A numerical algorithm for inverse problem from partial boundary measurement arising from mean field game problem

Yat Tin Chow¹**, Samy Wu Fung², Siting Liu³, Levon Nurbekyan¹ and Stanley Osher³

¹ Department of Mathematics, University of California, Riverside, CA, United States of America
² Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, United States of America
³ Department of Mathematics, University of California, Los Angeles, CA, United States of America

E-mail: yattinc@ucr.edu

Received 10 April 2022; revised 13 November 2022
Accepted for publication 24 November 2022
Published 7 December 2022

Abstract
In this work, we consider a novel inverse problem in mean-field games (MFGs). We aim to recover the MFG model parameters that govern the underlying interactions among the population based on a limited set of noisy partial observations of the population dynamics under the limited aperture. Due to its severe ill-posedness, obtaining a good quality reconstruction is very difficult. Nonetheless, it is vital to recover the model parameters stably and efficiently in order to uncover the underlying causes for population dynamics for practical needs. Our work focuses on the simultaneous recovery of running cost and interaction energy in the MFG equations from a finite number of boundary measurements of population profile and boundary movement. To achieve this goal, we formalize the inverse problem as a constrained optimization problem of a least squares residual functional under suitable norms. We then develop a fast and robust operator splitting algorithm to solve the optimization using techniques including harmonic extensions, three-operator splitting scheme, and primal–dual hybrid gradient method. Numerical experiments illustrate the effectiveness and robustness of the algorithm.

Keywords: inverse problem, mean-field games, optimization, primal–dual, coefficient recovery

(Some figures may appear in colour only in the online journal)
1. Introduction

The basis for the mean-field game (MFG) framework is the concept of Nash equilibrium, where agents cannot unilaterally improve their objectives. Under suitable regularity assumptions, a common MFG model reduces to the following system of partial differential equations (PDEs):

\[
\begin{align*}
-\partial_t \phi(x,t) - \nu \Delta \phi(x,t) + H(x, \nabla_x \phi(x,t)) &= F(x, \rho(\cdot, t)), \quad \text{in } \Omega' \times (0,1), \\
\partial_t \rho(x,t) - \nu \Delta \rho(x,t) - \nabla_x \cdot (\rho(x,t) \nabla_x H(x, \nabla_x \phi(x,t))) &= 0, \quad \text{in } \Omega' \times (0,1), \\
\rho(x,0) = \rho_0(x), \quad \phi(x,1) = g(x), \quad \text{in } \Omega'.
\end{align*}
\]

Here, \( \rho(\cdot, t), \ t \in [0,1] \) represents the population distribution over the state space \( \Omega' \) at time \( t \) satisfying a Fokker-Planck equation, and \( \phi(x,t) \) is the value function of each player that satisfies a Hamilton–Jacobi equation and governs optimal actions of players. The Hamiltonian, \( H \), is the Legendre transform of the Lagrangian, \( L \), representing the running cost for each agent. Furthermore, \( F \) represents an interaction between the agents and the population. Typical choices for \( H, L, F \) in crowd motion applications are

\[
H(x,p) = \frac{1}{2} \kappa(x) |p|^2, \quad L(x,v) = \frac{1}{2} \kappa(x) |v|^2, \quad F(x,\rho) = \int_{\Omega} K(x,y) \rho(y) dy. \quad (1.2)
\]

Introduced in [17–19] and [13, 14], MFG is an actively growing field significantly advancing the understanding of social cooperation and economics [1, 5, 10], biological systems [30], election dynamics [31], population games [20], robotic control [25], machine learning [22, 28], dynamics of multiple populations [7]. Recently, they are utilized to understand pandemic modeling and control such as COVID-19 [21].

With the significant descriptive power of MFGs, it is vital to consider inverse problems arising in MFGs. We aim to reconstruct MFG parameters for a class of nonlocal problems, including the geometry of the underlying space and the interactions between large crowds, based on partial population observations. More specifically, we are interested in the following problem.

**Problem 1.1.** Given a part of the solution to an MFG system (1.1) and (1.2), for instance,

\[
\left. \begin{array}{c}
\rho(x,s) , -\rho(x,s) \nabla_p H(x, \nabla_x \phi(x,t)) \\
\end{array} \right|_{\partial \Omega \times (0,T)},
\]

for finitely many examples of \( \rho_0 \) and terminal cost \( g \), can we numerically recover the speed field \( \kappa(x) \) and the interaction kernel \( K(x,y) \) from observations?

Such a model-recovery algorithm can help understand the underlying population dynamics in numerous problems, such as migration flow or contagious rate of COVID-19. We further envision applications to include rescue and exploration team management, policymaking, diplomacy, election modeling, catastrophe management, and evacuation planning.

Note that \( m(x,s) = -\rho(x,s) \nabla_p H(x, \nabla_x \phi(x,s)) \) represents the flux of the agents through the state \( x \) at time \( s \) as a result of optimal actions. The interpretation of the flux is straightforward for crowd-dynamics models and can be measured by counting people crossing checkpoints or parts of the border. For such models, the value function \( \phi(x,s) \) could represent the travel cost for a traveller who is at location \( x \) at time \( s \). Hence, one could also consider an inverse problem where one observes the value function, instead of the flux, by looking at travel companies’ expenses or consumer ticket prices (discounted for the companies’ profit margins).
For economic and finance models [1, 5, 11] the state variables typically represent asset (wealth, income, inventory) levels instead of a physical location. Hence, the value function represents maximal utility for agents with a given asset level, and the flux represents the total amount of transactions performed by them. Interestingly, in economic models one often has implicit mean-field interactions reflected in market-clearing type conditions instead of an explicit interaction functional $F(x, \rho)$. Hence, a related inverse problem is to find an appropriate market-clearing condition or tune its parameters for a given economy. This manuscript addresses explicit models with flux data leaving the implicit ones with other data types for future work.

1.1. Our contributions

We propose a new MFG inverse problem with non-invasive partial boundary measurements. Based on insights from [23, 24, 26] we postulate a feature expansion representation for the interaction kernel $K$ and formulate the forward problem (1.1) as a convex–concave saddle point problem. Furthermore, we design a three-operator splitting scheme [8] for the resulting inverse problem with a saddle-point constraint. The algorithm reduces to a forward–backward splitting for the parameter updates, a primal-dual hybrid gradient for the forward saddle point problem update, and a proximal-point algorithm for the adjoint problem update. Intriguingly, our algorithm applies to inverse problems whose forward problem has a saddle point structure beyond MFG.

1.2. Related work

Despite of the large body of work on theory, numerical methods, and applications [2], inverse problems arisen from MFG is still quite an unexplored terrain. To the best of our knowledge, only [4, 9, 16] study such problems. The work in [9] is the closest to our objective but considers the case with a full space-time measurement of data in the sampling domain. However, most inverse problems in practice only have partial boundary measurements available, either obtained via non-invasive measurement methods or because of the limited access to the sampling domain. Compared with the case with full space-time measurement in the domain, inverse problems with only partial boundary measurements are generally known to be more severely ill-posed. In this work, we focus on the recovery problem with only boundary measurements coming from several measurement events.

The rest of the paper is organized as follows. In section 2, we introduce an abstract inverse problem with a saddle point constraint and a generic algorithm to solve it. In section 3, we present the inverse MFG formulation. Next, in section 4 we discuss the implementation of the algorithm in section 2 for the inverse MFG in section 3. Section 5 contains three numerical examples to demonstrate the robustness and effectiveness of our algorithm. Finally, section 6 contains a discussion and concluding remarks.

2. An inverse problem with a saddle point forward model

In this section, we formulate an abstract inverse problem with a saddle point forward model. We discuss suitable Karush–Kuhn–Tucker (KKT) conditions and a generic algorithm to solve such inverse problems.
2.1. A forward saddle point problem

Consider a saddle point problem

\[
\min_{x \in X} \max_{y \in Y} \mathcal{F}(u, x, y, c),
\]

where \( \mathcal{F} : U \times X \times Y \times D \rightarrow \mathbb{R} \cup \{ \pm \infty \} \) a smooth functional such that \((x, y) \mapsto \mathcal{F}(u, x, y, c)\) is strongly convex–concave. Here, \(x\) is the primal variable, and \(y\) is the dual variable in the forward problem. Commonly, \(y\) is either used to handle constraints in the forward problem or linearize nonlinear components via some splitting scheme. The variable \(c\) represents model parameters associated with the functional \(\mathcal{F}\), while \(u\) represents boundary and initial-terminal conditions. Given model parameters \(c\), we define a boundary measurement map \(\Lambda_c\) as follows:

\[
\Lambda_c : \mathcal{U} \rightarrow \Pi_B X \times \Pi_B Y
\]

\[
u \mapsto (\Pi_{B, x}(x), \Pi_{B, y}(y)) \text{ where } (x, y) \in \arg\min_x \arg\max_y \mathcal{F}(u, x, y, c),
\]

where \(\Pi_{B, x}, \Pi_{B, y}\) denote a projection operator that represent the partial boundary measurements of \(x, y\) available. We note that \(\nu\) corresponds to boundary conditions of the forward problem, whereas \(B\) is the subset of the domain where the partial measurements are collected.

2.2. The inverse problem and a generic algorithm

Assume that

\[
(\tilde{r}_{B, i}, \tilde{\delta}_{B, i}) \approx \Lambda_c(u_i) = ([\Pi_{B, x}(x_i)], [\Pi_{B, y}(y_i)](u_i))
\]

are noisy measurements for a given \(\{u_i\}_{i=1}^N \in \mathcal{U}\). Our goal is to recover \(c \in \mathcal{D}\). We formulate this problem as a constrained optimization problem

\[
\inf_{\{x, y\} \in \mathcal{C}} \left\{ \sum_{i=1}^N \frac{1}{2} \|\Pi_{B, x}(x_i) - \tilde{r}_{B, i}\|^2 + \sum_{i=1}^N \frac{1}{2} \|\Pi_{B, y}(y_i) - \tilde{\delta}_{B, i}\|^2 + R(c) : \right. \\
\left. \times (x_i, y_i) \in \arg\min_x \arg\max_y \mathcal{F}(u_i, x, y, c) \right\},
\]

where \(R\) is a suitable regularizer and \(\| \cdot \|\) are suitable choices of (semi)-norms. Introducing Lagrange multipliers (dual variables) \((\lambda_{x_i}, \lambda_{y_i})\), \((2.2)\) reduces to

\[
\inf_{\{x, y\} \in \mathcal{C}} \sup_{(\lambda_{x_i}, \lambda_{y_i})} \left\{ \sum_{i=1}^N \frac{1}{2} \|\Pi_{B, x}(x_i) - \tilde{r}_{B, i}\|^2 + \sum_{i=1}^N \frac{1}{2} \|\Pi_{B, y}(y_i) - \tilde{\delta}_{B, i}\|^2 + R(c) \\
+ \sum_{i=1}^N \langle \partial_{x_i} \mathcal{F}(u_i, x, y, c), \lambda_{x_i} \rangle - \sum_{i=1}^N \langle \partial_{y_i} \mathcal{F}(u_i, x, y, c), \lambda_{y_i} \rangle \right\}.
\]

Thus, the KKT condition for \((2.2)\) and \((2.3)\) are as follows:

\[
\begin{align*}
\Pi_{B, x}^* [\Pi_{B, x}(x_i) - \tilde{r}_{B, i}] + \partial_{x_i}^2 \mathcal{F}(u_i, x, y, c) \lambda_{x_i} &= 0, \\
\Pi_{B, y}^* [\Pi_{B, y}(y_i) - \tilde{\delta}_{B, i}] - \partial_{y_i}^2 \mathcal{F}(u_i, x, y, c) \lambda_{y_i} &= 0, \\
\partial_i R(c) + \sum_{i=1}^N \langle \partial_{x_i} \mathcal{F}(u_i, x, y, c), \lambda_{x_i} \rangle - \sum_{i=1}^N \langle \partial_{y_i} \mathcal{F}(u_i, x, y, c), \lambda_{y_i} \rangle &= 0, \\
\partial_{x_i} \mathcal{F}(u_i, x, y, c) &= 0, \\
-\partial_{y_i} \mathcal{F}(u_i, x, y, c) &= 0,
\end{align*}
\]
for \( i = 1, \ldots, N \). Here, \( \Pi_{B,x}^*, \Pi_{B,y}^* \) are the adjoints of \( \Pi_{B,x}, \Pi_{B,y} \), respectively.

Finally, we formulate these KKT conditions as an inclusion problem

\[
0 \in A(c, (x, y), (\lambda_x, \lambda_y)) + C(c, (x, y), (\lambda_x, \lambda_y)) ,
\]

where

\[
A(c, (x, y), (\lambda_x, \lambda_y)) = \begin{pmatrix}
\partial_x R(c) \\
0
\end{pmatrix},
\]

and

\[
C(c, (x, y), (\lambda_x, \lambda_y)) = \begin{pmatrix}
\sum_{i=1}^N \langle \partial_i \partial_x F(u_i, x_i, y_i, c), \lambda_{x_i} \rangle - \sum_{i=1}^N \langle \partial_i \partial_y F(u_i, x_i, y_i, c), \lambda_{y_i} \rangle \\
\langle \partial_i^2 F(u_i, x_i, y_i, c), -\partial_y F(u_i, x_i, y_i, c) \rangle \\
\langle \partial_i^2 F(u_i, x_i, y_i, c), \lambda_{x_i} \rangle + \Pi_{B,x}^*, [\Pi_{B,x}(x_i) - \tilde{s}_{B,x,i}]
\end{pmatrix}.
\]

Note that \( A \) is monotone but \( C \) is not known to be monotone in general.

2.3. A generic algorithm

Here, we outline an iterative algorithm for solving (2.3). At \((n + 1)\)th iteration, we first update the adjoint variables \( \{(\lambda_{x_i}, \lambda_{y_i})\}_{i=1}^N \) using the Chambolle-Pock method [6]; then we update \( c \) for the inverse problem by taking a proximal gradient step; next we use the Chambolle-Pock method again to compute forward problems \( \{(x_i, y_i)\}_{i=1}^N \). Summarizing, a high level description of the \((n + 1)\)th iteration is as follows:

\[
\begin{align*}
\lambda_{x_i}^{n+1} \quad & = (I + \alpha_c \partial_x R)^{-1} [c^n - \alpha_c \sum_{i=1}^N \langle \partial_i \partial_x F(u_i, x_i^n, y_i^n, c^n), \lambda_{x_i}^{n+1} \rangle] \\
\lambda_{y_i}^{n+1} \quad & = 2 \lambda_{x_i}^{n+1, \text{temp}} - \lambda_{x_i}^{n, \text{temp}} \\
\lambda_{x_i}^{n+1, \text{temp}} \quad & = 1 - \alpha_{\lambda_x} \partial_x F(u_i, x_i^n, y_i^n, c^n) \alpha_{\lambda_x} \Pi_{B,x}^*, [\Pi_{B,x}(x_i^n) - \tilde{s}_{B,x,i}] \\
\lambda_{y_i}^{n+1, \text{temp}} \quad & = 1 - \alpha_{\lambda_y} \partial_y F(u_i, x_i^n, y_i^n, c^n) \alpha_{\lambda_y} \Pi_{B,y}^*, [\Pi_{B,y}(y_i^n) - \tilde{s}_{B,y,i}]
\end{align*}
\]

where \( (\alpha_{\lambda_x}, \alpha_{\lambda_y}, \alpha_c, \alpha_n, \alpha_{\nu_i}) \) are the corresponding time steps.

In what follows, we specify the MFG inverse problem and the implementation of the algorithm above for it.

3. An inverse MFG problem

Here, we explain the saddle point problem formulation of nonlocal MFG [23, 24, 26] and formulate the inverse MFG problem of our interest.
3.1. Saddle point formulation of MFG via feature-space expansions

Consider the following MFG system with nonlocal couplings:

\[
\begin{align*}
\phi_t(x,t) - \nu \Delta \phi(x,t) + \frac{\partial}{\partial x} \left[ \frac{1}{2} \| \nabla \phi(x,t) \|^2 \right] &= \int_{\Omega'} K(x,y) \rho(y,t) \, dy \quad \text{in } \Omega' \times (0,1), \\
\rho_t(x,t) - \nu \Delta \rho(x,t) - \nabla \cdot (\kappa(x) \rho(x,t) \nabla \phi(x,t)) &= 0 \quad \text{in } \Omega' \times (0,1), \\
\kappa(x) \rho(x,t) \nabla \phi(x,t) \cdot n &= 0 \quad \text{on } \partial \Omega' \times (0,1), \\
\rho(x,0) &= \rho_0(x), \quad \phi(x,1) = g(x) \quad \text{in } \Omega'.
\end{align*}
\] (3.1)

We assume that \( K \) is positive definite and translation invariant, which yields that the mean-field interaction satisfies the Lasry-Lions monotonicity condition [19] and agents are crowd averse. Moreover, (3.1) admits a saddle point formulation

\[
\inf_{\phi} \sup_{\rho, m} \left\{ - \int_{\Omega'} \phi(x,0) \rho_0(x) \, dx - \int_{\Omega'} \int_0^1 \left( \rho \phi_t + \nu \rho \Delta \phi + m \cdot \nabla \phi \right) \, dt \, dx \\
- \frac{1}{2} \int_{\Omega'} \int_0^1 \frac{1}{2 \kappa(x)} \left\| m \right\|^2 \, dt \, dx - \frac{1}{2} \int_{\Omega' \times \Omega'} K(x,y) \rho(x,t) \rho(y,t) \, dx \, dy \\
- \chi_{\rho \geq 0} + \chi_{\phi(x,1) = g(x)} \right\}.
\] (3.2)

Here, \( \chi_Z(z) \) is the indicator function over the set \( Z \) defined by

\[
\chi_Z(z) = \begin{cases} 
0, & \text{if } z \in Z \\
\infty, & \text{otherwise}.
\end{cases}
\]

Modeling the interaction term

\[
\frac{1}{2} \int_{\Omega' \times \Omega'} K(x,y) \rho(x,t) \rho(y,t) \, dx \, dy
\]

directly is costly for both forward model and the inverse problem. Moreover, based on the works from [3, 23, 24, 26], we model and approximate this term using feature-space expansions. More specifically, based on Bochner’s theorem [27], we postulate that

\[
K(x,y) = \sum_{k=1}^r \mu_k^2 \cos(\omega_k \cdot (x - y))
\]

for some \( \{\omega_k\} \subset \mathbb{R}^d \), and \( \{\mu_k^2\} \subset \mathbb{R}_+ \). Since the coefficients for each k-index pair of cosine and sine functions are the same, we denote

\[
\mu = (\mu_{1,1}, \mu_{1,2}, \mu_{2,1}, \ldots, \mu_{r,1}, \mu_{r,2}) \quad \omega = (\omega_{1,1}, \omega_{1,2}, \omega_{2,1}, \ldots, \omega_{r,1}, \omega_{r,2}),
\]

and introduce the set

\[
C_{\text{odd} = \text{even}} = \left\{ (x_{1,1}, x_{1,2}, x_{2,1}, \ldots, x_{r,1}, x_{r,2}) : x_{1,1} = x_{1,2} \right\}.
\]

With this notation in hand, we adopt the following basis functions formulation \( \zeta \)

\[
\zeta(x; \mu, \omega) = \left( \mu_{1,1} \cos(\omega_{1,1} \cdot x), \mu_{1,2} \sin(\omega_{1,1} \cdot x), \ldots, \mu_{r,1} \cos(\omega_{r,1} \cdot x), \mu_{r,2} \sin(\omega_{r,2} \cdot x) \right).
\]
and obtain
\[ K(x, y) = \zeta(x; \mu, \omega) \cdot \zeta(y; \mu, \omega), \quad \mu, \omega \in C_{\text{odd-even}}. \]

Using this representation, we obtain
\[
\frac{1}{2} \int_{\Omega \times \Omega} K(x, y) \rho(x, t) \rho(y, t) dt = \frac{1}{2} \left\| \int_{\Omega} \zeta(x; \mu, \omega) \rho(x, t) dx \right\|^2
\]
\[
= \sup_a \left\{ a(t) \cdot \int_{\Omega} \zeta(x; \mu, \omega) \rho(x, t) dx - \frac{1}{2} \int_0^1 \|a(t)\|^2 dt \right\},
\]
where \( a(t) = (a_{1,1}(t), a_{1,2}(t), \ldots, a_{r,1}(t), a_{r,2}(t)) \) are auxiliary dual variables. The last equality is a result from [26]. Hence, (3.2) transforms to
\[
\inf_{\phi, a, \rho, \mu} \sup_{\rho, m, \alpha} \left\{ \frac{1}{2} \int_0^1 \|a(t)\|^2 dt - \int_{\Omega} \phi(x, 0) \rho_0(x) dx - \int_{\Omega} \int_0^1 (\rho \phi_t + \nabla \phi \cdot m - m \cdot \nabla \phi) dx dt \right. \\
- \int_{\Omega} \int_0^1 \left( \frac{1}{2} \frac{m^2}{\rho} + \rho a(t) \cdot \zeta(x; \mu, \omega) \right) dx dt - \mu \phi_{(1,0)} + \chi_{\phi_{(1,0)}(\mu)} \right\}
\]
\[
:= \inf_{\phi, a, \rho, \mu} \sup_{\rho, m, \alpha} \left\{ - \mathcal{L}((\rho_0, g), (\rho, m), (\phi, a), (\kappa, \mu)) \right\}.
\]
(3.3)

For more details on the representations of nonlocal MFG interactions via a basis and computational methods, see [3, 23, 24, 26]. We also attach an example algorithm 3 for calculating the nonlocal MFG problem in the appendix.

3.2. An inverse mean-field game problem

Denoting by
\[
\begin{align*}
\mu &= (\rho, g), \quad x = (\rho, m), \quad y = (\phi, a), \quad c = (\kappa, \mu), \\
F((\rho_0, g), (\rho, m), (\phi, a), (\kappa, \mu)) &= \mathcal{L}((\rho_0, g), (\rho, m), (\phi, a), (\kappa, \mu)),
\end{align*}
\]
we place the MFG forward model in the abstract framework (2.1). Next, we assume that \( \Omega \subset \Omega' \) and \( \kappa(x) \) is known in the domain \( \Omega' \backslash \Omega \). We refer to \( \Omega \) and \( \Omega' \) as sampling and computational domains, respectively. An example is shown in figure 1, where the \( \Omega' \) is the large square domain, while \( \Omega \) is the inner square with its boundary highlighted in red.

Next, we take
\[
\Pi_{\mathcal{B}_{(\rho, m)}}(\rho, m) := (\rho, m \cdot n)|_{\partial \Omega^+}, \quad \Pi_{\mathcal{B}_{(\phi, a)}}(a, \phi) := (0, 0),
\]
for the partial boundary measurement along the boundary \( \partial \Omega \). Here, \( \partial \Omega^+ \) means that the normal vector \( n \) is pointing outward. Measuring the density and flux through \( \partial \Omega \) is reasonable based on physical meaning of the variables. We cannot measure \( a \) directly because it is a non-physical auxiliary variable introduced specifically for an efficient representation of nonlocal interactions.

We assume that the ground truth parameters \( (\kappa, \mu) \) represent a disturbance of background parameters \( (\kappa_0, \mu_0) \). Therefore, given an additional parameter \( \varepsilon \geq 0 \), we would also like to have a regularization term in the form of \( R_{\beta} = \chi_{\beta \geq \varepsilon}(\beta) \). We also write \( R(\kappa, \mu) = R_1(\kappa, \mu) + R_2(\kappa, \mu) \), where
Figure 1. Denote $\rho$ as the solution to the MFG system. From left to right, the pictures display the density distribution $\rho$ at time $t = 0.1, 0.5, 0.9$. The solid red line represents the boundary of domain $\Omega$. In this mean-filed game, the density travels from the right towards the left, crossing the boundary $\partial \Omega$ twice.

\[ R_1(\kappa, \mu) = \tilde{R}_1(\kappa) + \tilde{R}_2(\mu) := \gamma_\kappa \| \kappa - \kappa_0 \|_{L^1} + \gamma_\mu \| \mu - \mu_0 \|_{L^1}, \]

\[ R_2(\kappa, \mu) := \chi_{\kappa > \varepsilon_1}(\kappa) + \chi_{\mu^2 \leq \varepsilon_2}(\mu). \]

It is also possible to have other choices of regularization for $(\kappa, \mu)$, such as $TV, H^1$, Wavelet norms.

Now, we can formulate the inverse MFG as follows:

\[
\inf_{\{(\rho, m), (\phi, a)\}} \left\{ \sum_{i=1}^N \frac{1}{2} \| \Pi_{\beta}(\rho_i, m_i) - \tilde{r}_{\beta_i} \|^2 + R(\kappa, \mu) : (\rho, m), (\phi, a) \in \arg\min_{\rho, m} \arg\max_{\phi, a} F((\rho_0, \rho_i, g_i), (\rho, m), (\phi, a), (\kappa, \mu)) \right\}.
\]

(3.4)

4. The algorithm

We propose an inverse algorithm adapted from the three-operator splitting method [8], which has also been shown to predict Nash equilibria in traffic flows [12]. We also discuss stabilizing techniques that are essential in practice.

4.1. The three-operator splitting scheme

Denoting by $\lambda_{(\rho, m)} := (\lambda_{\rho}, \lambda_{m})$ and $\lambda_{(a, \phi)} := (\lambda_{a}, \lambda_{\phi})$ and applying the framework in section 2 to (3.4) we obtain the following inclusion formulation of the inverse MFG problem:

\[
0 \in A(\kappa, \mu) + B(\kappa, \mu) + C(\kappa, \mu, (\rho, m), (a, \phi), (\lambda_{(\rho, m)}, \lambda_{(a, \phi)})),
\]

(4.1)

where

\[
A(\kappa, \mu) = \left( \begin{array}{c} \partial R_1(\kappa, \mu) \\ (0, 0) \\ (0, 0) \end{array} \right), \quad B(\kappa, \mu) = \left( \begin{array}{c} \partial R_2(\kappa, \mu) \\ (0, 0) \\ (0, 0) \end{array} \right).
\]
and
\[
C \left( (\kappa, \mu), ((\rho, m), (\alpha, \phi)), (\lambda_{(\rho, m)}, \lambda_{(\alpha, \phi)}) \right) = \begin{pmatrix}
\sum_{i=1}^{N} \langle \partial_{(\kappa, \mu)} \partial_{(\rho, m)} \mathcal{L}, \lambda_{(\rho, m)} \rangle \\
- \sum_{i=1}^{N} \langle \partial_{(\alpha, \phi)} \partial_{(\rho, m)} \mathcal{L}, \lambda_{(\rho, m)} \rangle \\
\langle \partial_{(\rho, m)} \mathcal{L}, \lambda_{(\rho, m)} \rangle - \langle \partial_{(\alpha, \phi)} \mathcal{L}, \lambda_{(\rho, m)} \rangle \\
\langle \partial_{(\rho, m)}, \lambda_{(\rho, m)} \rangle + \Pi_{B, (\rho, m)}^* [\Pi_{B, (\rho, m)} (\rho, m) - \tilde{r}_{B, i}], \\
- \partial_{(\alpha, \phi)} \lambda_{(\rho, m)} \rangle \mathcal{L}\lambda_{(\alpha, \phi)} \rangle
\end{pmatrix}.
\]

The three-operator splitting scheme in [8] applies to optimization problems of the form
\[
\text{find } z \in \mathcal{H} \text{ such that } 0 \in A z + B z + C z,
\]
where $A, B, C$ are maximal monotone operators defined on a Hilbert space $\mathcal{H}$, and $C$ is cocoercive. Denote by $I_\mathcal{H}$ the identity map in $\mathcal{H}$, and $J_S := (I + S)^{-1}$ the resolvent of a monotone operator $S$. The splitting scheme for solving (4.2) can be summarized as follows
\[
z^{k+1} := (1 - \lambda_k) z^k + \lambda_k T z^k,
\]
where $\gamma$ is a scalar. If an operator $S$ is of the sub-differential forms; that is, $S = \partial f_S$ for some functional $f_S$, the resolvent $J_S$ reduces to the proximal map $x \mapsto \arg \min f_S(y) + \frac{1}{2\lambda} ||x - y||^2$.

Overall, the algorithm for (4.1) follows three components of the generic framework in section 2.3, upon some modification. In what follows, we discuss each component separately.

4.1.1 Update of the adjoint problem. Firstly, we choose $\| \cdot \|_{(\rho, m)}$ and $\| \cdot \|_{(\alpha, \phi)}$ with $L^2([0, T], H^{-1/2}(\partial \Omega)) \times L^2([0, T], H^{1/2}(\partial \Omega))$ semi-norm, and $L^2 \times L^2_{\partial \Omega}$ norms, respectively. Here the $H^1(\partial \Omega)$ and $H^{-1/2}(\partial \Omega)$ semi-norm are taken as follows:
\[
|v|^2_{H^1(\partial \Omega)} := \min_{v_0 \in H^1(\Omega), v_0 = \text{vom } \partial \Omega} |v_0|^2_{H^1(\Omega)},\\
|v|^2_{H^{-1/2}(\partial \Omega)} := \min_{v_0 \in H^1(\Omega), \partial v_0 = \text{vom } \partial \Omega} |v_0|^2_{H^1(\Omega)},
\]
where the right hand side denotes the standard $H^1(\Omega)$ semi-norm.

Assuming appropriate regularity of $(\rho, m)$, we recall that the operator $\Pi_{B, (\rho, m)}$ is the restriction/trace operator onto the appropriate Sobolev space on the boundary $L^2 ([0, T], H^{-1/2}(\partial \Omega)) \times L^2 ([0, T], H^{1/2}(\partial \Omega))$
\[
\Pi_{B, (\rho, m)} (\rho, m) := (\rho, m \cdot n) |[0, T] \times \partial \Omega^+.
\]

With the aforementioned choice of the semi-norms, we naturally have the (formal) adjoint of $\Pi_{B, (\rho, m)}$, $\Pi_{B, (\rho, m)}^*$, as the Dirichlet and Neumann harmonic extension operators by definition; that is,
\[ (\eta_i, \nabla \xi_i) := \Pi_{B,(\rho,m)}^* [\Pi_{B,(\rho,m)} (\rho^0_i, m^0_i) - \tilde{r}_B], \]

where \((\eta_i, \xi_i)\) satisfy

\[
\begin{cases}
\Delta \eta_i^{n+1} = 0 \text{ in } \Omega \\
\eta_i^{n+1} = \rho_i^0 - pr_1 \tilde{r}_B, \text{on } \partial \Omega^+ \\
\Delta \xi_i^{n+1} = 0 \text{ in } \Omega' \setminus \Omega \\
\xi_i^{n+1} = \rho_i^0 - pr_1 \tilde{r}_B, \text{on } \partial \Omega^- \\
\eta_i^{n+1} = 0 \text{ on } \partial \Omega' 
\end{cases}
\]

\[
\begin{cases}
\Delta \epsilon_i^{n+1} = 0 \text{ in } \Omega \\
\partial \epsilon_i^{n+1} = m_i^0 \cdot n + pr_2 \tilde{r}_B, \text{on } \partial \Omega^+ \\
\Delta \epsilon_i^{n+1} = 0 \text{ in } \Omega' \setminus \Omega \\
\partial \epsilon_i^{n+1} = m_i^0 \cdot n + pr_2 \tilde{r}_B, \text{on } \partial \Omega^- \\
\partial \epsilon_i^{n+1} = 0 \text{ on } \partial \Omega'.
\end{cases}
\]

(4.4)

Here we use \(pr_1, pr_2\) to denote the projection from the noisy data. The harmonic extension is taken at each time \(t \in [0,1]\) independently. In the implementation, we use a standard finite difference scheme to compute the harmonic extension on spatial grids for each time grid point.

Note that if we assume \(\kappa = \kappa_0\) to be known outside of domain \(\Omega\), the measurements of \(m\) and \(\nabla \phi\) are equivalent, as \(-\kappa_0(0)p(x)\partial_\nu \phi(x) = m(x) \cdot n\) on \(\partial \Omega^+\).

In the case when we only have noisy boundary measurements, noise enters the algorithm through the boundary conditions in (4.4), and hence the harmonic extensions may not have very high regularity. In this case, we introduce numerical stabilizing techniques, e.g. mollification, to address this issue and minimize error propagation. We refer the readers to section 4.2 for further details of the techniques we employ for this purpose. We remark that the techniques of harmonic extension have been applied to various other problems, e.g. over point clouds and in machine learning [29].

It is clear to see that \(\lambda_{\phi}\) is redundant and \(\lambda_{\eta} = 0\) whenever \(0 \in A + B + C\). Hence, we can consider only \(C \setminus \{(\kappa, \mu), (\rho, m), (\alpha, \phi), (\lambda_{(\rho, m)}, (0,0))\}\). In this case, we perform a primal-dual hybrid gradient method for updating \(\lambda_{(\rho, m)}\):

\[
\begin{align*}
\lambda^{n+1, \text{temp}}_{\rho_1} &= I + \alpha_n \left( \frac{\mu_n^i}{\alpha_n} \right) \left( \frac{1}{\kappa_n^i} \frac{m_n^i}{m_n^0} - \frac{1}{\kappa_n^i} \frac{m_n^0}{m_n^0} \right)^{-1} \left( \frac{\lambda_n^i}{\lambda_n^0} - \alpha_n \left( \lambda_n^i - \lambda_n^0 \right) \right) \\
\lambda^{n+1, \text{temp}}_{\mu_1} &= 2 \left( \lambda^{n, \text{temp}}_{\rho_1} \right) - \left( \lambda^{n, \text{temp}}_{n, \text{temp}} \right).
\end{align*}
\]

(4.5)

4.1.2. Update of the inverse problem. In this part, we focus on the update for the inverse problem variables \((\kappa, \mu)\):

\[
\begin{align*}
\kappa^{n+1, \text{temp}} &= S_{\alpha_n^i} \left( 2 \kappa^n - \tilde{\kappa}^n - \alpha_n \sum_{i=1}^N \Lambda_{\kappa_0} (\kappa_i^0, \mu_i^0, \lambda_i^{n+1} - \lambda_i^{n, \text{temp}}) - \kappa_0 \right) + \kappa_0 \\
\kappa^{n+1} &= \tilde{\kappa}^n + \kappa^{n+1, \text{temp}} - \kappa^n \\
\kappa^{*+1} &= \max \{ \tilde{\kappa}, \tilde{\kappa}^{*+1} \} \\
\mu^{n+1, \text{temp}} &= S_{\alpha_n^i} \left( 2 \Pi_{\mu} (\mu^n) - \tilde{\mu}^n - \alpha_n^i \sum_{i=1}^N \Lambda_{\mu} (\lambda_i^{n+1, \text{temp}}, \alpha_i^0) - \Pi_{\mu}^i (\mu_0) \right) + \Pi_{\mu}^i (\mu_0) \\
\mu^{n+1} &= \tilde{\mu}^n + \mu^{n+1, \text{temp}} - \mu^n \\
\mu^{*+1} &= \Pi_{\mu} \left( \min \{ \tilde{\mu}, \mu^{*+1} \} \right).
\end{align*}
\]

(4.6)
where $S_\alpha(r)$ is the shrinkage operator given as $S_\alpha(r) = \text{sign}(r)\max\{|r| - \alpha, 0\}$, and

$$\Lambda_\alpha(\kappa, \rho_0, \lambda_0, \lambda_m) = \int_0^T \frac{1}{2(\kappa)^2} \|m(\cdot, s)\|^2 \lambda_\rho(\cdot, s) - \frac{1}{(\kappa)^2} \rho(\cdot, s) \lambda_m(\cdot, s)ds,$$

$$\Lambda_\mu(\lambda_\rho, a) = \int_0^T \int_{\Omega} \Lambda_1(\omega, y) \lambda_\rho(y, s) L(y, s) dy ds,$$

$$\Lambda_1(\omega, x) = \begin{pmatrix} \cos(\omega_1 \cdot x) & 0 & \cdots & 0 \\ 0 & \sin(\omega_1 \cdot x) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \sin(\omega_r \cdot x) \end{pmatrix}.$$ 

Since we have the $\mu, \omega \in C_{\text{odd}} \cup C_{\text{even}} := \{(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, ..., x_{r,1}, x_{r,2}) : x_{1,1} = x_{1,2} \forall i = 1, ..., r\}$, we write the projector $\partial \chi_{C_{\text{odd}} \cup C_{\text{even}}}$ (where we identify $C_{\text{odd}} \cup C_{\text{even}}$ with $\mathbb{R}^r$) as

$$\Pi_\mu : \mathbb{R}^r \rightarrow C_{\text{odd}} \cup C_{\text{even}} \cong \mathbb{R}^r$$

$$(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, ..., x_{r,1}, x_{r,2}) \mapsto \left(\frac{x_{1,1} + x_{1,2}}{2}, \frac{x_{2,1} + x_{2,2}}{2}, ..., \frac{x_{r,1} + x_{r,2}}{2}\right)$$

and its adjoint as

$$\Pi_\mu^* : \mathbb{R}^r \rightarrow \mathbb{R}^r$$

$$(x_1, x_2, ..., x_r) \mapsto (x_1, x_1, x_2, x_2, ..., x_r, x_r).$$

4.1.3. Update of the forward problem. As for the forward problem, we use primal–dual hybrid gradient (PDHG) method [6] to update $((\rho_i, m_i), (\phi_i, a_i))$ for each event $i$, for $1 \leq i \leq N$. The iterative updates contains three parts: firstly a proximal gradient descent step for $(\rho_i, m_i)$ with stepsizes $(\alpha^\mu_i, \alpha^m_i)$; then a proximal gradient ascent step for $(\phi_i, a_i)$ of stepsizes $(\alpha^\phi_i, \alpha^a_i)$; lastly an extrapolating step for $(\phi_i, a_i)$. Note that we make the choice of norm $\|\phi\|_{L^2_{\mu,i}} = \|\phi\|^2_{L^2_{\mu,i}} + \|\nabla \phi\|^2_{L^2_{\mu,i}}$ for $\phi$, based on the general-proximal primal–dual hybrid gradient method [15] that can be interpreted as a preconditioning step for obtaining a mesh-size-free convergence rate for the algorithm. Overall, the computation for the forward model follows the computational method proposed in [23, 24].

\[
\begin{align*}
(\rho_i^{n+1}, m_i^{n+1}) &= \arg\min_{(\rho, m)} \left\{ \mathcal{L}(\rho, m, (\rho_i, m_i), (\phi_i, a_i), (\kappa^{n+1}, \mu^{n+1})) \\
&+ \frac{1}{2 \alpha^\mu_i} \|\rho_i^n - \rho\|^2_{L^2_{\mu,i}} + \frac{1}{2 \alpha^m_i} \|m_i^n - m\|^2_{L^2_{\mu,i}} \right\} \\
(\phi_i^{n+1}, a_i^{n+1}) &= \arg\min_{(\phi, a)} \left\{ -\mathcal{L}(\phi, a, (\rho_i, m_i), (\rho_i^{n+1}, m_i^{n+1}), (\phi_i^{n+1}, a_i^{n+1})) \\
&+ \frac{1}{2 \alpha^\phi_i} \|\phi_i^n - \phi\|^2_{H^1_{\phi,i}} + \frac{1}{2 \alpha^a_i} \|a_i^n - a\|^2_{H^1_{a,i}} \right\} \\
(\phi_i^{n+1}, a_i^{n+1}) &= 2(\phi_i^{n+1}, a_i^{n+1}) - (\phi_i^n, a_i^n)
\end{align*}
\]

Assembling all three components described above, we arrive at the following algorithm for solving (4.1).
Algorithm 1. Inverse method for the nonlocal mean-field game system

\textbf{Input:} $(p_i^0, g_i, r_B^i)$ for $i = 1, \ldots, N$, $(\kappa_0, \mu_0)$

\textbf{Output:} $(\kappa^n, \mu^n)$ for $n = 1, \ldots, N_{\text{max}}$

\textbf{while} iteration $n < N_{\text{max}}$ \textbf{do}
  \begin{enumerate}
  \item Update for the adjoint problem:
    \begin{align*}
    \kappa^{n+1} &= \kappa^n + T_{\text{mask}}(\kappa_n, \kappa_0) - \kappa^n \\
    \end{align*}
  \end{enumerate}
  \begin{enumerate}
  \item Update the forward problem:
    \begin{align*}
    \rho_i^{n+1}, m_i^{n+1}, \phi_i^{n+1}, a_i^{n+1} &= \text{compute} (\rho_i, m_i, \phi_i, a_i) \text{ for } i = 1, \ldots, N.
    \end{align*}
  \end{enumerate}
  \textbf{end while}

4.2. Stabilizing techniques

Here, we discuss key numerical strategies for stabilizing algorithm 1. We refer to appendix 2 (in particular, algorithm 2) for more implementation details.

While the change of $\kappa^{n+1}$ is made from the accumulation of all measurement events (through $(\lambda_{p_i}^{n+1}, \lambda_{m_i}^{n+1})$), there is sometimes unexpected change of $\kappa^{n+1}(x)$ that makes the algorithm highly unstable. For instance, there may be a large $\kappa^{n+1}(x)$ at a single grid point. Moreover, we are using harmonic expansion method to update $(\lambda_{p_i}^{n+1}, \lambda_{m_i}^{n+1})$, which causes large variances of $\kappa(x)$ along the boundary $\partial \Omega$. Therefore, we add a cut-off function and a convolution kernel to the step (4.6) to have a smoother change in $\kappa^{n+1}(x)$ in space. Specifically, we have

\[
\begin{cases}
\tilde{\kappa}^{n+1} = \kappa^n + T_{\text{mask}}(\kappa_{n, \text{temp}}, \kappa_0) - \kappa^n \\
\kappa^{n+1} = \max \left\{ \varepsilon_1, \tilde{\kappa}^{n+1} * \psi \right\}
\end{cases}
\]

where $T_{\text{mask}}$ is a cut–off function that truncates the change of $\kappa$ near $\partial \Omega$ given by

\[ T_{\text{mask}}(\kappa, \kappa_0)(x) = \xi(x)(\kappa - \kappa_0)(x) + \kappa_0(x), \]

for a function $\xi(x)$ vanishing near $\partial \Omega$. As for the convolution

\[ \kappa^{n+1}(x) = \max \left\{ \varepsilon_1, \int_{\Omega'} \tilde{\kappa}^{n+1}(y) \psi(x-y) dy \right\}, \]

where the convolution kernel $\psi(x)$ satisfies $\int_{\Omega} \psi(x) dx = 1$.

On the other hand, after the inverse problem parameters $(\kappa^{n+1}, \mu^{n+1})$ are updated, we get a new pair of parameters for a set of MFG problems. It is unclear whether starting from $(\rho_i^n, m_i^n, \phi_i^n, a_i^n)$ and taking the update rule (4.7) once produces physical solutions for the new MFG system due to highly nonlinear dependence of the solution on the system parameters. Therefore, instead of preforming one iteration for the forward problem, we apply the PDHG algorithm for the forward problem until its error reaches a preset tolerance. More specifically, at every iteration $n$, with new system parameters $(\kappa^{n+1}, \mu^{n+1})$, we use $(\rho_i^n, m_i^n, \phi_i^n, a_i^n)$ as an initial guess and calculate the MFG solution accurately so that the primal–dual gap is smaller than residual the preset tolerance.
5. Numerical examples

This section demonstrates the efficiency and robustness of the inverse MFG algorithm with three examples. We also discuss details on the rule we used to choose the best reconstruction parameters.

5.1. Numerical implementation details

In this section, we present several numerical examples to illustrate the effectiveness of the new algorithm for the reconstruction of parameters in the MFG problem. We consider the spatial-time domain \( \Omega' \times [0, T] = [-1, 1]^2 \times [0, 1] \). In the following examples, the partial boundary measurements are taken along the domain \( \Omega = [-0.5, 0.5]^2 \). We refer as \( \partial \Omega \). The figure 1 gives an example of the forward measurement event.

In order to collect our observed data of the forward problem, we solve a set of MFG (3.3) with given \((\rho_{0,i}, g_i)\) and \((\kappa, \mu)\) by finite difference method with a mesh of size \((0.05, 0.04)\) in space-time. Each problem is solved via primal-dual optimization approach with primal-dual gap \( \varepsilon_{\text{tol}} < 2 \varepsilon - 3 \). The initial density function \( \rho_{0,i} \) is the average of two Gaussian functions with centers \( x_G \in \Omega' \setminus \Omega \). The final cost function \( g(x) \) is smooth and has a smaller value around a single point \( x_0 \in \Omega' \setminus \Omega \) such that densities are concentrated in the neighborhood of \( x_0 \) at the final time. We want to point out that there is room to improve the initial density function and final cost function choices. We choose this set of \((\rho_{0,i}, g_i)\) to ensure that the density’s movement covers the domain \( \Omega \) as completely as possible. We also expect the nonlocal interaction among agents to be better reflected at the partial boundary measurements by setting the initial density as two Gaussians rather than one.

We only take 16 forward measurement events for each of the following numerical examples. The partial boundary measurement means that we only collect the \( \rho, m \) along the boundary of \( \partial \Omega \) in each event. Therefore, the resulting inverse problem is severely ill-posed.

To test the robustness of our reconstruction algorithm, we add some random noise to the measurements as follows:

\[
(\rho, m \cdot n)^\delta(t_i, x_j) = ((1 + \epsilon_n \delta_{ij, 1})\rho, (1 + \epsilon_n \delta_{ij, 2})m \cdot n)(t_i, x_j), \tag{5.1}
\]

where \( \{(t_i, x_j)\}_{i=1,...,I, j=1,...,J} \in [0, T] \times \partial \Omega^+ \) represents sampling points on the measurement boundary \([0, T] \times \partial \Omega^+ \), \{\delta_{ij, 1}, \delta_{ij, 2}\}_{i=1,...,I, j=1,...,J} \) are i.i.d. random variables uniformly distributed on the interval \([-0.5, 0.5]\) and \( \epsilon_n \) corresponds to the noise level in the data, which is always set to be \( \epsilon_n = 10\% \) in all our examples.

From the noisy observed data \( \{(\rho, m \cdot n)^\delta(t_i, x_j)\}_{i=1,...,I, j=1,...,J} \) on the sampling points of the measurement surface, we then use the algorithm to reconstruct the forward problem parameters \((\kappa, \mu)\). Recall that we parameterized the running cost \( L(x, v) := \frac{1}{\sqrt{\kappa(\gamma v)}} |v|^2 \) by \( \kappa \) and nonlocal kernel \( K(x, y) := \zeta(x; \mu, \omega) \cdot \zeta(y; \mu, \omega) \) by \( \mu \). Since we aim at recovering the model on a given domain with fixed grid points, we fix the choice of \( \omega \), and only seek sparse recovery of \( \mu \).

In the following examples, we use a set of parameters uniformly, without tuning. \( \gamma_c = 0.2, \gamma_\mu = 0.1, \alpha_c = 0.1, \alpha_\mu = 0.1, \alpha_\lambda = 10 \). We set the lower-bound projection parameter \( \epsilon_1 = \kappa_c \), this is based on the additional assumption of the model parameters that \( \kappa(x) \geq \kappa_c \) for \( x \in \Omega' \). The projection parameter for kernel coefficient is \( \epsilon_2 = 1 \).
To account for unknown ground truth of the model parameters, we introduce $Res$ to quantify the quality of the reconstructed parameters.

$$Res^n = \sum_i \int_{[\Omega]^{1+}} \left( \| \rho^n_i - pr_1 \tilde{r}_{B,i} \|^2 + \| m^n_i \cdot n - pr_2 \tilde{r}_{B,i} \|^2 \right),$$

where $\rho^n_i, m^n_i$ are the solution of the forward MFG problem with $i$th choice of initial density and final cost function with the reconstructed parameter $(\kappa^n, \mu^n)$ at the $n$th iteration of the algorithm. The boundary residual $Res$ measures how much the new boundary measurements of the MFG model with the recovered parameters deviate from the given partial measurements.

If $(\kappa, \mu) = (\kappa_{\text{true}}, \mu_{\text{true}})$, we would expect that $Res$ is close to 0. Therefore, we pick the reconstructed parameters at $n_{\text{opt}}$th iteration by taking

$$n_{\text{opt}} = \arg \min_n Res^n, \quad (\kappa_{\text{opt}}, \mu_{\text{opt}}) = (\kappa_{\text{opt}}^n, \mu_{\text{opt}}^n).$$

When we implemented the algorithm, we observed that the quantity $Res^n$ first decreased then increased with respect to the iteration. We also observed that with large enough number of iterations, (for example, 1500), the inverse problem is contaminated and the reconstruction of MFG coefficients are very bad. In the following examples, we take fixed number of iterations $N = 1500$ for the inverse algorithm, and pick the reconstructed model parameters accordingly.

5.2. Example 1

This example tests a running cost $\kappa(x)$ with a bump at point $(0.25, 0.25)$, which means the density that travel crossing near this point has a lower cost than other routes. The density are also expected to accelerate when they travel across this point. The nonlocal kernel $K(x, y)$ is constructed via a Gaussian function plus some sparse terms in forms of $\mu^2 \cos(\omega_k \cdot x)$. The nonlocal kernel, in general, penalizes being too concentrated. The amplify of certain Fourier frequencies determines the agents’ particular interaction preferences. Specifically, we have the following:

$$\kappa(x) = 2 + 4 \exp \left( -\frac{(x_1 - 0.25)^2 + (x_2 - 0.25)^2}{0.1^2} \right),$$

$$\kappa_0(x) = 2,$$

$$K(x, y) = K_0(x, y) + K_s(x, y) + k_0,$$

$$K_0(x, y) = \frac{1}{5} \exp \left( -\frac{1}{2} x^2 + y^2 \right),$$

$$K_s(x, y) = 0.2094^2 \left( \cos(\pi(x_1 - y_1)) + \cos(\pi(x_2 - y_2)) \right) + 0.2613^2 \left( \cos(\pi(x_1 - y_1)) + \pi(x_2 - y_2) \right) + \cos(-\pi(x_1 - y_1) + \pi(x_2 - y_2)), $$

where $x = [x_1, x_2], y = [y_1, y_2]$. We have $\mu_0$, which represents $K_0(x, y)$ via the expansions form $\mu^2 \cos(\omega_k \cdot x)$, known. The variable $k_0$ is a given constant value that makes the kernel integration $\int K(x, y) dxdy = 1$. Varying this constant corresponding to changing the coefficient of the zero Fourier mode $(0, 0)$. This constant $k_0$ does not change the intensity of repulsion effect among the agents, since $\int k_0 \rho(y) dy = k_0 \int \rho_0(y) dy$ is uniform over the domain $\Omega'$. With $K(x, y) = \zeta(x; \mu, \omega) \cdot \zeta(y; \mu, \omega)$ for $\mu, \omega \in C_{\text{odd} = \text{even}}$, we omit the even entries (e.g. $\mu_k, \omega_k$) and express the kernel $K$ as follows:
Figure 2. The residual $\text{Res}^n$ and the $\max_x \kappa^n(x)$ at $n$th iteration.

Figure 3. From left to right: the true running cost $\kappa(x)$; the reconstructed running cost $\kappa_{\text{opt}}(x)$ at iteration $n_{\text{opt}}$; the coefficient representation of nonlocal kernel $K(x, y)$ in vector form, where $x$-axis represents different Fourier mode $\omega$ and the $y$-axis corresponds to the coefficients $\mu$.

$$
\mu_s = (0.2094, 0.2094, 0.2613, 0.2613),
$$

$$
\omega_s = ((\pi, 0), (0, \pi), (\pi, \pi), (-\pi, \pi)).
$$

Here, we also assume that $\kappa_c = 2$ and $\kappa(x) = 2$ for $x \in \Omega' \setminus \Omega$ is known.

Given $(\kappa_0, \mu_0)$ and the noisy partial boundary measurements with corresponding event parameters $(\rho_{0,i}, g_i, r_{R,i})$, we apply our inverse algorithm. The results are shown in figures 2 and 3. In figure 2, we plot the residual $\text{Res}^n$ and the $\max_x \kappa^n(x)$ along the iteration. We see that the residual oscillates and decreases first, then bounces back and increases. In figure 3, we show the reconstruction of model parameters by taking $n_{\text{opt}} = \arg \min_n \text{Res}^n$, $(\kappa_{\text{opt}}, \mu_{\text{opt}}) = (\kappa_{\text{true}}, \mu_{\text{true}})$. We see that the reconstructed $\kappa_{\text{opt}}(x)$ has a single bump sits near $(0.25, 0.25)$. The shape of the bump is not as sharp as the ground truth $\kappa$. The maximal value of running cost $\max_x \kappa_{\text{true}}(x) = 6$; while $\max_x \kappa_{\text{opt}}(x) = 4$. As for the non-local kernel, we have $\mu_{\text{opt}}$ nicely reconstructed, where $\mu_{\text{true}} = \mu_0 + \mu_s \approx \mu_{\text{opt}}$. This example shows that our inverse algorithm is robust to noise and can recover the model parameters $(\kappa, \mu)$ simultaneously.
5.3. Example 2

In this example, we make the $\kappa(x)$ more complicated by having two bumps sitting diagonally. We except that if the density travels across these two bumps, it will accelerate twice. The model set-up is as follows:

$$\kappa(x) = 2 + 4 \exp \left(-\frac{(x_1 + 0.25)^2 + (x_2 - 0.25)^2}{0.1^2}\right) + 4 \exp \left(-\frac{(x_1 - 0.25)^2 + (x_2 + 0.25)^2}{0.1^2}\right),$$

$$\kappa_0(x) = 2, \kappa_c = 2,$$

$$K(x,y) = K_0(x,y) + K_s(x,y) + k_0,$$

$$K_0(x,y) = \frac{1}{5} \exp \left(-\frac{1}{2} \frac{x^2 + y^2}{0.4^2}\right),$$

$$\mu_s = (0.3374, 0.3374, 0.2942, 0.2942),$$

$$\omega_s = ((\pi,0), (0, \pi), (2\pi,0), (0,2\pi)).$$

We can see from the figure 4 that recovered bumps are well separated, and their locations are accurately captured. Reconstructed bumps are more spread compared to the ground truth, and there is some noise on upper left and bottom right corners of the domain $\Omega'$. The nonlocal kernel is reconstructed nicely as shown in figure 4(right). A precise sparse representation of $K_s(x,y)$ is recovered. Considering the severe ill-posedness of the inverse problem with 10% multiplicative noise added to the boundary measurements, the reconstruction quality is quite satisfactory.

5.4. Example 3

In this example, we modify the $\kappa(x)$ by having two bumps sitting in parallel. Similar to the Example 2, the density would prefer to move crossing these bumps. We set the nonlocal kernel with $K_s$ containing Fourier modes with higher frequency.
We have the reconstruction result shown in figure 5. The two parallel sitting bumps are well separated and located with reasonable accuracy. Again, the bumps are diffused with some noise near the upper boundary of $\Omega$. The nonlocal kernel is recovered very nicely.

6. Conclusion

In this paper, we formulate a new class of inverse MFG problems given only partial boundary measurements. A novel model recovery algorithm is proposed based on the saddle point formulation of MFGs. We demonstrate the robustness and effectiveness of the numerical inverse algorithm with several examples, where the MFG model parameters are reconstructed accurately. Our algorithm can be further generalized to other inverse problems with saddle point structure in the forward problem.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).
Acknowledgments

Y T Chow, S W Fung, S Liu, L Nurbekyan and S Osher thank the funding from AFOSR MURI FA9550-18-1-0502 and ONR Grant Nos. N00014-18-1-2527, N00014-20-1-2093 and N00014-20-1-2787. Authors are listed in alphabetical order.

Appendix A. The KKT conditions of the inverse mean-field game problem

We denote by

$$\mathcal{L}(\cdot) := \mathcal{L}\left(\rho, g, (\rho_j, \phi_j), (\kappa, \mu)\right),$$

the function $\mathcal{L}$ defined in equation (3.3) for simplicity. The KKT conditions for the inverse MFG problem are then

$$\begin{align*}
\Pi^*_B[(\rho, m)] \Pi_B[(\rho, m)] (\rho_j, m_j - \tilde{r}_B) + \partial^2_{(\rho, m)} \mathcal{L}(\cdot) \lambda(\rho, m) &= 0, \quad i = 1, \ldots, N, \\
\Pi^*_B[\alpha, \phi] \Pi_B[\alpha, \phi] (\alpha_i, \phi_i) - \bar{s}_B, i &- \partial^2_{(\alpha_i, \phi_i)} \mathcal{L}(\cdot) \lambda(\alpha_i, \phi_i) = 0, \quad i = 1, \ldots, N, \\
\partial_{(\kappa, \mu)} R_1(\kappa, \mu) + \partial_{(\kappa, \mu)} R_2(\kappa, \mu) + \sum_{i=1}^N \left( \partial_{(\kappa, \mu)} \partial_{(\rho, m)} \mathcal{L}(\cdot) \lambda(\rho, m) \right), \\
- \sum_{i=1}^N \left( \partial_{(\kappa, \mu)} \partial_{(\alpha_i, \phi_i)} \mathcal{L}(\cdot) \lambda(\alpha_i, \phi_i) \right) &= 0,
\end{align*}$$

Furthermore, the derivatives of $\mathcal{L}$ are given by

$$\begin{align*}
\partial_{(\rho, m)} \mathcal{L} &= \left( \partial_\phi + \nu \Delta \phi - \frac{1}{\kappa} |m|^2 \rho^2 + a \cdot \zeta(\cdot; \mu, \omega), \nabla \phi + \frac{1}{\kappa} m \right), \\
\partial_{(\alpha_i, \phi_i)} \mathcal{L} &= \left( -a + \int_{\Omega'} \rho(y) \zeta(y; \mu, \omega) dx, -\partial_\phi \nu + \nu \Delta \rho - \nabla \cdot m \right), \\
\partial^2_{(\rho, m), (\rho, m)} \mathcal{L} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \\
\partial^2_{(\alpha_i, \phi_i), (\alpha_i, \phi_i)} \mathcal{L} &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \\
\partial_{(\kappa, \mu)} \partial_{(\rho, m)} \mathcal{L} &= \begin{pmatrix} \frac{1}{\kappa} |m|^2 \rho^2 - \frac{1}{\kappa} m \rho \end{pmatrix}, \\
\partial_{(\kappa, \mu)} \partial_{(\alpha_i, \phi_i)} \mathcal{L} &= \begin{pmatrix} \frac{1}{\kappa} \rho \Lambda_1(\cdot, x) a - \frac{1}{\kappa} \rho \Lambda_1(\cdot, x) \rho \end{pmatrix}, \\
\partial_{(\kappa, \mu)} \partial_{(\alpha_i, \phi_i)} \mathcal{L} &= \begin{pmatrix} 0 & 0 \\ \Lambda_1(\cdot, x) a \rho \Lambda_1(\cdot, x) \rho \end{pmatrix}
\end{align*}$$

where the variable $\Lambda_1(\cdot, x)$ is defined in section 4.1.2.
Appendix B. The inverse algorithm

Algorithm 2. A detailed inversion algorithm for the nonlocal mean-field game system

Input: \((\rho_{0,i}, \hat{g}_i, \hat{f}_i)\) for \(i = 1, \ldots, N\), \((\kappa_0, \mu_0)\)

Output: \((\kappa^n, \mu^n)\) for \(n = 1, \ldots, N_{\max}\)

while \(n < N_{\max}\) do

1. Update for the adjoint problem by computing \((\lambda_{n_i}^{n+1}, \lambda_{\mu_i}^{n+1})\)

\[
\begin{align*}
\Delta \eta^n_{n_i} &= 0 \quad \text{in} \ \Omega_i \\
\eta^n_{n_i} &= \rho_i^n - \text{pr}_i \hat{f}_i \quad \text{on} \ \partial \Omega^+ \\
\Delta \eta^n_{\mu_i} &= 0 \quad \text{in} \ \Omega_i \setminus \Omega \\
\eta^n_{\mu_i} &= \rho_i^n - \text{pr}_i \hat{f}_i \quad \text{on} \ \partial \Omega^- \\
\eta^n_{\mu_i} &= 0 \quad \text{on} \ \partial \Omega' \\
\Delta \zeta^n_{n_i} &= 0 \quad \text{in} \ \Omega_i \\
\partial \zeta^n_{n_i} &= n_i \quad \text{on} \ \partial \Omega^+ \\
\Delta \zeta^n_{\mu_i} &= 0 \quad \text{in} \ \Omega_i \setminus \Omega \\
\partial \zeta^n_{\mu_i} &= n_i \quad \text{on} \ \partial \Omega^- \\
\partial \zeta^n_{\mu_i} &= 0 \quad \text{on} \ \partial \Omega' \\
\end{align*}
\]

\[
\begin{align*}
\left( \lambda_{n_i}^{n+1}, \lambda_{\mu_i}^{n+1} \right) &= \\
&= \left( \lambda_{n_i}^{n+1, \text{temp}}, \lambda_{\mu_i}^{n+1, \text{temp}} \right) = \\
&= \left[ 1 + \alpha_n^{n+1} \left( \frac{[\nabla \rho_i^n]^2}{4 \rho_i^n} - \frac{1}{4 \rho_i^n} \frac{\partial \rho_i^n}{\partial \rho_i^n} \right) \right]^{-1} \left( \left( \lambda_{n_i}^n \lambda_{\mu_i}^n \right) - \alpha_n^{n+1} \left( \eta^n_{n_i} - \nabla \mu_i^{n+1} \right) \right) \\
&= 2 \left( \lambda_{n_i}^{n+1, \text{temp}}, \lambda_{\mu_i}^{n+1, \text{temp}} \right) - \left( \lambda_{n_i}^{n, \text{temp}}, \lambda_{\mu_i}^{n, \text{temp}} \right)
\end{align*}
\]

2. Update for the inverse problem by computing \((\kappa^{n+1}, \mu^{n+1})\)

\[
\begin{align*}
\kappa^{n+1} &= S_{\omega_{\kappa}} \left( 2 \kappa^n - \kappa^n - \alpha_{\kappa} \sum_{i=1}^{N} \Lambda_{\kappa} \left( \kappa^n, m_i^n, \beta_i^n, \lambda_{n_i}^{n+1}, \lambda_{\mu_i}^{n+1} \right) - \kappa_0 \right) + \kappa_0 \\
\kappa^{n+1} &= \kappa^n + \max \{ \epsilon_1, \kappa^{n+1} \} \\
\kappa^{n+1} &= \max \left\{ \epsilon_1, \kappa^{n+1} \right\} \\
\mu^{n+1} &= S_{\omega_{\mu}} \left( 2 \Pi_{\mu}^i (\mu^n) - \mu^n - \alpha^\mu \sum_{i=1}^{N} \Lambda_{\mu} \left( \lambda_{n_i}^{n+1}, m_i^n, \phi_i^n, a_i^n \right) - \Pi_{\mu}^i (\mu_0) \right) + \Pi_{\mu}^i (\mu_0) \\
\mu^{n+1} &= \mu^n + \mu^{n+1, \text{temp}} - \mu^n \\
\mu^{n+1} &= \Pi_{\mu} (\min \{ \epsilon_2, \mu^{n+1} \}) \\
\end{align*}
\]

3. Update forward problem by computing \((\rho_{i}^{n+1}, m_i^{n+1}, \phi_i^{n+1}, a_i^{n+1})\) with the forward mean-field game algorithm for \(i = 1, \ldots, N\).

Apply the iterative Algorithm 3 given input \((\rho_{0,i}, \hat{g}_i, \hat{f}_i, \phi_i^n, a_i^n)\) and assign

\[
(\rho_{n+1}, m_{n+1}, \phi_{n+1}^{n+1}) := (\rho_i^n, m_i^n, \phi_i^n, a_i^n) \\
n := n + 1
\]

end while
Appendix C. The forward algorithm for nonlocal mean-field games

Algorithm 3. Iterative algorithm for the nonlocal mean-field game system

Input: \((\rho_0, g), (\kappa, \mu)\), a set of initial guess \((\rho^0, m^0, \phi^0, a^0)\), \(\varepsilon_{tol}\), a set of stepsizes \((\alpha^l, \alpha^l_{\phi}, \alpha^l_{a}, \alpha^l_{a})\)

Output: \((\rho^l, m^l, \phi^l, a^l)\)

while iteration \(j < J_{\text{max}}\) and primal-dual gap \(PD(\rho^l, m^l, \phi^l, a^l) \geq \varepsilon_{tol}\) do

\[
\begin{align*}
\rho^{l+1, \text{temp}} &= \arg\min_{\rho} \left\{ \mathcal{L}((\rho_0, g), (\rho, m), (a^l, \phi^l), (\kappa, \mu)) + \frac{1}{2\alpha_{\rho}} \|\rho - \rho_0\|^2_{L^2} + \frac{1}{2\alpha_{m}} \|m - m^0\|^2_{L^2} \right\} \\
\phi^{l+1, \text{temp}} &= \arg\min_{\phi} \left\{ -\mathcal{L}(\rho_0, g, (\rho^{l+1}, m^{l+1}), (a, \phi), (\kappa, \mu)) + \frac{1}{2\alpha_{\phi}} \|\phi - \phi_0\|^2_{H^1} + \frac{1}{2\alpha_{a}} \|a - a_0\|^2_{L^2} \right\} \\
\end{align*}
\]

\(j \leftarrow j + 1\)

end while

return \((\rho^l, m^l, \phi^l, a^l)\)

References

[1] Achdou Y, Buera F J, Lasry J-M, Lions P-L and Moll B 2014 Partial differential equation models in macroeconomics Phil. Trans. R. Soc. A 372 20130397
[2] Achdou Y, Cardaliaguet P, Delarue Fclos, Porretta A and Santambrogio F 2020 Mean Field Games (Lecture Notes in Mathematics vol 2281) ed P Cardaliaguet and A Porretta (Cham, Florence: Springer, Centro Internazionale Matematico Estivo (C.I.M.E.)) Fondazione CIME/CIME Foundation Subseries
[3] Agrawal S, Lee W, Fung S W and Nurbekyan L 2022 Random features for high-dimensional non-local mean-field games J. Comput. Phys. 459 111136
[4] Burger M, Piechmann J-F and Wolfram M-T 2020 Data assimilation in price formation Inverse Probl. 36 064003
[5] Cardaliaguet P and Lehalle C-A 2018 Mean field game of controls and an application to trade crowding Math. Financ. Econ. 12 335–63
[6] Chambloue A and Pock T 2011 A first-order primal-dual algorithm for convex problems with applications to imaging J. Math. Imaging Vis. 40 120–45
[7] Cirant M 2015 Multi-population mean field games systems with Neumann boundary conditions J. Math. Pures Appl. 103 1294–315
[8] Davis D and Yin W 2017 A three-operator splitting scheme and its optimization applications Set-Valued Var. Anal. 25 829–58
[9] Ding L, Wuchen Li, Osher S and Yin W 2020 A mean field game inverse problem (arXiv:2007.11551)
[10] Gomes D A, Nurbekyan L and Pimentel E A 2015 Economic Models and Mean-Field Games Theory (Rio de Janeiro: IMPA Mathematical Publications. Instituto Nacional de Matemática Pura e Aplicada (IMPA))
[11] Gomes D A, Pimentel E A and Voskanyan V 2016 Regularity Theory for Mean-Field Game Systems (Springer Briefs in Mathematics) (Cham: Springer)
[12] Heaton H, McKenzie D, Qiuwei Li, Fung S W, Osher S and Yin W 2021 Learn to predict equilibria via fixed point networks (arXiv:2106.00906)

ORCID iD

Yat Tin Chow © https://orcid.org/0000-0002-7703-5901
[13] Huang M, Caines P E and Malhamé R P 2007 Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized $\varepsilon$-Nash equilibria IEEE Trans. Automat. Control 52 1560–71
[14] Huang M, Malhamé R P and Caines P E 2006 Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle Commun. Inf. Syst. 6 221–51
[15] Jacobs M, Léger F, Wuchen Li and Osher S 2019 Solving large-scale optimization problems with a convergence rate independent of grid size SIAM J. Numer. Anal. 57 1100–23
[16] Kachroo P, Agarwal S and Sastry S 2015 Inverse problem for non-viscous mean field control: Example from traffic IEEE Trans. Automat. Control 61 3412–21
[17] Lasry J-M and Lions P-L 2006 Jeux à champ moyen. I. Le cas stationnaire C. R. Math. Acad. Sci. Paris 343 619–25
[18] Lasry J-M and Lions P-L 2006 Jeux à champ moyen. II. Horizon fini et contrôle optimal C. R. Math. Acad. Sci. Paris 343 679–84
[19] Lasry J-M and Lions P-L 2007 Mean field games Jpn. J. Math. 2 229–60
[20] Laurière M and Tangpi L 2020 Convergence of large population games to mean field games with interaction through the controls (arXiv:2004.08351)
[21] Lee W, Liu S, Tembine H, Wuchen Li and Osher S 2021 Controlling propagation of epidemics via mean-field control SIAM J. Appl. Math. 81 190–207
[22] Lin A T, Fung S W, Wuchen Li, Nurbekyan L and Osher S J 2021 Alternating the population and control neural networks to solve high-dimensional stochastic mean-field games Proc. Natl Acad. Sci. 118 e2024713118
[23] Liu S, Jacobs M, Wuchen Li, Nurbekyan L and Osher S J 2021 Computational methods for first-order nonlocal mean field games with applications SIAM J. Numer. Anal. 59 2639–68
[24] Liu S and Nurbekyan L 2021 Splitting methods for a class of non-potential mean field games J. Dyn. Games 8 467–86
[25] Liu Z, Wu B and Lin H 2018 A mean field game approach to swarming robots control 2018 Annual American Control Conf. (ACC) (IEEE) pp 4293–8
[26] Nurbekyan L and Saüde J 2018 Fourier approximation methods for first-order nonlocal mean-field games Port. Math. 75 367–96
[27] Rong X 1998 A bochner theorem and applications Duke Math. J. 91 381–92
[28] Ruthotto L, Osher S J, Wuchen Li, Nurbekyan L and Fung S W 2020 A machine learning framework for solving high-dimensional mean field game and mean field control problems Proc. Natl Acad. Sci. 117 9183–93
[29] Shi Z, Sun J and Tian M 2018 Harmonic extension on the point cloud Multiscale Model. Simul. 16 215–47
[30] Stella L and Bauso D 2018 Mean-field games for bio-inspired collective decision-making in dynamical networks (arXiv:1802.03435)
[31] Yang J, Xiaojing Y, Trivedi R, Huan X and Zha H 2018 Learning deep mean field games for modeling large population behavior Int. Conf. Learning Representations (ICLR) p 2018