)$; this sensitivity bound is also used in the proof of the generalization result below.

Generalization to unseen trajectories. In practice, the neural network φ_θ is trained on a finite dataset of observed trajectories. A natural question is whether the learned parameters generalize to new, unseen trajectories drawn from the same distribution. The following proposition provides a uniform bound on the gap between the population risk (expected loss on new data) and the empirical risk (average loss on training data). The key insight is that the implicit layer Φ inherits Lipschitz regularity from the Picard operator (via Proposition B.2), which in turn controls the complexity of the loss landscape.

Proposition B.3 (Uniform Generalization Bound). *Let Assumption B.1 hold, and assume $\Xi(\mu, \gamma)$ is $L_{\Xi,\gamma}$-Lipschitz in γ uniformly over μ, as in Proposition B.2. Let the neural network parameter space $\Theta \subset \mathbb{R}^p$ be compact with ℓ^2-diameter $D := \sup_{\theta,\theta' \in \Theta} \|\theta - \theta'\|_2$. Assume the network map $\theta \mapsto \varphi_\theta(\cdot, \mu)$ is $L_{\varphi,\theta}$-Lipschitz in θ (in the ℓ^2 norm) uniformly over μ. Suppose the training data $\mathcal{S} = \{(\mu^{\mathrm{obs},i}, \mu_0^i)\}_{i=1}^M$ consists of M i.i.d. samples from a distribution $\mathcal{P}$. Define the per-sample loss $\ell(\theta; \mu^{\mathrm{obs}}, \mu_0) := \|\mu^{\mathrm{obs}} - \Phi(\varphi_\theta(\cdot, \mu^{\mathrm{obs}}), \mu_0)\|_{2,N}^2$, where $\|\cdot\|_{2,N}$ denotes the discrete L^2 norm introduced above. Since for any two simplex-valued discrete trajectories μ, ν one has $\|\mu - \nu\|_{2,N}^2 \le 2(N+1)\Delta t$, the loss is bounded by $B := 2(N+1)\Delta t = 2T(1 + 1/N)$. Define the empirical and population risks as $\hat{\mathcal{R}}(\theta) = \frac{1}{M}\sum_{i=1}^M \ell(\theta; \mu^{\mathrm{obs},i}, \mu_0^i)$ and $\mathcal{R}(\theta) = \mathbb{E}_{\mathcal{P}}[\ell(\theta; \mu^{\mathrm{obs}}, \mu_0)]$. Then, for any $\delta \in (0, 1)$ and $\epsilon > 0$, with probability at least $1 - \delta$ over the draw of $\mathcal{S}$, for all $\theta \in \Theta$:*

$$\mathcal{R}(\theta) \le \hat{\mathcal{R}}(\theta) + 2L_\ell \epsilon + B\sqrt{\frac{\log(2\mathcal{N}(\epsilon, \Theta)/\delta)}{2M}}, \tag{B.5}$$

where $\mathcal{N}(\epsilon, \Theta)$ denotes the ϵ-covering number of Θ in the ℓ^2 norm, $C_{N,d} := \sqrt{dT(1 + 1/N)}$, and $L_\ell = 2\sqrt{B} \cdot C_{N,d} \cdot \frac{L_{\Xi,\gamma}}{1-\kappa} \cdot L_{\varphi,\theta}$ is the Lipschitz constant of ℓ with respect to θ.

For fixed ϵ, δ, and parameter class Θ, the stochastic term in this bound scales as $M^{-1/2}$; the additional term $2L_\ell\epsilon$ is the price of passing from a finite cover to the full parameter class. For a bounded subset of $\mathbb{R}^p$ with diameter D one may use the standard bound $\mathcal{N}(\epsilon, \Theta) \le (1 + 2D/\epsilon)^p$.

Proof. We first bound the Lipschitz constant L_ℓ of $\theta \mapsto \ell(\theta; \mu^{\text{obs}}, \mu_0)$. For $\theta_1, \theta_2 \in \Theta$, write $\bar{\mu}_j := \Phi(\varphi_{\theta_j}(\cdot, \mu^{\text{obs}}), \mu_0)$ for $j = 1, 2$. Then:

$$|\ell(\theta_1) - \ell(\theta_2)| = \left| \|\mu^{\text{obs}} - \bar{\mu}_1\|_{2,N}^2 - \|\mu^{\text{obs}} - \bar{\mu}_2\|_{2,N}^2 \right| \le 2\sqrt{B}\, \|\bar{\mu}_1 - \bar{\mu}_2\|_{2,N},$$

using the identity $|a^2 - b^2| \le (a + b)|a - b|$ and $\|\mu^{\text{obs}} - \bar{\mu}_j\|_{2,N} \le \sqrt{B}$.

By the norm comparison $\|\cdot\|_{2,N} \le C_{N,d}\, \|\cdot\|_\infty$ and the Lipschitz bound on Φ from Proposition B.2:

$$\|\bar{\mu}_1 - \bar{\mu}_2\|_\infty \le \frac{L_{\Xi,\gamma}}{1 - \kappa}\, \|\varphi_{\theta_1} - \varphi_{\theta_2}\|_\infty \le \frac{L_{\Xi,\gamma}}{1 - \kappa}\, L_{\varphi,\theta}\, \|\theta_1 - \theta_2\|_2.$$

Combining: $L_\ell = 2\sqrt{B}\, C_{N,d}\, \frac{L_{\Xi,\gamma}}{1-\kappa}\, L_{\varphi,\theta}$.

The generalization bound then follows from a standard covering number argument. Let $C_\epsilon \subset \Theta$ be an ϵ-cover with $|C_\epsilon| = \mathcal{N}(\epsilon, \Theta)$. For any θ, there exists $\theta_j \in C_\epsilon$ with $\|\theta - \theta_j\|_2 \le \epsilon$, so $|\mathcal{R}(\theta) - \mathcal{R}(\theta_j)| \le L_\ell\epsilon$ and $|\hat{\mathcal{R}}(\theta) - \hat{\mathcal{R}}(\theta_j)| \le L_\ell\epsilon$. By Hoeffding's inequality applied to each fixed θ_j (with ℓ bounded by B) and a union bound over C_ϵ:

$$\mathbb{P}\left(\max_{\theta_j \in C_\epsilon} \left(\mathcal{R}(\theta_j) - \hat{\mathcal{R}}(\theta_j) \right) \ge t \right) \le \mathcal{N}(\epsilon, \Theta)\, \exp\left(-\frac{2Mt^2}{B^2} \right).$$

Choosing $t = B\sqrt{\frac{\log(2\mathcal{N}(\epsilon,\Theta)/\delta)}{2M}}$ makes the right-hand side at most $\delta/2$, hence at most δ; the extra factor 2 in the logarithm is harmless for this one-sided bound. For any θ: $\mathcal{R}(\theta) \le \mathcal{R}(\theta_j) + L_\ell\epsilon \le \hat{\mathcal{R}}(\theta_j) + t + L_\ell\epsilon \le \hat{\mathcal{R}}(\theta) + t + 2L_\ell\epsilon$. □

In the next section, we illustrate this $M^{-1/2}$ finite-sample scaling on a simple sampled-parameter experiment.

C Empirical Illustration of the Generalization Bound

To empirically illustrate the $\mathcal{O}(M^{-1/2})$ scaling predicted by Proposition B.3, we evaluate a finite-sample proxy for the generalization gap. The core objective is to isolate the effect of the finite sample size (M) from optimization error; the observation-noise level is fixed throughout the experiment.

The procedure works by evaluating, for each candidate parameter $\theta \in \Theta$, the difference between the empirical risk $\hat{\mathcal{R}}_M(\theta)$ on a finite sample of M initial conditions and the population risk $\mathcal{R}(\theta)$ approximated via Monte Carlo integration over a large, separate test set. As a Monte Carlo proxy for the uniform gap in Proposition B.3, we sample many parameters θ uniformly at random from Θ and report the maximum gap over this sampled set. This process is repeated across multiple subsamples to compute the standard error. The full methodology is detailed in Algorithm 3.

Algorithm 3 Generalization Bound Empirical Verification.

Require: True parameter θ^*; training set sizes $\mathbb{M} = \{M_1, M_2, \ldots\}$; subsamples $S \in \mathbb{N}$; test size $N_{\text{test}} \in \mathbb{N}$; number of random parameters $P \in \mathbb{N}$; parameter space Θ

Generate N_{test} test initial conditions $\mu_0^{\text{test},i} \sim \text{Dir}(\mathbf{1})$ ▷ *Data Generation*
Simulate ground-truth test trajectories: $\bar{\mu}^{\text{test},i} \leftarrow \Phi(\theta^*, \mu_0^{\text{test},i})$ for $i \in \{1, \ldots, N_{\text{test}}\}$
Generate a large pool of N_{train} training initial conditions and corresponding true trajectories
Sample P parameters $\{\theta_p\}_{p=1}^P$ uniformly at random from Θ ▷ *No training; fixed θ*
for $M \in \mathbb{M}$ **do**
  for $s \in \{1, \ldots, S\}$ **do**
    Sample M trajectories $\{(\mu_0^j, \bar{\mu}^j)\}_{j=1}^M$ from the training pool ▷ *Draw a finite dataset*
    for $p \in \{1, \ldots, P\}$ **do**
      $\hat{\mathcal{R}}_M(\theta_p) \leftarrow \frac{1}{M} \sum_{j=1}^M \|\bar{\mu}^j - \Phi(\theta_p, \mu_0^j)\|_{2,N}^2$ ▷ *Empirical Risk*
      $\mathcal{R}(\theta_p) \leftarrow \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \|\bar{\mu}^{\text{test},i} - \Phi(\theta_p, \mu_0^{\text{test},i})\|_{2,N}^2$ ▷ *Population Risk*
    $\text{Gap}_{M,s} \leftarrow \sup_p \left|\mathcal{R}(\theta_p) - \hat{\mathcal{R}}_M(\theta_p)\right|$ ▷ *Supremum over sampled θ*
  $\mu_{\text{Gap}}(M) \leftarrow \frac{1}{S} \sum_{s=1}^S \text{Gap}_{M,s}$ ▷ *Mean Gap*
  $\sigma_{\text{Gap}}(M) \leftarrow \text{std}(\text{Gap}_M)/\sqrt{S}$ ▷ *Standard Error*
return $\{\mu_{\text{Gap}}(M), \sigma_{\text{Gap}}(M)\}_{M \in \mathbb{M}}$ for log-log regression analysis

Experimental setup. We apply Algorithm 3 to the Linear-Quadratic MFG model from Sec. 5 with a 2-dimensional linear parameterization $\gamma_\theta(t) = \theta_0 + \theta_1 t$, true parameter $\theta^* = (1.0, 2.0)$, and $d = 3$ discrete states. To study a sampled proxy for the uniform bound of Proposition B.3 without conflating optimization error, we bypass any training step: we evaluate the generalization gap $|\mathcal{R}(\theta) - \hat{\mathcal{R}}_M(\theta)|$ for $P = 500$ parameters θ drawn uniformly at random from $\Theta = [0.1, 5.0]^2$, and report the maximum over these sampled parameters. The hyperparameters are summarized in Table 1.

Table 1: Hyperparameters for the generalization bound verification experiment.

Parameter	Value
Discrete states (d)	3
Time horizon (T)	1.0
Number of points in time ($N+1$)	20
True parameter θ^*	$(1.0, 2.0)$
Parameter domain Θ	$[0.1, 5.0]^2$
Parameterization	Linear: $\gamma(t) = \theta_0 + \theta_1 t$
Noise level (Dirichlet κ^{-1})	10^{-4}
Training pool size N_{train}	5000
Test set size N_{test}	5000
Sample sizes $\mathbb{M}$	$\{1, 5, 10, 20, 50, 100, 200\}$
Number of random θ's	500
Subsamples S	100

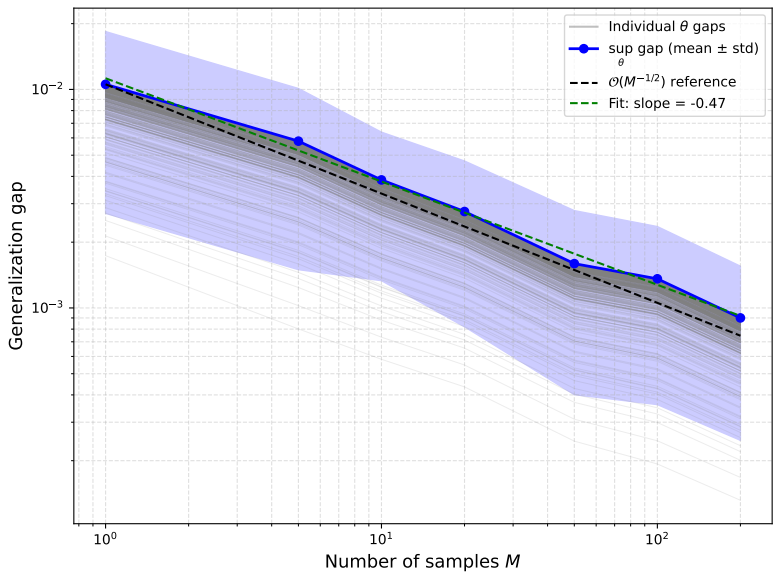


Figure 7: Empirical illustration of the generalization-bound scaling. Gray curves show the generalization gap $|\mathcal{R}(\theta) - \hat{\mathcal{R}}_M(\theta)|$ averaged over $S = 100$ subsamples for each of the 500 randomly sampled parameters $\theta \in \Theta$. The blue curve shows the maximum over the sampled parameters of these gaps, with shaded ± 1 standard deviation. The dashed black line is the theoretical $\mathcal{O}(M^{-1/2})$ reference. The fitted log-log slope of -0.50 is consistent with the predicted rate.

Discussion. Fig. 7 shows that the empirical supremum over the sampled parameters decays at a rate of roughly $M^{-0.50}$ in log-log scale, in agreement with the $\mathcal{O}(M^{-1/2})$ rate predicted by Proposition B.3. The gap ranges from approximately 10^{-2} at $M = 1$ to 9×10^{-4} at $M = 200$. The individual θ-level gaps (gray curves) all exhibit the same scaling behavior, and the supremum envelope (blue curve) closely tracks them, indicating that no single parameter dominates the worst case disproportionately. We note that the error bars (standard deviation across subsamples) shrink proportionally with the mean, confirming the stability of the Monte Carlo estimation procedure.

Remark C.1 (Gap between the theoretical constant and empirical observations)**.** *While the empirical generalization gap confirms the theoretically predicted $\mathcal{O}(M^{-1/2})$ convergence rate, the absolute value of the theoretical upper bound (Proposition B.3) is typically several orders of magnitude larger than the observed gap. This discrepancy arises from several compounding sources of conservatism:*

1. ***Worst-case loss bound*** *B. The bound assumes $\ell(\theta; \mu^{\text{obs}}) \leq B$ for all $\theta \in \Theta$ and all possible observations, where $B = 2T(1 + 1/N)$ bounds the squared L^2 distance between any two distributions on the simplex. In practice, the actual losses are concentrated in a narrow range far below this worst-case envelope.*

2. ***Parameter space diameter*** *D. The covering number $\mathcal{N}(\epsilon, \Theta) \leq (1 + 2D/\epsilon)^p$ grows polynomially with the diameter $D = \text{diam}(\Theta)$ of the full parameter space. In practice, the loss function varies appreciably only over a much smaller effective region around the true parameter θ^*.*

3. ***Distribution-free nature.*** *The Hoeffding–union bound argument holds for* any *data-generating distribution over initial conditions μ_0. It cannot exploit the specific concentration properties of the Dirichlet distribution or the smoothness of the MFG equilibrium map Φ, both of which reduce the variance of finite-sample averages in our setting.*

We emphasize that the primary contribution of Proposition B.3 is the rate *$\mathcal{O}(M^{-1/2})$ and its structural dependence on the Lipschitz constants of the MFG equilibrium map, rather than the absolute value of the constant.*

D Empirical Validation of the Picard Contraction Assumption

Assumption B.1 requires the discrete Picard operator $\Xi(\cdot, \gamma)$ to be a global κ-contraction in $\|\cdot\|_\infty$. This section reports an assessment of the validity of this assumption across state dimension d, damping coefficient α, convergence tolerance, and horizon length N. All reported quantities are computed from the ground-truth parameter trajectories, so that the diagnostics characterize the solver itself rather than a particular φ_θ.

Definitions. For each sampled instance, the (possibly damped) Picard iteration $\mu^{(k+1)} = \Xi_\alpha(\mu^{(k)}, \gamma)$ is run for up to N steps from $\mu_k^{(0)} \equiv \mu_0$ for all k. We report

- **Failure rate**: the fraction of instances for which $\|\mu^{(k+1)} - \mu^{(k)}\|_\infty$ does not fall below the solver tolerance within N iterations.
- **Mean iterations to convergence** (n_{iters}), with standard deviation, over converged instances (non-converged instances are counted at N).
- **Empirical contraction factor**: the mean of the step ratios $r_k = \|\mu^{(k+1)} - \mu^{(k)}\|_\infty / \|\mu^{(k)} - \mu^{(k-1)}\|_\infty$ over the mean field trajectory before its convergence.
- **Spectral radius**: a matrix-free power-iteration estimate of $\rho(\mathcal{J}_\mu)$ at the converged fixed point. Here, we use the Jacobian $\mathcal{J}_\mu = \partial\Xi/\partial\mu$ as it appears in Eq. B.1.

All sweeps use 100 instances per configuration.

A Note on the Locality of Observations. Both diagnostics above are *local* quantities, evaluated near the reached fixed point; neither directly measures the global supremum in Assumption B.1. Therefore, the tables below are merely included to provide evidence within the fixed regime.

Sweeping over State Dimensions. Table 2 varies the state dimension d at fixed horizon and damping $\alpha = 0$.

Table 2: Picard solver diagnostics as a function of state dimension d (damping $\alpha = 0$, tolerance 10^{-5}).

d	Failure rate	Mean iterations ($\pm$ std)	Contraction factor ($\pm$ std)
3	0.0%	5.26 ± 0.46	0.0682 ± 0.0022
4	0.0%	5.04 ± 0.31	0.0640 ± 0.0044
5	0.0%	5.49 ± 0.57	0.0739 ± 0.0145
6	9.0%	94.47 ± 284.78	0.0445 ± 0.0123

For $d \in \{3, 4, 5\}$ the solver converges without failures in roughly 5–6 iterations, with a small, stable empirical contraction factor around 0.06–0.07. At $d = 6$, the failure rate rises to 9%, and the mean iteration count and its standard deviation both strongly increase, indicating that a small subset of instances converges very slowly or exhausts the iteration budget, rather that there's a uniform slowdown across all instances. This is consistent with the solver approaching a lower contraction factor (under undamped iteration, which is in line with Remark B.1.

Calculating the Spectral Radius. To accompany the contraction factor with a Jacobian-based estimate, Table 3 reports the power-iteration estimate of $\rho(\mathcal{J}_\mu)$ at the reached fixed point, for $d \in \{3, 4\}$.

Table 3: Empirical contraction factor vs. power-iteration spectral radius estimate at the converged fixed point, for $d \in \{3, 4\}$.

d	Contraction factor ($\pm$ std)	Spectral radius $\rho(\mathcal{J}_\mu)$ ($\pm$ std)
3	0.0682 ± 0.0022	$0.0685 \pm 2.3 \times 10^{-7}$
4	0.0640 ± 0.0044	$0.0650 \pm 3.9 \times 10^{-7}$

The two estimates agree closely at both dimensions, while the power-iteration estimates are associated with low noise levels. This suggests that both diagnostics are measuring the same underlying local contraction rate.

Damping and Tolerance Table 4 fixes $d = 4$ and varies the damping coefficient $\alpha \in \{0, 0.3, 0.6, 0.9\}$ and the convergence tolerance.

Table 4: Picard solver diagnostics as a function of damping coefficient α and tolerance, at $d = 4$.

α	tol	Failure rate	Mean iters ($\pm$ std)	Contraction factor ($\pm$ std)
0.0	10^{-3}	0.0%	3.70 ± 0.46	0.0624 ± 0.0033
0.0	10^{-5}	0.0%	5.05 ± 0.33	0.0643 ± 0.0020
0.0	10^{-7}	2.0%	10.89 ± 27.02	0.0985 ± 0.1339
0.3	10^{-3}	0.0%	6.06 ± 0.24	0.2838 ± 0.0130
0.3	10^{-5}	0.0%	9.99 ± 0.22	0.2905 ± 0.0108
0.3	10^{-7}	0.0%	13.94 ± 0.31	0.2968 ± 0.0089
0.6	10^{-3}	0.0%	11.73 ± 0.47	0.5887 ± 0.0077
0.6	10^{-5}	0.0%	20.55 ± 0.59	0.5915 ± 0.0074
0.6	10^{-7}	0.0%	29.61 ± 0.63	0.5933 ± 0.0067
0.9	10^{-3}	0.0%	38.65 ± 2.28	0.8968 ± 0.0019
0.9	10^{-5}	0.0%	81.28 ± 2.18	0.8974 ± 0.0020
0.9	10^{-7}	0.0%	124.80 ± 2.46	0.8979 ± 0.0018

At $d = 4$, the only failures observed anywhere in this sweep occur at $\alpha = 0$ with the tightest tolerance (10^{-7}), where a small fraction (2%) of instances fail to converge and the iteration count becomes highly variable (10.89 ± 27.02). Any damping $\alpha \geq 0.3$ eliminates failures entirely across all tested tolerances, at the cost of a substantially larger and tolerance-dependent iteration count (e.g., $\alpha = 0.9$ requires 39–125 iterations depending on tolerance, versus 4–11 at $\alpha = 0$). The contraction factor of the damped map increases with α (from ≈ 0.06 at $\alpha = 0$ to ≈ 0.90 at $\alpha = 0.9$). This is as expected since Ξ_α's Lipschitz constant interpolates toward α as $\alpha \to 1$. This is direct empirical support for the qualitative claim in Remark B.1 that a sufficiently large damping coefficient restores reliable convergence, at a quantified cost in iteration count.

Refining the Temporal Discretization Table 5 fixes $d = 4$, $\alpha = 0$ with a tolerance of 10^{-5} and varies the number of time steps N.

Table 5: Picard solver diagnostics as a function of the number of discretization steps N, at $d = 4$.

N	Failure rate	Mean iters ($\pm$ std)	Contraction factor ($\pm$ std)
25	14.0%	33.01 ± 67.38	0.1054 ± 0.0139
50	6.0%	16.99 ± 46.24	0.0709 ± 0.0045
100	0.0%	5.05 ± 0.33	0.0643 ± 0.0020
250	0.0%	5.00 ± 0.14	0.0624 ± 0.0016
500	0.0%	4.99 ± 0.10	0.0606 ± 0.0024

Failure rate and iteration-count variance both decrease monotonically as N increases, with the solver being failure-free and low-variance for $N \geq 100$.

E Experimental Details

E.1 Synthetic Data Generation

In the LQ and cybersecurity setting, we generate synthetic data by solving the forward-backward MFG system with true parameters sampled from specific functional forms. Specifically, we consider three forms for all true parameters $\gamma_i \in [\gamma_{i,\min}, \gamma_{i,\max}]$:

1. **Constant**: $\gamma_i(t, \mu) = (\gamma_{i,\min} + \gamma_{i,\max})/2$.
2. **Time-dependent**: $\gamma_i(t, \mu) = \gamma_{i,\min} + \frac{4(\gamma_{i,\max} - \gamma_{i,\min})}{T^2} t(T - t)$.
3. **Mean field dependent**: $\gamma_i(t, \mu) = \gamma_{i,\min} + (\gamma_{i,\max} - \gamma_{i,\min})\mu(0)$, scaling linearly with the mass in the first state.

All hyperparameters and instantiations for $\gamma_{i,\min}$ and $\gamma_{i,\max}$ are reported in App. J.

Noise Robustness. We evaluate the robustness of our approach by introducing Dirichlet noise to the observed mean field distributions. Specifically, for a given noise level $\sigma > 0$, the noisy observation $\tilde{\mu}_t$ at each time step is sampled from a Dirichlet distribution:

$$\tilde{\mu}_t \sim \text{Dir}(p_t), \quad \text{where} \quad p_t = \frac{1}{\sigma}(\mu_t + \epsilon \mathbf{1}) \tag{E.1}$$

Here, μ_t is the true mean field distribution, $\epsilon = 10^{-6}$ is a small constant to ensure strict positivity, and σ acts as the variance-controlling noise level (with smaller σ yielding samples tightly concentrated around the true mean). We apply this analysis to the synthetic linear-quadratic and cybersecurity settings, where we have full control over the noise.

E.2 Linear-Quadratic

In the LQ MFG, the players are free to choose their transition rates a_{xy} between the states without being influenced by predefined dynamics. In the absence of predefined transitions, the game is flexible and can be scaled to an arbitrary number of states d. The players' transition rates between those states are subject to the following running and terminal costs,

$$f^{\gamma}(x, a, \mu) := b \sum_{y \neq x} (a_{xy} - 2)^2 + \mu(x); \quad g(x, \mu) = \frac{1}{2}\mu(x)^2. \tag{E.2}$$

Here, the parameter of interest is $\gamma \equiv b \in \mathbb{R}$, which determines the cost associated with the state transitions, while the addition of $\mu(x)$ penalizes the players for being in crowded states. As a consequence, large values for b disincentivize the players to deviate from $a_{xy} = 2$, which is expected to result in a smoothly evolving mean field distribution μ. In contrast, small values of b increase the dominance of $\mu(x)$ in the running cost, resulting in a fast redistribution of the population over the states. Due to the linear-quadratic structure of the game, an analytical solution for the optimal transition rate a^* can be obtained through a first-order minimization of the Hamiltonian Bensoussan et al. (2016):

$$\begin{aligned} H_b(t, x, \mu, u) &= \mu(x) + \inf_{a \in [1,3]^{d-1}} \left\{ \sum_{y \neq x} b(a_{xy} - 2)^2 + a_{xy}(u(y) - u(x)) \right\}, \\ a^*_{xy}(x, u) &= \Pi_{[1,3]}\left(2 - \frac{u(y) - u(x)}{2b}\right). \end{aligned} \tag{E.3}$$

where $\Pi_{[1,3]}(z) = \min\{3, \max\{1, z\}\}$.

E.3 Cybersecurity

The cybersecurity game considers a botnet-like population of computers that may become victims of a cyberattack. Each computer can be defended against such attacks by updating its protection level. In this case, the computer moves from the Unprotected (U) to the Defended (D) state. An update takes the time $1/\lambda > 0$ to complete. In addition, each computer is either Susceptible (S) to the attack, or already Infected (I) with the malware. Therefore the $d = 4$ states are UI, US, DI, DS; in the transition matrix below we order rows and columns as (DI, DS, UI, US). The game admits two actions $\mathcal{A} = \{0, 1\}$. If $a = 0$ then the security level of the computer remains unchanged. For $a = 1$, the security level of the computer is updated, effectively reversing its protection level.

Depending on their protection level, infected computers can recover to the susceptible state at a recovery rate q^D_{rec} or q^U_{rec}. There are two ways for a susceptible computer to become infected: either (1) directly, via an attack from the hacker, or (2) indirectly, via an infected computer. Direct attacks occur at the rates $v_H q^D_{inf}$ and $v_H q^U_{inf}$, for the defended and unprotected computers, respectively. The rate of infection through indirect attacks depends on the distribution of states μ among the vulnerable computers. A computer in DI infects a computer in DS with rate $\beta_{DD}\mu(DI)$ and a computer in US with rate $\beta_{DU}\mu(DI)$. Similarly, computer in UI infects a computer in DS with rate $\beta_{UD}\mu(UI)$ and a computer in US with rate $\beta_{UU}\mu(UI)$. The resulting transition matrix is presented in equation E.4.

$$Q^{\gamma}(\cdot, \cdot, \mu, v_H, a) = \begin{array}{c} \\ DI \\ DS \\ UI \\ US \end{array} \begin{array}{c} \begin{array}{cccc} DI & DS & UI & US \end{array} \\ \begin{bmatrix} \dots & q^D_{rec} & \mathbf{1}_{\{1\}}(a)\lambda & 0 \\ \begin{matrix} v_H q^D_{inf} \\ +\beta_{DD}\mu(DI) \\ +\beta_{UD}\mu(UI) \end{matrix} & \dots & 0 & \mathbf{1}_{\{1\}}(a)\lambda \\ \mathbf{1}_{\{1\}}(a)\lambda & 0 & \dots & q^U_{rec} \\ 0 & \mathbf{1}_{\{1\}}(a)\lambda & \begin{matrix} v_H q^U_{inf} \\ +\beta_{UU}\mu(UI) \\ +\beta_{DU}\mu(DI) \end{matrix} & \dots \end{bmatrix} \end{array} \tag{E.4}$$

Each computer owner aims to minimize the running cost that combines k_D, the costs of the defense system per unit of time, and the k_I per unit of time loss associated with an infected device:

$$f^{\gamma}(t, x, a, \mu) = k_D \mathbf{1}_{\{D\}}(x) + k_I \mathbf{1}_{\{I\}}(x). \tag{E.5}$$

We model the unknown parameters $\gamma \equiv (q^D_{rec}, q^U_{rec}, q^D_{inf}, q^U_{inf}, \beta_{DD}, \beta_{DU}, \beta_{UU}, \beta_{UD})$. The parameters λ and v_H are not learned, as they describe the unobservable dynamics of the representative player (λ) or are inseparable from q^U_{inf} and q^D_{inf} (v_H).

E.4 Susceptible-Infected-Recovered

In real-world infectious disease dynamics, state transitions are not just influenced by parameters that are associated with the disease. Instead, the infectious disease transmission also depends on the intensity of contact between individuals. This is the player's control: each player has the option to minimize their contact with others, for example, by minimizing their movements or wearing protective equipment. Therefore, we define the transition rate of a susceptible representative agent to the infected state as $\beta\alpha_t\mu_t(I)$. Here, β denotes the base infection rate, α_t is the socialization-level control of the susceptible individual, and $\mu_t(x)$ is the fraction of the population that is in state x at time t. We implicitly assume that the infected individuals will follow any distancing guidelines that are suggested to them. We keep the state transitions for infected and recovered players (i.e., the recovery rate, respectively) fixed, making them dependent on disease properties only. Because we are modelling the viral dynamics per season, we assume that the players retain full immunity after having recovered. Summarizing these dynamics, we construct the transition matrix as

$$Q^{\gamma}(a_t, \mu_t) = \begin{bmatrix} -\beta a_t \mu_t(I) & \beta a_t \mu_t(I) & 0 \\ 0 & -\gamma & \gamma \\ 0 & 0 & 0 \end{bmatrix}, \tag{E.6}$$

where the row-wise sum of the elements is 0 by design to ensure that $Q^{\gamma} \in \mathbb{M}^d$. The agents face a cost for being ill c_I and for having a low contact factor c_α in the Susceptible state

$$f^{\gamma}(t, x, a, \mu, c_\alpha, c_I) = \frac{c_\alpha}{2}(1 - \alpha_t(x))^2 \mathbf{1}_{\{S\}}(x) + c_I \mathbf{1}_{\{I\}}(x). \tag{E.7}$$

Both cost weights are learned by the neural network. When these weights are small, the strategic component has little influence on the fitted dynamics; however, in the degenerate case $c_\alpha = c_I = 0$, the optimal control is not uniquely selected, so any mean-field calibration baseline requires an explicit selection rule, such as fixing $\alpha_t \equiv 1$.

E.5 Urban Mobility

To further explore the relevance of the MFG mechanics, we consider a bike-sharing network where commuters strategically choose their destinations based on station congestion. We compare two models calibrated on the same data: a *mean field dynamics* model (baseline), which describes population flow via exogenous transition rates, and a *mean field game* model, which additionally captures strategic congestion avoidance through an optimal control formulation. The MFG model contains the forward dynamics baseline as a special case: if the congestion/general-cost term is zero, $w_g = 0$, and the zero HJB solution is selected, then $a^*_{xy} = q^{\text{base}}_{xy}$ and the FPK equation reduces to the baseline.

Mean field dynamics model (baseline). In this model, agents follow exogenous, time-dependent transition rates without any strategic behavior. The population distribution evolves according to the Kolmogorov forward equation:

$$\dot{\mu}_i = \sum_{j \neq i} \mu_j \, q_{ji}(\gamma) - \mu_i \sum_{j \neq i} q_{ij}(\gamma), \tag{E.8}$$

where the rates $q_{ij}(t) = \text{softplus}(\gamma_{ij}(t))$ are parameterized by a neural network $\varphi_\theta : [0, T] \to \mathbb{R}^{d(d-1)}$. Depending on the test regime (see below), φ_θ is either a constant vector, a single affine layer, or a multi-layer perceptron. The network is trained by minimizing the L_2 discrepancy between the predicted trajectory μ^{γ_θ} and the observed distribution $\hat{\mu}$, analogously to the procedure described in Sec. 3.

Mean field game model. In the MFG formulation, each commuter optimizes their destination choice to minimize a cost that penalizes both deviating from the natural demand pattern and arriving at congested stations. The running cost is:

$$f(x, a, \mu) = b \sum_{y \neq x} (a_{xy} - q_{xy}^{\text{base}})^2 + c \sum_{y \neq x} \mu(y)\, a_{xy}, \tag{E.9}$$

where $q_{xy}^{\text{base}}(t) = \text{softplus}(\gamma_{xy}(t))$ are the same base rates as in the baseline model, $b > 0$ controls the cost of deviating from the natural demand, and $c \geq 0$ weights the destination congestion penalty. The terminal cost is $g(x) = w_g(x)$, where $w_g \in \mathbb{R}^d$ is a learnable vector initialized to zero. The Hamiltonian minimization yields the optimal transition rate:

$$a_{xy}^* = \max\left(0,\ q_{xy}^{\text{base}} - \frac{c\,\mu(y) + u(y) - u(x)}{2b}\right), \tag{E.10}$$

where u is the value function obtained from the backward HJB equation with terminal condition $u(T, x) = w_g(x)$.

Physical interpretation. The baseline model captures the dominant origin-destination demand patterns, morning commute from residential to commercial areas, and the evening reverse flow, through the learned rates $q_{ij}^{\text{base}}(t)$. The MFG model adds a strategic correction: when a destination station y is crowded ($\mu(y)$ is large), the effective rate a_{xy}^* is reduced by $c\mu(y)/(2b)$. In Tests 1–2, the implementation additionally clips these rates above at 100 for numerical stability; Test 3 uses only the nonnegativity projection shown in the formula. Furthermore, the backward equation provides *forward-looking* congestion avoidance through the value function u, allowing agents to anticipate future bottlenecks rather than merely reacting to current occupancy.

Experimental setup. For the MFG model, the cost parameters are initialized to $b_0 = 1.0$ and $c_0 = 0.5$, and $w_g = 0$. All parameters (θ, b, c, w_g) are jointly optimized using the Adam optimizer Kingma & Ba (2014). The MFG equilibrium is computed at each training step using the damped Picard iteration described in App. A. We report the mean field L_2 and relative L_2 prediction errors on both training and held-out test trajectories, averaged over 5 seeds.

F Mean Field Control: Formulation and Optimality Conditions

While MFGs describe the Nash equilibrium of non-cooperative agents, Mean Field Control (MFC) describes the social optimum where a central planner minimizes the aggregate cost of the population. In this section, we derive the optimality conditions for MFC in a finite-state space setting using the calculus of variations, following the approach of Bensoussan et al. (2013).

F.1 Problem Definition

Consider a population of agents distributed over a finite-state space $[d]$. The central planner aims to choose a feedback control policy $\alpha_t = (\alpha_{t,xy})_{x,y \in [d]}$ to minimize the total cost functional:

$$J(\alpha) = \int_0^T \sum_{x \in [d]} f_\gamma(t, x, \alpha_t(x), \mu_t)\mu_t(x)dt + \sum_{x \in [d]} g_\gamma(x, \mu_T)\mu_T(x), \tag{F.1}$$

subject to the Fokker-Planck equation (forward ODE) for the mean field $\mu_t \in \mathcal{P}([d])$:

$$\partial_t \mu_t(x) = \sum_{y \neq x} (\mu_t(y)\alpha_{t,yx} - \mu_t(x)\alpha_{t,xy}), \quad \mu_0(x) \text{ given.} \tag{F.2}$$

For simplicity, this appendix restricts to rate-control dynamics of the form equation F.2, where the control variables $\alpha_{t,xy}$ are the transition rates themselves. This covers the LQ MFC example, where $\gamma_\theta(t, \mu)$ enters the running cost, and hence the optimal rates, but not an additional primitive generator $Q^\gamma(t, \alpha, \mu)$. Extending the derivation to a general parameter-dependent generator would require adding the corresponding derivatives of the dynamics to the adjoint equation. For clarity, the variational derivation below is written in the interior case. When transition rates are constrained to be nonnegative or bounded, the pointwise minimization condition should be read as the corresponding variational inequality condition over the admissible rate set.

F.2 The Optimality System

The necessary conditions for the optimal control α^* are characterized by a forward-backward system of ODEs. Specifically, there exists an adjoint state $u_t(x)$ such that:

1. **Forward Equation**: The mean field μ_t satisfies the FPK equation equation F.2 with the optimal control α^*.
2. **Backward Adjoint Equation**: The adjoint state u_t satisfies:

$$-\partial_t u_t(x) = \sum_{y \neq x} \alpha^*_{t,xy}(u_t(y) - u_t(x)) + f_\gamma(t, x, \alpha^*_t, \mu_t) + \sum_{y \in [d]} \mu_t(y) \left[\frac{\partial f_\gamma}{\partial \mu(x)} + \frac{\partial f_\gamma}{\partial \gamma} \frac{\partial \gamma}{\partial \mu(x)} \right], \tag{F.3}$$

   where the partial derivatives inside the sum are evaluated at $(t, y, \alpha^*_t(y), \mu_t)$. The terminal condition is given by:

$$u_T(x) = g_\gamma(x, \mu_T) + \sum_{y \in [d]} \mu_T(y) \left[\frac{\partial g_\gamma}{\partial \mu(x)} + \frac{\partial g_\gamma}{\partial \gamma} \frac{\partial \gamma}{\partial \mu(x)} \right]. \tag{F.4}$$

3. **Optimality Condition**: At each time t, the control $\alpha^*_t(x)$ minimizes the Hamiltonian:

$$\alpha^*_t(x) \in \arg\min_a \left[f_\gamma(t, x, a, \mu_t) + \sum_{y \neq x} a_{xy}(u_t(y) - u_t(x)) \right]. \tag{F.5}$$

F.3 Derivation via Calculus of Variations

We use a perturbation argument to derive the adjoint equation. Let α be the optimal control and let β be an arbitrary perturbation. We define $\alpha^\epsilon = \alpha + \epsilon\beta$. Let μ^ϵ be the mean field corresponding to α^ϵ, and write $\mu^\epsilon = \mu + \epsilon\nu + \mathcal{O}(\epsilon^2)$, where ν satisfies the linearized dynamics:

$$\partial_t \nu_t(x) = \sum_{y \neq x} (\nu_t(y)\alpha_{yx} + \mu_t(y)\beta_{yx} - \nu_t(x)\alpha_{xy} - \mu_t(x)\beta_{xy}), \quad \nu_0(x) = 0. \tag{F.6}$$

The first-order variation of the cost functional equation F.1 is given by:

$$\delta J = \int_0^T \sum_{x \in [d]} \left[\frac{\partial f_\gamma}{\partial \alpha} \beta_t(x) \mu_t(x) + \left(f_\gamma + \sum_{y \in [d]} \mu_t(y) \frac{df_\gamma(t, y, \alpha_t(y), \mu_t)}{d\mu_t(x)} \right) \nu_t(x) \right] dt + \sum_{x \in [d]} \left[g_\gamma + \sum_{y \in [d]} \mu_T(y) \frac{dg_\gamma(y, \mu_T)}{d\mu_T(x)} \right] \nu_T(x), \tag{F.7}$$

where $\frac{df_\gamma}{d\mu(x)}$ denotes the total derivative accounting for the implicit dependence of γ on μ. To eliminate the dependence on ν, we introduce the adjoint state $u_t(x)$ and observe that for any u_t:

$$\int_0^T \sum_x u_t(x) \left(\partial_t \nu_t(x) - \sum_{y \neq x} (\nu_t(y)\alpha_{yx} + \mu_t(y)\beta_{yx} - \nu_t(x)\alpha_{xy} - \mu_t(x)\beta_{xy}) \right) dt = 0. \tag{F.8}$$

Integrating by parts $\int u \partial_t \nu dt = [u\nu]_0^T - \int \partial_t u \nu dt$ and using $\nu_0 = 0$:

$$\sum_x u_T \nu_T - \int_0^T \sum_x \left[\partial_t u + \sum_{y \neq x} \alpha_{xy}(u(y) - u(x)) \right] \nu dt - \int_0^T \sum_x \mu(x) \sum_{y \neq x} \beta_{xy}(u(y) - u(x)) dt = 0. \tag{F.9}$$

Substituting this expression into δJ, we require the terms involving ν and ν_T to cancel out. This leads directly to the backward adjoint ODE and the terminal condition:

$$-\partial_t u_t(x) = \sum_{y \neq x} \alpha_{xy}(u(y) - u(x)) + f_\gamma + \sum_y \mu(y) \frac{df_\gamma}{d\mu(x)}, \quad u_T(x) = g_\gamma + \sum_y \mu_T(y) \frac{dg_\gamma}{d\mu(x)}. \tag{F.10}$$

Finally, the remaining terms in δJ involve only the arbitrary perturbation β:

$$\delta J = \int_0^T \sum_x \mu_t(x) \left[\frac{\partial f_\gamma}{\partial \alpha} \beta + \sum_{y \neq x} \beta_{xy}(u(y) - u(x)) \right] dt. \tag{F.11}$$

Requiring $\delta J = 0$ for all β yields the optimality condition for the control policy.

F.4 Numerical Approximation via Fixed-Point Iteration

To numerically compute the social optimum characterized by the forward-backward optimality system, we employ a Picard iteration scheme analogous to Algo. 2. However, solving the Mean Field Control system presents a unique mathematical and computational challenge when the true parameter γ depends explicitly on the population distribution, i.e., $\gamma_t = \gamma(t, \mu_t)$.

In this scenario, the backward adjoint equation strictly requires the evaluation of the chain rule term $\frac{\partial \gamma}{\partial \mu}$ to internalize the planner's impact on the environment. If we were to embed the nonlinear function $\gamma(\cdot)$ directly into the backward ODE solver, the implicit numerical differentiation of the fixed-point operator would become highly recursive and computationally prohibitive.

For a practical implementation, we linearly decouple the parameter and Jacobian trajectories during the iteration. At each Picard step k, given the previous distribution trajectory $\mu^{(k)}$, we pre-compute both the parameter values and their Jacobian across the entire time horizon:

$$\gamma_t^{(k)} = \gamma(t, \mu_t^{(k)}), \qquad D_t^{(k)} = \frac{\partial \gamma}{\partial \mu}(t, \mu_t^{(k)}). \tag{F.12}$$

These pre-computed trajectories, $(\gamma^{(k)}, D^{(k)})$, are passed as fixed, exogenous inputs into the ODE solver. The backward adjoint equation is then integrated using the fixed trajectory $D_t^{(k)}$ to evaluate the extra MFC term $\frac{\partial f}{\partial \gamma} D_t^{(k)}$.

At a converged fixed point with $\mu^{(k+1)} = \mu^{(k)}$, the precomputed $\gamma^{(k)}$ and $D^{(k)}$ coincide with the quantities evaluated along that fixed point, so the iteration matches the discretized optimality system above for the restricted dynamics.

Crucially, this decoupled architecture elegantly supports the inverse problem of neural parameter calibration. In the inverse setting, we aim to learn the neural network parameters θ that define $\gamma_\theta(t, \mu)$. Mirroring the approach taken for MFGs, we employ a teacher-forcing methodology where we replace the iterative trajectory $\mu_t^{(k)}$ entirely with the *observed* empirical data μ_t^{obs}.

During training, we evaluate the model predictions and its Jacobians explicitly on the observed trajectory:

$$\gamma_t^{obs} = \gamma_\theta(t, \mu_t^{obs}), \qquad D_t^{obs} = \frac{\partial \gamma_\theta}{\partial \mu}(t, \mu_t^{obs}). \tag{F.13}$$

These pre-computed trajectories (γ^{obs}, D^{obs}) are then passed to the ODE solver to generate the predicted mean field μ^{pred}, which is scored against μ^{obs} via the L_2 loss. This approach decouples the neural network training loop from nested, recursive ODE backpropagation and provides a scalable teacher-forced training objective.

F.5 Numerical Experiments & Results

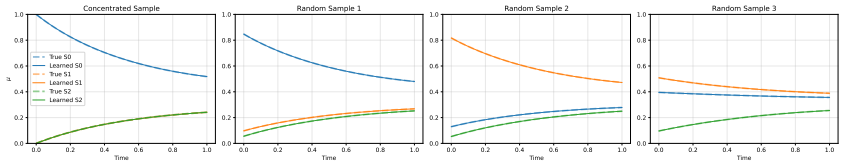
To demonstrate the versatility of our implicit differentiation framework, we extend the LQ model from finding a non-cooperative Nash equilibrium (MFG) to identifying the social optimum formulated by a central planner (Mean Field Control).

While the Nash equilibrium is characterized by selfish agents who only consider their immediate running costs, the social planner minimizes the aggregate cost of the entire population $\int_0^T \sum_x \mu_t(x) f(x, a, \mu) dt$. The fundamental mathematical distinction lies in the backward HJB equation: the social planner's value function must account for the exact derivative of the Lagrangian with respect to the population density.

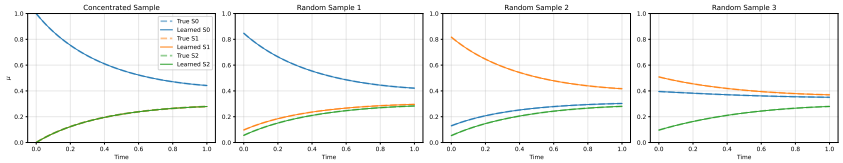
To highlight the physical divergence between the Nash equilibrium and the social optimum, we modify the LQ cost structure in two ways:

1. **Reduced Base Transition Rate:** We lower the base transition rate to $a_{xy} \approx 0.3$. This significantly reduces the natural "free" diffusion of the agents, meaning they remain stationary unless explicitly pushed by the congestion gradients.
2. **Squared Congestion Cost:** We adopt a squared congestion penalty $c\mu(x)^2$, with $c = 2$. This anticipatory "systemic risk" term strongly forces the planner to evacuate crowded states faster than the selfish MFG agents.

Due to the hypersensitivity introduced by the reduced base transition rate, the forward-backward Picard solver naturally oscillates or diverges. To restore mathematical stability, we applied a Picard damping factor of $\lambda = 0.5$ (as detailed in App. A).



(a) Nash Equilibrium (MFG)



(b) Social Optimum (MFC)

Figure 8: Comparison of the mean field evolution μ_t between the non-cooperative Nash equilibrium (top) and the social optimum (bottom). The social planner preemptively diffuses the population out of the concentrated initial state faster to avoid the squared congestion penalty.

G Mean Field Game Calibration of the Cybersecurity Model

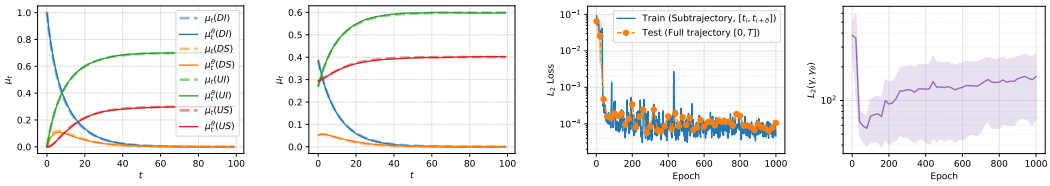


Figure 9: Predicted mean field flow for the Cybersecurity MFG for two random samples from the test set. The two rightmost panes respectively show the train and test error and the L_2 error of the predicted parameters.

H Cycling with Running Costs

To isolate and demonstrate the structural, forward-looking power of the MFG formulation, we evaluate the two models from Sec. 5 under three restricted-capacity testing regimes.

Test 1: Constant Base Rates. We restrict the base transition rates to be entirely constant over time: $\varphi_\theta(t) = \gamma_0$ (a learned but time-independent vector). A standard ODE with constant rates decays toward a stationary distribution and cannot recreate the complex, double-peaked morning and evening commute patterns. The MFG, however, learns a non-trivial terminal cost $w_g(x)$ representing commuters' end-of-day location preferences. The backward HJB equation propagates this terminal incentive backward in time, causing agents to strategically adjust their flows throughout the day. As shown in Table 6, the MFG achieves a 41.7% reduction in Test ID L_2 error compared to the baseline.

Test 2: Linear Base Rates. We allow the base rates to be an affine function of time: $\varphi_\theta(t) = Wt + b_{\text{bias}}$, where $W \in \mathbb{R}^{d(d-1)\times 1}$ and $b_{\text{bias}} \in \mathbb{R}^{d(d-1)}$. This enables the baseline to capture monotonic trends over the day but prevents it from fitting highly non-linear dynamics. Even with this added expressivity, the MFG maintains a 12.0% Test ID improvement (Table 6).

Test 3: Restricted Dynamics with General Cost. We consider a practitioner who relies on a rigid phenomenological model for the baseline dynamics (linear base rates, as in Test 2) and does not wish to alter it. Instead, we augment the MFG with a flexible cost network $\psi_\phi : [0, T] \times \mathcal{P}([d]) \to \mathbb{R}^{d(d-1)}$ (a 2-layer MLP with 64 hidden units and $\tanh$ activations), which replaces the scalar congestion cost

$c \cdot \mu(y)$ in equation 5.1 with a general, potentially negative, cost vector $\gamma_{\text{cost}}(t, \mu_t) := \psi_\phi(t, \mu_t) \in \mathbb{R}^{d(d-1)}$. The running cost becomes:

$$f(x, a, \mu, t) = b \sum_{y \neq x} (a_{xy} - q_{xy}^{\text{base}})^2 - \sum_{y \neq x} [\gamma_{\text{cost}}]_{xy}(t, \mu)\, a_{xy}, \tag{H.1}$$

and the corresponding optimal control is:

$$a_{xy}^* = \max\left(0,\ q_{xy}^{\text{base}} + \frac{[\gamma_{\text{cost}}]_{xy} - (u(y) - u(x))}{2b}\right). \tag{H.2}$$

Note that the sign of the linear cost term is reversed compared to equation 5.1, and the network ψ_ϕ can output negative values, enabling the model to learn both congestion penalties and incentives. As shown in Table 6, this formulation achieves a 19.3% reduction in Test ID L_2 error over the linear baseline, nearly doubling the 12.0% improvement of Test 2.

Table 6: Prediction Errors on Citi Bike Trajectories (Mean $\pm$ SD over 5 seeds). Test ID is the held-out weekdays; Test OOD is the weekends. Absolute L_2 errors are scaled by 10^3.

		Train		Test ID		Test OOD	
Model	**Complexity**	L_2	**Rel.** L_2	L_2	**Rel.** L_2	L_2	**Rel.** L_2
MF Dynamics	Constant	1.55 ± 0.00	0.153 ± 0.000	1.32 ± 0.00	0.054 ± 0.000	3.09 ± 0.02	0.091 ± 0.000
Mean-Field Game	Constant	$\mathbf{0.94 \pm 0.00}$	$\mathbf{0.083 \pm 0.000}$	$\mathbf{0.77 \pm 0.01}$	$\mathbf{0.036 \pm 0.000}$	3.14 ± 0.05	0.101 ± 0.001
MF Dynamics	Linear	1.02 ± 0.00	0.092 ± 0.001	0.83 ± 0.00	0.035 ± 0.000	$\mathbf{3.07 \pm 0.02}$	0.104 ± 0.000
Mean-Field Game	Linear	0.91 ± 0.00	0.085 ± 0.001	0.73 ± 0.00	0.033 ± 0.000	3.18 ± 0.06	0.103 ± 0.001
Mean-Field Game	Linear + General Cost	$\mathbf{0.84 \pm 0.00}$	$\mathbf{0.081 \pm 0.001}$	$\mathbf{0.67 \pm 0.00}$	$\mathbf{0.032 \pm 0.000}$	3.19 ± 0.00	0.105 ± 0.002

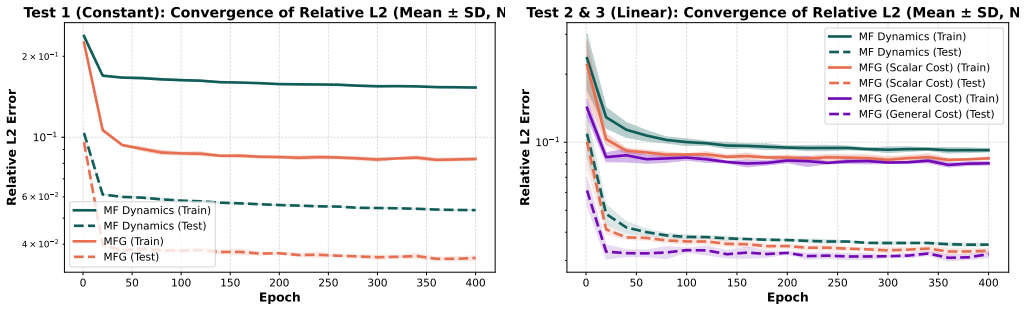


Figure 10: Relative L_2 error convergence. Left (Test 1): Constant Base Rates. Right (Test 2 & 3): Linear Base Rates. The General Cost MFG (purple) clearly achieves the lowest test error, demonstrating the advantage of highly flexible cost networks. Shaded regions denote ± 1 standard deviation over 5 seeds.

Interestingly, both models fail to generalize to the Out-of-Distribution (OOD) test set, which comprises the weekends in March 2024. This occurs due to two distinct domain shifts. First, environmental shocks: the first OOD day (Saturday, March 2nd) experienced rain, causing station usage to stagnate. Second, latent structural shifts: sunny weekend days (e.g., Sunday, March 3rd) exhibit broad, smooth leisure riding rather than the sharp bimodal commuting spikes of weekdays. Because the models were trained exclusively on weekday commute rhythms, they incorrectly predict commute-like surges on weekends. As illustrated in Fig. 13, this yields substantial errors, demonstrating a generalization failure when the underlying user utility shifts. The slightly smaller OOD error of the MF dynamics baseline should therefore be interpreted as a less severe failure under this particular weekend shift, rather than as evidence of better out-of-distribution modelling.

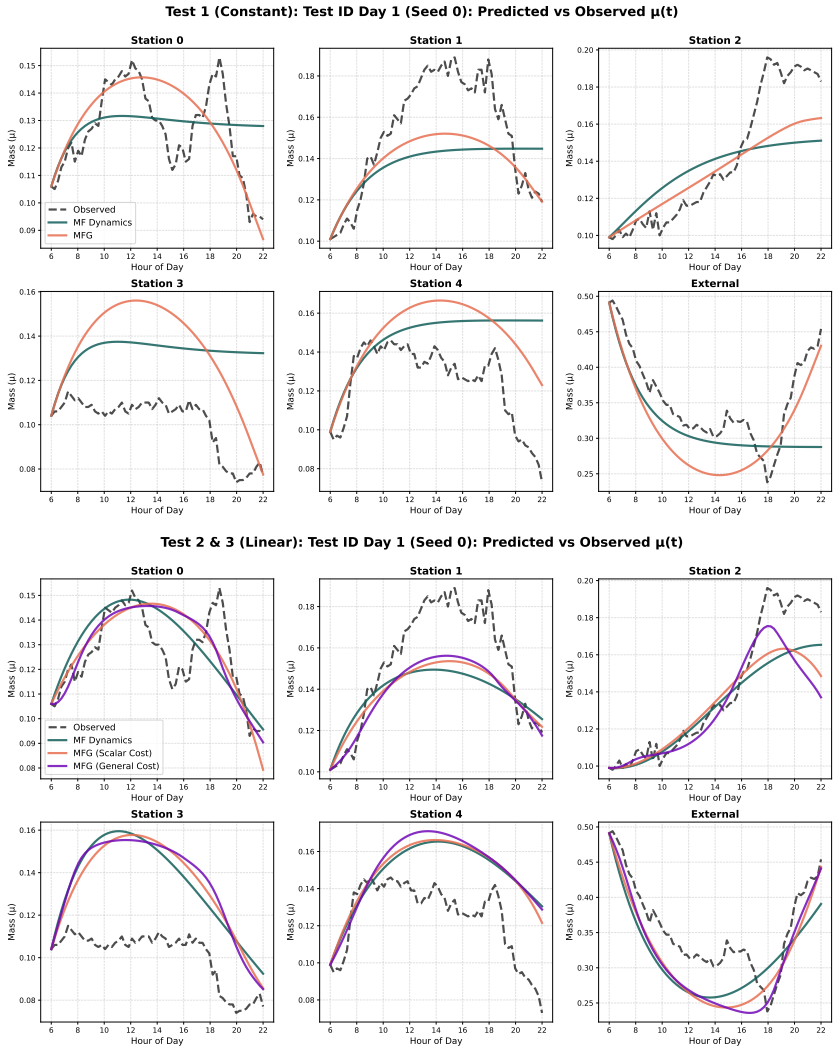


Figure 11: Predicted versus Observed Trajectories on a typical Weekday (Test ID, Seed 0). Top (Test 1): Constant Base Rates. Bottom (Test 2 & 3): Linear Base Rates. Please refer to the text for interpretation.

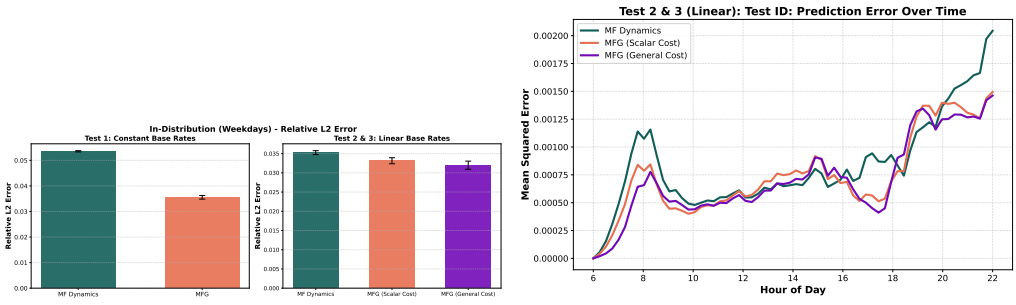


Figure 12: Left: In-Distribution (Test ID) Relative L_2 comparison across all regimes. **Right:** Prediction Error Over Time for Test 2 & 3.

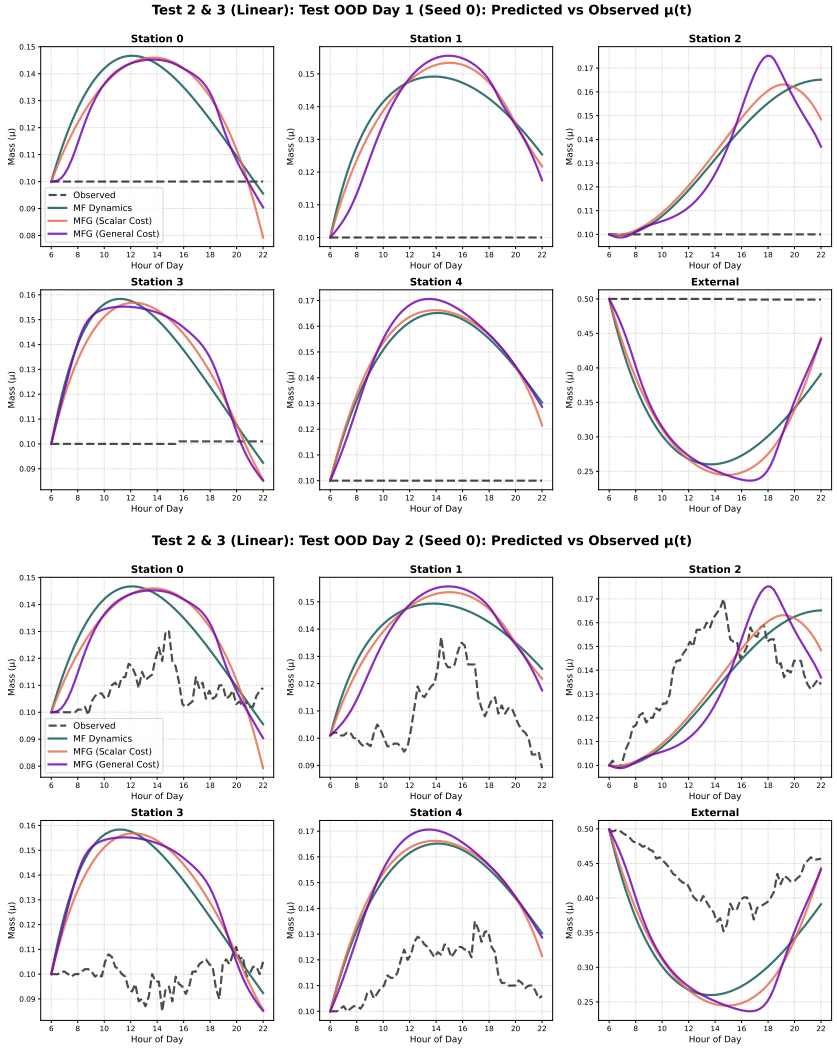


Figure 13: OOD Test Set (Weekends) using Test 2 & 3 models. Top (Day 1): A rainy day caused reduced movement; models predicted weekday commutes. Bottom (Day 2): A sunny Sunday with smooth leisure riding, unlike the sharp commute peaks the models learned.

I CYCLING WITH INTERVENTIONS

We present the mean field mass per station for the two different intervention scenarios of the urban mobility analysis below. Fig. 14 presents the predicted dynamics for intervention scenario 1, where station 1 is closed from 14:45 until 17:30. Fig. 15 presents the predicted dynamics for intervention scenario 2, which involves the closure of station 2 from 07:15 until 09:45.

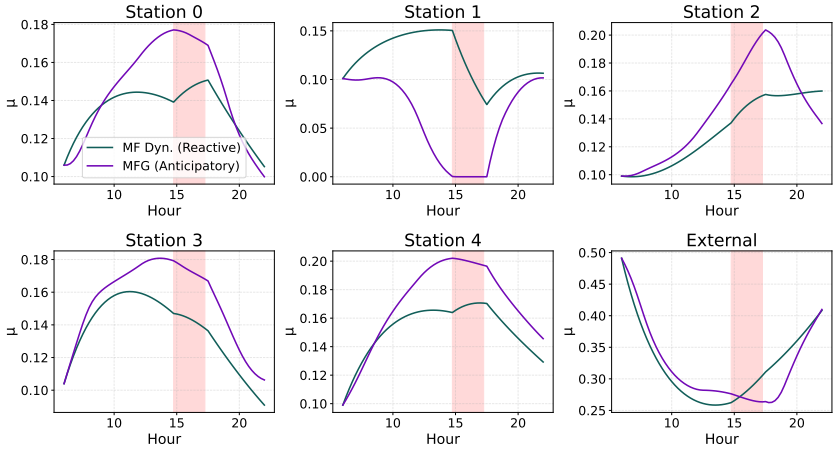


Figure 14: The mean field mass per station for intervention scenario 1.

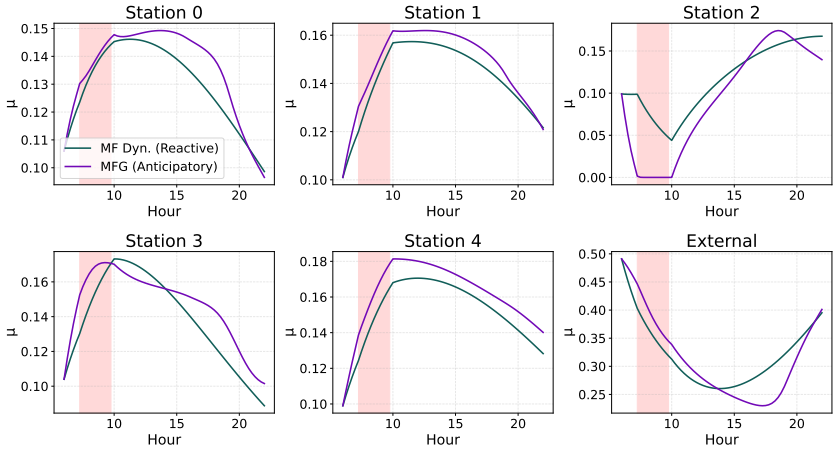


Figure 15: The mean field mass per station for intervention scenario 2.

J EXPERIMENTAL PARAMETERS

J.1 HARDWARE

All experiments are implemented in Python using JAX Bradbury et al. (2018) and are executed on a MacBook Pro equipped with an Apple M1 Pro chip and 16 GB of RAM. Each experiment is performed $n = 5$ times with different random seeds.

J.2 RANDOM SEEDS

For all experiments we use the random seeds $[42, 43, 44, 45, 46]$ to initialize the numpy and JAX pseudorandom number generation processes.

J.3 LINEAR-QUADRATIC

We model the unknown parameter $\gamma \equiv b(t, \mu)$ using a feed-forward neural network. To improve the network's ability to capture highly dynamic and localized parameter shapes, we employ a sub-

trajectory training approach, where gradients are computed over short, randomized time windows (20% of the total horizon). The full set of physical and optimization parameters used for data generation and network training is summarized in Table 7.

Table 7: Physical parameters and neural network hyperparameters for the LQ MFG experiment.

Physical Parameters	**Value**	**Training Hyperparameters**	**Value**
Number of states (d)	3	Learning rate	5×10^{-3}
Time horizon (T)	2.0	Optimizer	Adam
Number of points in time ($N+1$)	100	Batch size	10
Min. parameter ($b_{\min}$)	0.1	Training samples	200
Max. parameter ($b_{\max}$)	3.0	Test samples	20
		Sub-trajectory length	20% of T
		Number of neural network layers	3
		Layer width	64
		Activation	ReLU
		Final activation	Softplus
		Training Epochs	500

J.4 Cybersecurity

We model the unknown parameters $\gamma \equiv (q^D_{rec}, q^U_{rec}, q^D_{inf}, q^U_{inf}, \beta_{DD}, \beta_{DU}, \beta_{UU}, \beta_{UD})$ using a feed-forward neural network. To improve the network's ability to capture highly dynamic and localized parameter shapes, we employ a sub-trajectory training approach, where gradients are computed over short, randomized time windows (20% of the total horizon). The full set of physical and optimization parameters used for data generation and network training is summarized in Table 8.

Table 8: Physical parameters and neural network hyperparameters for the cybersecurity MFG experiment.

Physical Parameters	**Value**	**Training Hyperparameters**	**Value**
Number of states (d)	4	Learning rate	5×10^{-3}
Time horizon (T)	10.0	Optimizer	Adam
Number of points in time ($N+1$)	100	Batch size	10
q^D_{rec}	$[0.0, 0.5]$	Training samples	200
q^U_{rec}	$[0.0, 0.4]$	Test samples	20
q^D_{inf}	$[0.0, 0.4]$	Sub-trajectory length	20% of T
q^U_{inf}	$[0.0, 0.3]$	Number of neural network layers	3
β_{DD}	$[0.0, 0.4]$	Layer width	64
β_{DU}	$[0.0, 0.3]$	Activation	ReLU
β_{UU}	$[0.0, 0.3]$	Final activation	Softplus
β_{UD}	$[0.0, 0.4]$	Training Epochs	1000
λ	0.8		
v_H	0.6		
k_D	0.3		
k_I	0.5		

J.5 Susceptible-Infected-Recovered

We model the unknown parameters $\gamma \equiv (\beta, \gamma, c_I, c_\alpha)$ using a feed-forward neural network. To improve the network's ability to capture highly dynamic and localized parameter shapes, we employ a sub-trajectory training approach, where gradients are computed over short, randomized time windows (20% of the total horizon). The full set of physical and optimization parameters used for data generation and network training is summarized in Table 9. Because the parameters β and γ interact with the players' contact factor $\alpha \in [0, 1]$, they are unbounded in theory. In practice, the parameters often converge to a reasonable domain under Picard Damping.

Table 9: Physical parameters and neural network hyperparameters for the SIR MFG experiment.

Physical Parameters	**Value**	**Training Hyperparameters**	**Value**
Number of states (d)	3	Learning rate	5×10^{-3}
Time horizon (T)	30	Optimizer	Adam
Number of points in time ($N+1$)	30	Batch size	10
γ	$[0, \infty)$	Training samples	200
β	$[0, \infty)$	Test samples	20
c_a	$[0, \infty)$	Sub-trajectory length	20% of T
c_I	$[0, \infty)$	Number of neural network layers	3
Picard Max Iterations	200	Layer width	64
Picard Tolerance	10^{-5}	Activation	ReLU
Picard Damping (α)	0.5	Final activation	Softplus
		Training Epochs	1000

J.6 URBAN MOBILITY

The physical parameters and training hyperparameters for the bike-sharing experiment are presented in Table 10.

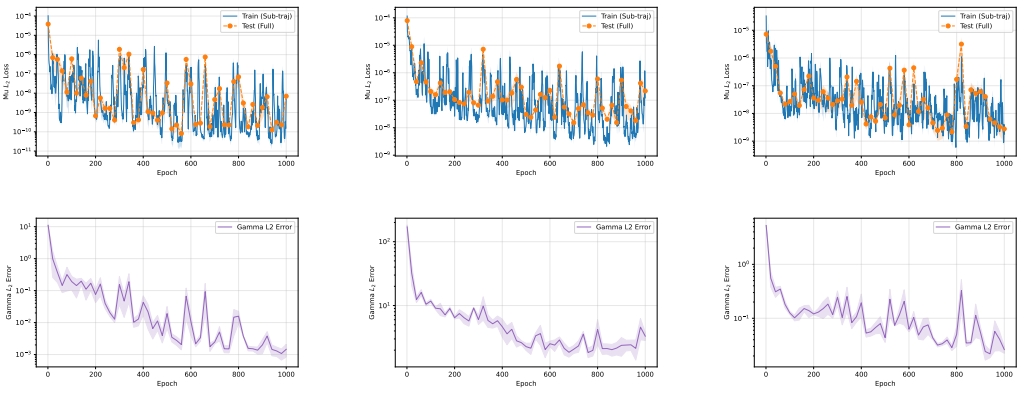
Table 10: Hyperparameters for Urban Mobility Experiments.

Physical parameters	**Value**	**Training Hyperparameters**	**Value**
Number of Stations (d)	6	Learning Rate	5×10^{-3}
Time Horizon (T)	1.0 (16 hours)	Optimizer	Adam
Number of points in time ($N+1$)	64	Epochs	400
Picard Max Iterations	200		
Picard Tolerance	10^{-5}	Initial b_0	1.0
Picard Damping (α)	0.5	Initial c_0	0.5
Base Network (Tests 1–2)	Constant / Affine	Cost Network (Test 3)	2-Layer MLP (64, $\tanh$)

K CONVERGENCE ANALYSIS

Figs. 16-20 present the convergence of the training and testing losses for the mean field μ, as well as the L_2 error of the parameter γ across training epochs. The test loss evaluates the full trajectory prediction, while the training loss reflects the localized sub-trajectory optimization.

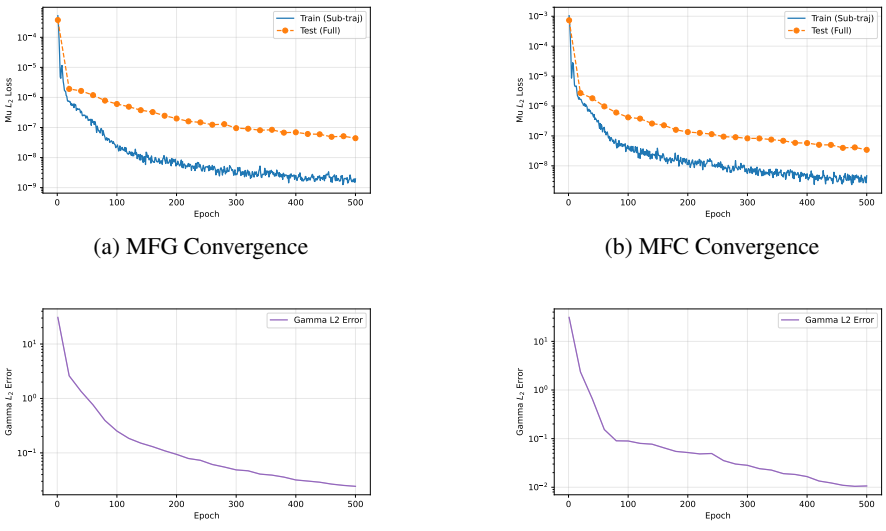
K.1 LINEAR-QUADRATIC



(a) Constant (b) Time-dependent (c) MF-dependent

Figure 16: Convergence of the mean field L_2 loss (top row) and the parameter L_2 error (bottom row) during training for the three test cases without noise. Shaded regions denote the standard error over multiple random seeds.

K.2 MEAN FIELD CONTROL



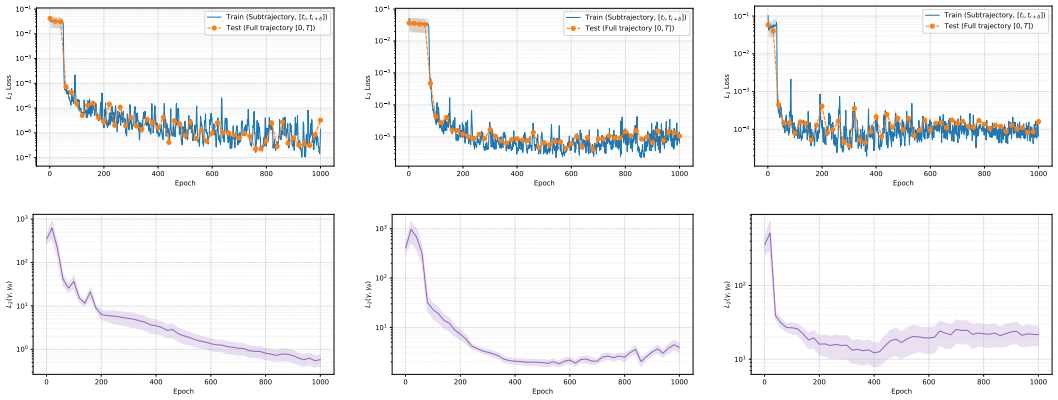
(a) MFG Convergence (b) MFC Convergence

(c) MFG Parameter Error (d) MFC Parameter Error

Figure 17: Training loss for the mean field predictions (top) and the corresponding parameter estimation error (bottom). Despite the modified ODE structure and the inclusion of Picard damping, our framework successfully recovers the underlying parameters for both the MFG and MFC configurations.

K.3 CYBERSECURITY

Fig. 18 presents the mean field L_2 loss and parameter for the cybersecurity example. In the bottom row, we present the mean and standard error of the loss over five seeds.



(a) Constant (b) Time-dependent (c) MF-dependent

Figure 18: Convergence of the mean field L_2 loss (top row) and the parameter L_2 error (bottom row) during training for the three test cases without noise. Shaded regions denote the standard error over multiple random seeds.

K.4 SUSCEPTIBLE-INFECTED-RECOVERED

We provide the L_2 loss associated with the predictions on the CDC influenza dataset in Fig. 19.

When considering the real-world influenza data, we only have access to a proxy for the infected population fraction, namely the influenza test positivity rate $\mu_t(I)$. Therefore, the calculated L_2 only considers the agents at the infected state. We cannot report the parameter L_2 error due to the unavailability of the true parameters γ.

K.5 URBAN MOBILITY.

Lastly, Fig. 20 presents the convergence of the mean field dynamics model and mean field game for the urban mobility example. The figure presents the mean and standard error over the mean as calculated over the five seeds.

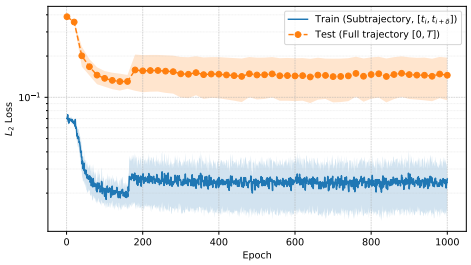


Figure 19: Convergence of the mean field L_2 loss between the predicted infected coordinate $\mu_t^\theta(I)$ and the observed influenza test positivity rate. Shaded regions denote the standard error over multiple random seeds.

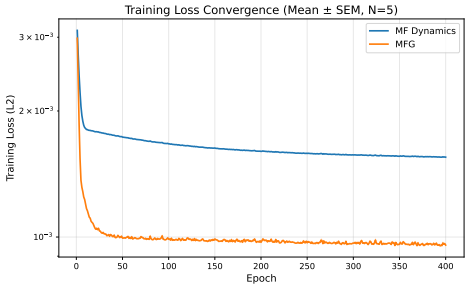


Figure 20: Convergence of the mean field dynamics model and the mean field game for the urban mobility example.