A NUMERICAL ALGORITHM FOR INVERSE PROBLEM FROM PARTIAL BOUNDARY MEASUREMENT ARISING FROM MEAN FIELD GAME PROBLEM

YAT TIN CHOW, SAMY WU FUNG, SITING LIU, LEVON NURBEKYAN, AND STANLEY OSHER

ABSTRACT. In this work, we consider a novel inverse problem in mean-field games (MFG). 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.

1. INTRODUCTION

The basis for the 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 (PDE):

$$(1.1) \begin{cases} -\partial_t \phi(x,t) - \nu\Delta\phi(x,t) + H(x, \nabla_x\phi(x,t)) = F(x, \rho(\cdot,t)), \text{ in } \Omega' \times (0,1), \\ \partial_t\rho(x,t) - \nu\Delta\rho(x,t) - \nabla_x \cdot (\rho(x,t)\,\nabla_p H(x, \nabla_x\phi(x,t))) = 0, \text{ in } \Omega' \times (0,1), \\ \rho(x,0) = \rho_0(x), \ \phi(x,1) = g(x), \text{ in } \Omega'. \end{cases}$$

Here, $\rho(\cdot,t)$, $t \in [0,1]$ represents the population distribution over the state space Ω' 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

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

Introduced in [19, 17, 18] and [14, 13], 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 [28, 22], 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), (1.2), *for instance,*

$$\Big(\rho\left(x,s\right), -\rho\left(x,s\right)\nabla_p H(x,\nabla_x\phi(x,t))\Big)\big|_{\partial\Omega\times(0,T)},$$

for finitely many examples of ρ_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, policy-making, diplomacy, election modeling, catastrophe management, and evacuation planning.

Note that $m(x,s) = -\rho\left(x,s\right)\nabla_p H(x,\nabla_x\phi(x,s))$ 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 [11, 1, 5] 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 [26, 23, 24] 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 [9, 16, 4] 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

$$(2.1) \qquad \min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} F(u, x, y, c),$$

where $F : \mathcal{U} \times X \times Y \times \mathcal{D} \to \mathbb{R} \bigcup \{\pm\infty\}$ a smooth functional such that $(x, y) \mapsto F(u, x, y, c)$ 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 F, while u represents boundary and initial-terminal conditions. Given model parameters c, we define a boundary measurement map Λ_c as follows:

$$\begin{aligned} \Lambda_c : \mathcal{U} &\to \Pi_B X \times \Pi_B Y \\ u &\mapsto (\Pi_{B,x}(x), \Pi_{B,y}(y)) \text{ where } (x, y) \in \arg\min_x \arg\max_y F(u, x, y, c), \end{aligned}$$

where $\Pi_{B,x}, \Pi_{B,y}$ denote a projection operator that represent the partial boundary measurements of x, y available. We note that u 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

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

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

$$(2.2) \quad \inf_{\{x_i,y_i\}_{i=1}^N,c} \left\{ \sum_{i=1}^N \frac{1}{2} \|\Pi_{B,x}(x_i) - \tilde{r}_{B,i}\|^2 + \sum_{i=1}^N \frac{1}{2} \|\Pi_{B,y}(y_i) - \tilde{s}_{B,i}\|^2 + R(c) : \right.$$
$$\left. (x_i, y_i) \in \arg\min_x \arg\max_y F(u_i, x, y, c) \right\},$$

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

$$(2.3) \quad \inf_{\{x_i,y_i\}_{i=1}^N,c} \sup_{\{\lambda_{x_i},\lambda_{y_i}\}_{i=1}^N} \left\{ \sum_{i=1}^N \frac{1}{2} \|\Pi_{B,x}(x_i) - \tilde{r}_{B,i}\|^2 + \sum_{i=1}^N \frac{1}{2} \|\Pi_{B,y}(y_i) - \tilde{s}_{B,i}\|^2 + R(c) \right.$$
$$\left. + \sum_{i=1}^N \langle \partial_{x_i} F(u_i, x_i, y_i, c), \lambda_{x_i} \rangle - \sum_{i=1}^N \langle \partial_{y_i} F(u_i, x_i, y_i, c), \lambda_{y_i} \rangle \right\}.$$

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

$$(2.4) \quad \begin{cases} \Pi_{B,x}^*[\Pi_{B,x}(x_i) - \tilde{r}_{B,i}] + \partial_{x_i,x_i}^2 F(u_i, x_i, y_i, c)\lambda_{x_i} & = 0, \\ \Pi_{B,y}^*[\Pi_{B,x}(y_i) - \tilde{s}_{B,i}] - \partial_{y_i,y_i}^2 F(u_i, x_i, y_i, c)\lambda_{y_i} & = 0, \\ \partial_c R(c) + \sum_{i=1}^N \langle \partial_c \partial_{x_i} F(u_i, x_i, y_i, c), \lambda_{x_i} \rangle - \sum_{i=1}^N \langle \partial_c \partial_{y_i} F(u_i, x_i, y_i, c), \lambda_{y_i} \rangle & = 0, \\ \partial_{x_i} F(u_i, x_i, y_i, c) & = 0, \\ -\partial_{y_i} F(u_i, x_i, y_i, c) & = 0, \end{cases}$$

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

$$A(c, (x, y), (\lambda_x, \lambda_y)) = \begin{pmatrix} \partial_c R(c) \\ (0, 0) \\ (0, 0) \end{pmatrix},$$

and

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

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:

$$
\begin{cases}
\begin{cases}
\lambda_{y_i}^{n+1} & = [1 - \alpha_{\lambda_{y_i}} \partial_{y_i,y_i}^2 F(u_i, x_i^n, y_i^n, c^n)]^{-1} \left(\lambda_{y_i}^n - \alpha_{\lambda_{y_i}} \Pi_{B,y}^*[\Pi_{B,y}(y_i^n) - \tilde{s}_{B,i}] \right) \\
\lambda_{x_i}^{n+1,\text{temp}} & = [1 + \alpha_{\lambda_{x_i}} \partial_{x_i,x_i}^2 F(u_i, x_i^n, y_i^n, c^n)]^{-1} \left(\lambda_{x_i}^n - \alpha_{\lambda_{x_i}} \Pi_{B,x}^*[\Pi_{B,x}(x_i^n) - \tilde{r}_{B,i}] \right) \\
\lambda_{x_i}^{n+1} & = 2\lambda_{x_i}^{n+1,\text{temp}} - \lambda_{x_i}^{n,\text{temp}}
\end{cases} \\
\begin{cases}
c^{n+1} & = (I + \alpha_c \partial_c R)^{-1} \Big[c^n - \alpha_c \sum_{i=1}^N \langle \partial_c \partial_{x_i} F(u_i, x_i^n, y_i^n, c^n), \lambda_{x_i}^{n+1} \rangle \\
& \quad + \alpha_c \sum_{i=1}^N \langle \partial_c \partial_{y_i} F(u_i, x_i^n, y_i^n, c^n), \lambda_{y_i}^{n+1} \rangle \Big]
\end{cases} \\
\begin{cases}
x_i^{n+1} & = [1 + \alpha_{x_i} \partial_{x_i} F(u_i, \cdot, y_i^n, c^{n+1})]^{-1}(x_i^n) \\
y_i^{n+1,\text{temp}} & = [1 - \alpha_{y_i} \partial_{y_i} F(u_i, x_i^{n+1}, \cdot, c^{n+1})]^{-1}(y_i^n) \\
y_i^{n+1} & = 2y_i^{n+1,\text{temp}} - y_i^{n,\text{temp}},
\end{cases}
\end{cases}
$$

where $(\alpha_{\lambda_{x_i}}, \alpha_{\lambda_{y_i}}, \alpha_c, \alpha_{x_i}, \alpha_{y_i})$ 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 [26, 23, 24] 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:

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

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

(3.2)

$$
\inf_{\phi} \sup_{\rho,m} \left\{ -\int_{\Omega'} \phi(x,0)\rho_0(x)dx - \int_{\Omega'} \int_0^1 (\rho\phi_t + \nu\rho\Delta\phi + m \cdot \nabla\phi) \, dtdx \right.
$$

$$
\left. - \int_{\Omega'} \int_0^1 \frac{1}{2\kappa(x)} \frac{\|m\|^2}{2\rho} dtdx - \frac{1}{2} \int_{\Omega' \times \Omega'} K(x,y)\rho(x,t)\rho(y,t)dxdy - \chi_{\rho \geq 0} + \chi_{\phi(x,1)=g(x)} \right\}
$$

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

$$\frac{1}{2}\int_{\Omega'\times\Omega'} K(x,y)\rho(x,t)\rho(y,t)dxdy$$

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

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

$$= \sum_{k=1}^{r} \left(\mu_k^2 \cos(\omega_k \cdot x)\cos(\omega_k \cdot y) + \mu_k^2 \sin(\omega_k \cdot x)\sin(\omega_k \cdot y)\right)$$

for some $\{\omega_k\} \subset \mathbb{R}^d$, and $\{\mu_k^2\} \subset \mathbb{R}_+$. Denoting by

$$\mu = \left(\mu_{1,1}, \mu_{1,2}, \mu_{2,1}, \cdots, \mu_{r,1}, \mu_{r,2}\right) \quad \omega = \left(\omega_{1,1}, \omega_{1,2}, \omega_{2,1}, \cdots, \omega_{r,1}, \omega_{r,2}\right)$$

$$C_{\text{odd=even}} = \left\{ (x_{1,1}, x_{1,2}, x_{2,1}, \cdots, x_{r,1}, x_{r,2}) : x_{i,1} = x_{i,2} \right\}$$

$$\zeta(x;\mu,\omega) = \left(\mu_{1,1}\cos(\omega_{1,1}\cdot x), \mu_{1,2}\sin(\omega_{1,2}\cdot x), \cdots, \mu_{r,1}\cos(\omega_{r,1}\cdot x), \mu_{r,2}\sin(\omega_{r,2}\cdot x) \right)$$

we 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 $a(t) = (a_{1,1}(t), a_{1,2}(t), .., a_{r,1}(t), a_{r,2}(t))$ are auxiliary dual variables. The last equality is a result from [26]. Hence, (3.2) transforms to

$$\inf_{\phi,a} \sup_{\rho,m} \left\{ \frac{1}{2}\int_0^1 \|a(t)\|^2 dt - \int_{\Omega'} \phi(x,0)\rho_0(x)dx - \int_{\Omega'}\int_0^1 (\rho\phi_t + \nu\rho\Delta\phi + m\cdot\nabla\phi)\, dtdx \right.$$

(3.3) $$\left. -\int_{\Omega'}\int_0^1 \left(\frac{1}{2\kappa(x)}\frac{\|m\|^2}{2\rho} + \rho\, a(t)\cdot\zeta(x;\mu,\omega)\right)dtdx - \chi_{\rho\geq 0} + \chi_{\phi(x,1)=g(x)} \right\}$$

$$:= \inf_{\phi,a} \sup_{\rho,m} \left\{ -\mathcal{L}\left((\rho_0,g),(\rho,m),(\phi,a),(\kappa,\mu)\right) \right\},$$

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

3.2. **An inverse mean-field game problem.** Denoting by

$$u = (\rho_0, g), \quad x = (\rho, m), \quad y = (\phi, a), \quad c = (\kappa, \mu),$$

$$F\left((\rho_0,g),(\rho,m),(\phi,a),(\kappa,\mu)\right) = \mathcal{L}\left((\rho_0,g),(\rho,m),(\phi,a),(\kappa,\mu)\right),$$

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 Ω and Ω' as sampling and

computational domains, respectively. An example is shown in Figure 1, where the Ω' is the large square domain, while Ω is the inner square with its boundary highlighted in red.
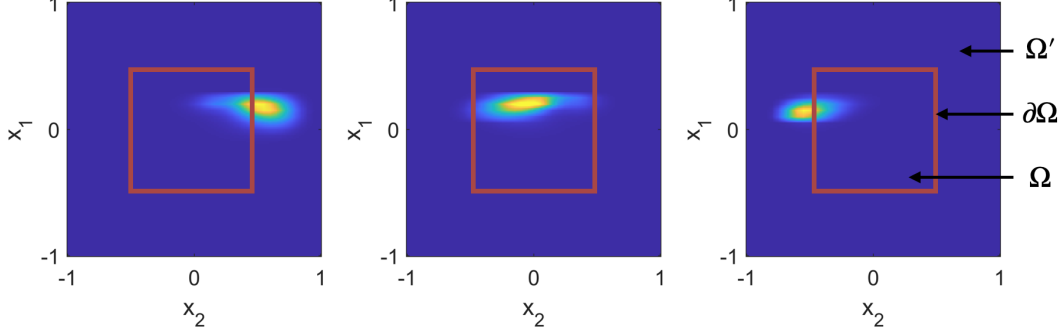

FIGURE 1. Denote ρ as the solution to the mean-field game system. From left to right, the pictures display the density distribution ρ at time $t = 0.1, 0.5, 0.9$. The solid red line represents the boundary of domain Ω. In this mean-filed game, the density travels from the right towards the left, crossing the boundary $\partial\Omega$ twice.

Next, we take

$$\Pi_{B,(\rho,m)}\left(\rho, m\right) := \left(\rho, m \cdot n\right)|_{[0,T]\times\partial\Omega^+}, \quad \Pi_{B,(a,\phi)}(a, \phi) := (0, 0),$$

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 (κ, μ) represent a disturbance of background parameters (κ_0, μ_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

$$R_1(\kappa, \mu) = \tilde{R}_1(\kappa) + \tilde{R}_2(\mu) := \gamma_\kappa \|\kappa - \kappa_0\|_{L^1} + \gamma_\mu \|\mu - \mu_0\|_{L^1},$$
$$R_2(\kappa, \mu) := \chi_{\kappa \geq \varepsilon_1}(\kappa) + \chi_{\mu^2 \leq \varepsilon_2}(\mu).$$

It is also possible to have other choices of regularization for (κ, μ), such as TV, H^1, Wavelet norms.

Now, we can formulate the inverse MFG as follows:

(3.4)
$$\inf_{\substack{\{(\rho_i,m_i),(\phi_i,a_i)\}_{i=1}^N, \\ \kappa,\mu}} \left\{ \sum_{i=1}^N \frac{1}{2}\|\Pi_B(\rho_i, m_i) - \tilde{r}_{B,i}\|^2 + R(\kappa, \mu) : \right.$$

$$\left. (\rho_i, m_i), (\phi_i, a_i) \in \arg\min_{\rho,m} \arg\max_{\phi,a} F((\rho_{0,i}, g_i), (\rho, m), (\phi, a), (\kappa, \mu)) \right\}.$$

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:

$$(4.1) \qquad 0 \;\in\; A(\kappa,\mu) + B(\kappa,\mu) + C\left((\kappa,\mu),((\rho,m),(a,\phi)),(\lambda_{(\rho,m)},\lambda_{(a,\phi)})\right),$$

where

$$A(\kappa,\mu) = \begin{pmatrix} \partial R_1(\kappa,\mu) \\ (0,0) \\ (0,0) \end{pmatrix}, \quad B(\kappa,\mu) = \begin{pmatrix} \partial R_2(\kappa,\mu) \\ (0,0) \\ (0,0) \end{pmatrix},$$

and

$$
\begin{aligned}
&C\left((\kappa,\mu),((\rho,m),(a,\phi)),(\lambda_{(\rho,m)},\lambda_{(a,\phi)})\right) \\
&= \begin{pmatrix} \sum_{i=1}^N \left\langle \partial_{(\kappa,\mu)}\partial_{(\rho_i,m_i)}\mathcal{L}, \lambda_{(\rho_i,m_i)} \right\rangle \\ -\sum_{i=1}^N \left\langle \partial_{(\kappa,\mu)}\partial_{(a_i,\phi_i)}\mathcal{L}, \lambda_{(a_i,\phi_i)} \right\rangle \\ \left(\partial_{(\rho_i,m_i)}\mathcal{L}, -\partial_{(a_i,\phi_i)}\mathcal{L}\right) \\ \left(\partial^2_{(\rho_i,m_i),(\rho_i,m_i)}\mathcal{L}\lambda_{(\rho_i,m_i)} + \Pi^*_{B,(\rho,m)}[\Pi_{B,(\rho,m)}(\rho_i,m_i) - \tilde{r}_{B,i}],\right. \\ \left. -\partial^2_{(a_i,\phi_i),(a_i,\phi_i)}\mathcal{L}\lambda_{(a_i,\phi_i)}\right) \end{pmatrix}.
\end{aligned}
$$

The three-operator splitting scheme in [8] applies to optimization problems of the form

$$(4.2) \qquad \text{find} \quad z \in \mathcal{H} \quad \text{such that} \quad 0 \in Az + Bz + Cz,$$

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

$$(4.3) \qquad \begin{aligned} z^{k+1} &:= (1 - \lambda_k)z^k + \lambda_k T z^k, \\ T &:= I_\mathcal{H} - J_{\gamma B} + J_{\gamma A} \circ (2J_{\gamma B} - I_\mathcal{H} - \gamma C \circ J_{\gamma B}), \end{aligned}$$

where γ 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 $x \mapsto \arg\min_y f_S(y) + \frac{1}{2}\|x - y\|^2$.

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 $\|\cdot\|_{(a,\phi)}$ with $L^2([0,T], H^{-1/2}(\partial\Omega)) \times L^2([0,T], H^{1/2}(\partial\Omega))$ semi-norm, and $L^2_t \times L^2_{x,t}$ norms, respectively. Here the $H^{\frac{1}{2}}(\partial\Omega)$ and

$H^{-\frac{1}{2}}(\partial\Omega)$ semi-norm are taken as follows:

$$|v|^2_{H^{\frac{1}{2}}(\partial\Omega)} \quad := \quad \min_{v_0 \in H^1(\Omega),\, v_0 = v \text{ on } \partial\Omega} |v_0|^2_{H^1(\Omega)}\,,$$

$$|v|^2_{H^{-\frac{1}{2}}(\partial\Omega)} \quad := \quad \min_{v_0 \in H^1_0(\Omega),\, \partial_n v_0 = \partial_n v \text{ on } \partial\Omega} |v_0|^2_{H^1(\Omega)}\,,$$

where the right hand side denotes the standard $H^1(\Omega)$ semi-norm.

Assuming appropriate regularity of (ρ, 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,

$$(\eta_i, \nabla\xi_i) := \Pi^*_{B,(\rho,m)}[\Pi_{B,(\rho,m)}(\rho^n_i, m^n_i) - \tilde{r}_{B,i}],$$

where (η_i, ξ_i) satisfy

$$(4.4) \quad \begin{cases} \begin{cases} \Delta\eta^{n+1}_i = 0 \text{ in } \Omega \\ \eta^{n+1}_i = \rho^n_i - \mathrm{pr}_1\tilde{r}_{B,i} \text{ on } \partial\Omega^+ \\ \Delta\eta^{n+1}_i = 0 \text{ in } \Omega'\backslash\Omega \\ \eta^{n+1}_i = \rho^n_i - \mathrm{pr}_1\tilde{r}_{B,i} \text{ on } \partial\Omega^- \\ \eta^{n+1}_i = 0 \text{ on } \partial\Omega' \end{cases} \quad \begin{cases} \Delta\xi^{n+1}_i = 0 \text{ in } \Omega \\ \partial_n\xi^{n+1}_i = m^n_i \cdot n - \mathrm{pr}_2\tilde{r}_{B,i} \text{ on } \partial\Omega^+ \\ \Delta\xi^{n+1}_i = 0 \text{ in } \Omega'\backslash\Omega \\ \partial_n\xi^{n+1}_i = m^n_i \cdot n + \mathrm{pr}_2\tilde{r}_{B,i} \text{ on } \partial\Omega^- \\ \partial_n\xi^{n+1}_i = 0 \text{ on } \partial\Omega' \end{cases} \end{cases}$$

Here we use $\mathrm{pr}_1, \mathrm{pr}_2$ 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 Ω, the measurements of m and $\nabla\phi$ are equivalent, as $-\kappa_0(x)\rho(x)\partial_n\phi(x) = m(x) \cdot n$ on $\partial\Omega^+$.

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 λ_ϕ is redundant and $\lambda_a = 0$ whenever $0 \in A + B + C$. Hence, we can consider only $C((\kappa,\mu),((\rho,m),(a,\phi)),(\lambda_{(\rho,m)},(0,0)))$. In this case, we preform a primal-dual hybrid gradient method for updating $\lambda_{(\rho,m)}$:

$$(4.5) \quad \begin{cases} \begin{pmatrix} \lambda^{n+1,\text{temp}}_{\rho_i} \\ \lambda^{n+1,\text{temp}}_{m_i} \end{pmatrix} = \\ \left[I + \alpha^n_{\lambda_{(\rho_i,m_i)}}\rho^n_i \begin{pmatrix} \frac{1}{\kappa^n}\frac{|m^n_i|^2}{(\rho^n_i)^3} & -\frac{1}{\kappa^n}\frac{m^n_i}{(\rho^n_i)^2} \\ -\frac{1}{\kappa^n}\frac{m^n_i}{(\rho^n_i)^2} & \frac{1}{\kappa^n\rho^n_i}I \end{pmatrix} \right]^{-1} \left(\begin{pmatrix} \lambda^n_{\rho_i} \\ \lambda^n_{m_i} \end{pmatrix} - \alpha^n_{\lambda_{(\rho_i,m_i)}}\rho^n_i \begin{pmatrix} \eta^{n+1}_i \\ \nabla\xi^{n+1}_i \end{pmatrix} \right) \\ \begin{pmatrix} \lambda^{n+1}_{\rho_i} \\ \lambda^{n+1}_{m_i} \end{pmatrix} = 2 \begin{pmatrix} \lambda^{n+1,\text{temp}}_{\rho_i} \\ \lambda^{n+1,\text{temp}}_{m_i} \end{pmatrix} - \begin{pmatrix} \lambda^{n,\text{temp}}_{\rho_i} \\ \lambda^{n,\text{temp}}_{m_i} \end{pmatrix} \end{cases}$$

4.1.2. *Update of the inverse problem.* In this part, we focus on the update for the inverse problem variables (κ, μ).

(4.6)
$$
\begin{cases}
\begin{cases}
\kappa^{n+1,\text{temp}} &= S_{\alpha_\kappa^n \gamma}\left(2\kappa^n - \tilde{\kappa}^n - \alpha_\kappa^n \sum_{i=1}^N \Lambda_\kappa(\kappa^n, m_i^n, \rho_i^n, \lambda_{\rho_i}^{n+1}, \lambda_{m_i}^{n+1}) - \kappa_0\right) + \kappa_0 \\
\tilde{\kappa}^{n+1} &= \tilde{\kappa}^n + \kappa^{n+1,\text{temp}} - \kappa^n \\
\kappa^{n+1} &= \max\left\{\varepsilon_1, \tilde{\kappa}^{n+1}\right\}
\end{cases} \\
\begin{cases}
\mu^{n+1,\text{temp}} &= S_{\alpha_\mu^n \gamma}\left(2\Pi_\mu^*(\mu^n) - \tilde{\mu}^n - \alpha_\mu^n \sum_{i=1}^N \Lambda_\mu(\lambda_{\rho_i}^{n+1}, a_i^n) - \Pi_\mu^*(\mu_0)\right) + \Pi_\mu^*(\mu_0) \\
\tilde{\mu}^{n+1} &= \tilde{\mu}^n + \mu^{n+1,\text{temp}} - \mu^n \\
\mu^{n+1} &= \Pi_\mu\left(\min\left\{\varepsilon_2, \tilde{\mu}^{n+1}\right\}\right)
\end{cases}
\end{cases}
$$

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

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

$$
\Lambda_\mu(\lambda_\rho, a) = \int_0^T \int_{\Omega'} \Lambda_1(\omega, y)\lambda_\rho(y, s)a(s)dyds,
$$

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

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

$$
\Pi_\mu : \mathbb{R}^{2r} \to C_{\text{odd}=\text{even}} \cong \mathbb{R}^r
$$
$$
(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, ..., x_{r,1}, x_{r,2}) \mapsto \left(\frac{x_{1,1} + x_{1,2}}{2}, \frac{x_{2,1} + x_{2,2}}{2}, ..., \frac{x_{r,1} + x_{r,2}}{2}\right)
$$

and its adjoint as

$$
\Pi_\mu^* : \mathbb{R}^r \to \mathbb{R}^{2r}
$$
$$
(x_1, x_2, ..., x_r) \mapsto (x_1, x_1, x_2, x_2, ..., x_r, x_r).
$$

4.1.3. *Update of the forward problem.* As for the forward problem, we use primal–dual hybrid gradient method (PDHG) [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 (ρ_i, m_i) with stepsizes $(\alpha_{\rho_i}^n, \alpha_{m_i}^n)$; then a proximal gradient ascent step for (ϕ_i, a_i) of stepsizes $(\alpha_{\phi_i}^n, \alpha_{a_i}^n)$; lastly an extrapolating step for (ϕ_i, a_i). Note that we make the choice of norm $\|\phi\|_{H^1_{x,t}}^2 = \|\phi_t\|_{L^2_{x,t}}^2 + \|\nabla_x \phi\|_{L^2_{x,t}}^2$ for ϕ, based on the General-proximal Primal-Dual Hybrid Gradient (G-prox PDHG) 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].

(4.7)

$$
\begin{cases}
(\rho_i^{n+1}, m_i^{n+1}) & = \mathrm{argmin}_{(\rho,m)}\Big\{ \mathcal{L}\Big((\rho_{0,i}, g_i), (\rho, m), (a_i^n, \phi_i^n), (\kappa^{n+1}, \mu^{n+1}) \Big) \\
& \quad + \frac{1}{2\alpha_{\rho_i}^n}\|\rho_i^n - \rho\|_{L_{x,t}^2}^2 + \frac{1}{2\alpha_{m_i}^n}\|m_i^n - m\|_{L_{x,t}^2}^2 \Big\} \\
(\phi_i^{n+1,\mathrm{temp}}, a_i^{n+1,\mathrm{temp}}) & = \mathrm{argmin}_{(a,\phi)}\Big\{ -\mathcal{L}\Big((\rho_{0,i}, g_i), (\rho^{n+1}, m^{n+1}), (a, \phi), (\kappa^{n+1}, \mu^{n+1}) \Big) \\
& \quad + \frac{1}{2\alpha_{\phi_i}^n}\|\phi_i^n - \phi\|_{H_{x,t}^1}^2 + \frac{1}{2\alpha_{a_i}^n}\|a_i^n - a\|_{L_t^2}^2 \Big\} \\
(\phi_i^{n+1}, a_i^{n+1}) & = 2(\phi_i^{n+1,\mathrm{temp}}, a_i^{n+1,\mathrm{temp}}) - (\phi_i^n, a_i^n)
\end{cases}
$$

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

---

    Input: $(\rho_{0,i}, g_i, \tilde{r}_{B,i})$ for $i = 1, ..., N$, (κ_0, μ_0)
    Output: (κ^n, μ^n) for $n = 1, ..., \mathcal{N}_{\max}$
  while iteration $n < \mathcal{N}_{\mathrm{maximal}}$ **do**
    1.Update for the adjoint problem:
      compute $(\lambda_{\rho_i}^{n+1}, \lambda_{m_i}^{n+1,})$ use (4.4)(4.5) for $i = 1, ..., N$.
    2. Update for the inverse problem:
      compute $(\kappa^{n+1}, \mu^{n+1})$ use (4.6)
    3. Update forward problem:
      compute $(\rho_i^{n+1}, m_i^{n+1}, \phi_i^{n+1}, a_i^{n+1})$ use (4.7) for $i = 1, ..., N$.
    $n \leftarrow n + 1$
  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 κ^{n+1} is made from the accumulation of all measurement events (through $(\lambda_{\rho_i}^{n+1}, \lambda_{m_i}^{n+1})$), 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 $(\lambda_{\rho_i}^{n+1}, \lambda_{m_i}^{n+1})$, 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

$$
\begin{cases}
\tilde{\kappa}^{n+1} & = \tilde{\kappa}^n + \mathcal{T}_{mask}(\kappa^{n+1,\mathrm{temp}}, \kappa_0) - \kappa^n \\
\kappa^{n+1} & = \max\{\varepsilon_1, \tilde{\kappa}^{n+1} * \psi\}
\end{cases},
$$

where $\mathcal{T}_{mask}$ is a cut–off function that truncates the change of κ near $\partial\Omega$ given by

$$
\mathcal{T}_{mask}(\kappa, \kappa_0)(x) = \xi(x)(\kappa - \kappa_0)(x) + \kappa_0(x),
$$

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 mean-field game 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 mean-field game 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 mean-field game 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 mean-field game 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 mean-field game 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 mean-field game problem (3.3) with given $(\rho_{0,i}, g_i)$ and (κ, μ) 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 $e_{tol} < 2e-3$. The initial density function $\rho_{0,i}$ is the average of two Gaussian functions with centers $x_G \in \Omega'\backslash\Omega$. The final cost function $g(x)$ is smooth and has a smaller value around a single point $x_g \in \Omega'\backslash\Omega$ such that densities are concentrated in the neighborhood of x_g 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 Ω 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 ρ, m along the boundary $\partial\Omega$ 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:

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

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 ϵ_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 (κ, μ). Recall that we paramatrized the running cost $L(x,v) := \frac{1}{2\kappa(x)}|v|^2$ by κ and nonlocal kernel $K(x,y) := \zeta(x; \mu, \omega) \cdot \zeta(y; \mu, \omega)$ by μ. Since we aim at recovering the model on a given domain with fixed grid points, we fix the choice of ω, and only seek sparse recovery of μ.

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 $\varepsilon_1 = \kappa_c$, 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 $\varepsilon_2 = 1$.

To account for unknown ground truth of the model parameters, we introduce Res to quantify the quality of the reconstructed parameters.

$$\text{Res}^n = \sum_i \int_{\partial\Omega^+} \left(\|\rho_i^n - \text{pr}_1 \tilde{r}_{B,i}\|^2 + \|m_i^n \cdot n - \text{pr}_2 \tilde{r}_{B,i}\|^2 \right),$$

where ρ_i^n, m_i^n are the solution of the forward mean-field game problem with i-th choice of initial density and final cost function with the reconstructed parameter (κ^n, μ^n) at the n-th iteration of the algorithm. The boundary residual Res measures how much the new boundary measurements of the mean-field game model with the recovered parameters deviate from the given partial measurements. If $(\kappa, \mu) = (\kappa_{true}, \mu_{true})$, we would expect that Res is close to 0. Therefore, we pick the reconstructed parameters at n_{opt}-th iteration by taking

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

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 mean-field game 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 $\mu_k^2 \cos(\omega_k \cdot x)$. 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:

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

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

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

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

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

where $x = [x_1, x_2], y = [y_1, y_2]$. We have μ_0, which represents $K_0(x, y)$ via the expansions form $\mu_k^2\cos(\omega_k \cdot x)$, known. The variable k_0 is a given constant value that makes the kernel integration $\int\int K(x, y)dxdy = 1$. 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 Ω'. With $K(x, y) = \zeta(x; \mu, \omega) \cdot \zeta(y; \mu, \omega)$ for $\mu, \omega \in C_{odd=even}$, we omit the even entries (eg. $\mu_{k,2}, \omega_{k,2}$) and express the kernel K_s as follows:

$$\mu_s = (0.2094, 0.2094, 0.2613, 0.2613),$$
$$\omega_s = ((\pi, 0), (0, \pi), (\pi, \pi), (-\pi, \pi)).$$

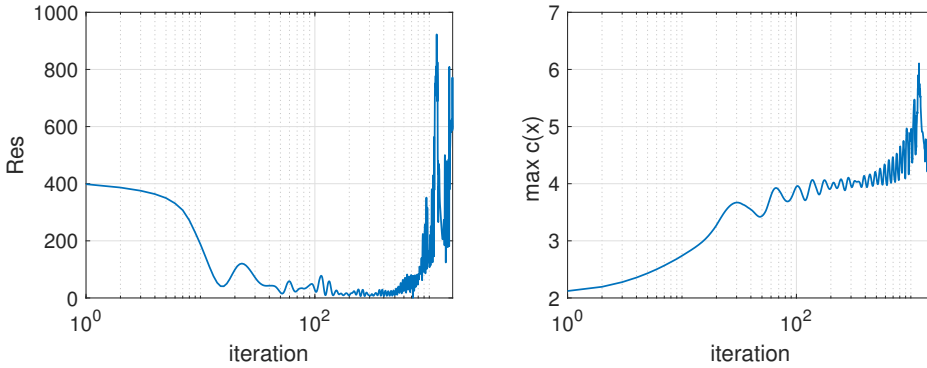Here, we also assume that $\kappa_c = 2$ and $\kappa(x) = 2$ for $x \in \Omega'\backslash\Omega$ is known.

FIGURE 2. The residual Res^n and the $\max_x \kappa^n(x)$ at n-th iteration.

Given (κ_0, μ_0) and the noisy partial boundary measurements with corresponding event parameters $(\rho_{0,i}, g_i, \tilde{r}_{B,i})$, we apply our inverse algorithm. The results are shown in Figure 2,3. In Figure 2, we plot the residual 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 $n_{opt} = \arg\min_n \text{Res}^n$, $(\kappa_{opt}, \mu_{opt}) = (\kappa^{n_{opt}}, \mu^{n_{opt}})$. We see that the reconstructed $\kappa_{opt}(x)$ has a single bump sits near $(0.25, 0.25)$. The shape of the bump is not as sharp as the ground truth κ. The maximal value of running cost $\max_x \kappa_{true}(x) = 6$; while $\max_x \kappa_{opt}(x) = 4$. As for the non-local kernel, we have μ_{opt} nicely reconstructed, where $\mu_{true} = \mu_0 + \mu_s \approx \mu_{opt}$. This example shows that our inverse algorithm is robust to noise and can recover the model parameters (κ, μ) simultaneously.
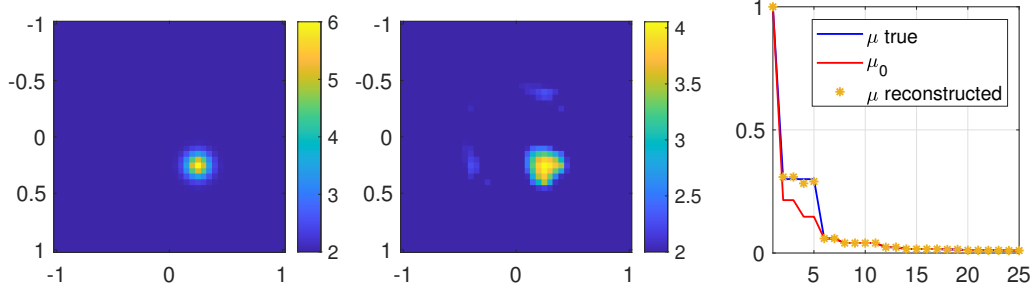
FIGURE 3. From left to right: the true running cost $\kappa(x)$; the reconstructed running cost $\kappa_{opt}(x)$ at iteration n_{opt}; the coefficient representation of non-local kernel $K(x, y)$ in vector form, where x-axis represents different Fourier mode ω and the y-axis corresponds to the coefficients μ.

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:

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

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

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

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

$$\mu_s = (0.3374, 0.3374, 0.2942, 0.2942),$$
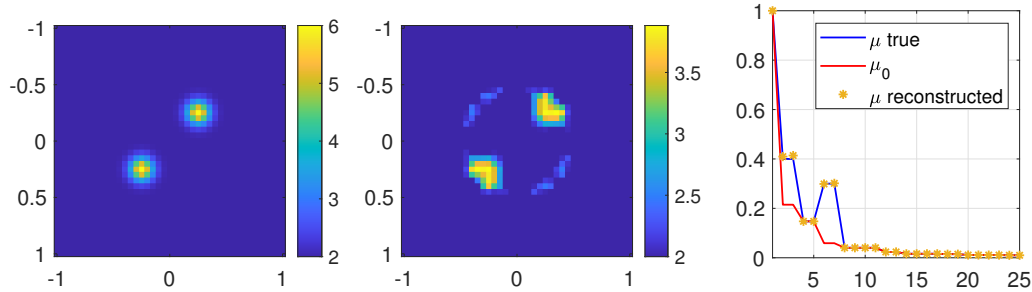
$$\omega_s = ((\pi, 0), (0, \pi), (2\pi, 0), (0, 2\pi)).$$

FIGURE 4. From left to right: the true running cost $\kappa(x)$; the reconstructed running cost $\kappa_{opt}(x)$ at iteration n_{opt}; the coefficient representation of non-local kernel $K(x, y)$ in vector form.

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 Ω'. 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.

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

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

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

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

$$\mu_s = (0.2973, 0.2973, 0.2973, 0.2973),$$

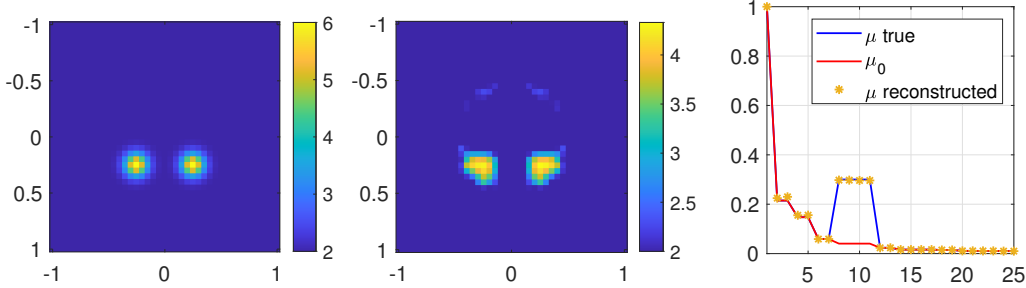$$\omega_s = ((2\pi, -\pi), (2\pi, \pi), (\pi, 2\pi), (\pi, -2\pi)).$$

FIGURE 5. From left to right: the ground true running cost $\kappa(x)$; the reconstructed running cost $\kappa_{opt}(x)$ at iteration n_{opt}; the coefficient representation of nonlocal kernel $K(x, y)$ in vector form.

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 Ω. The nonlocal kernel is recovered very nicely.

6. CONCLUSION

In this paper, we formulate a new class of inverse mean-field game 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.

References

[1] Yves Achdou, Francisco J. Buera, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll. Partial differential equation models in macroeconomics. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 372(2028):20130397, 19, 2014.

[2] Yves Achdou, Pierre Cardaliaguet, François Delarue, Alessio Porretta, and Filippo Santambrogio. *Mean field games*, volume 2281 of *Lecture Notes in Mathematics*. Springer, Cham; Centro Internazionale Matematico Estivo (C.I.M.E.), Florence, [2020] ©2020. Edited by Pierre Cardaliaguet and Alessio Porretta, Fondazione CIME/CIME Foundation Subseries.

[3] Sudhanshu Agrawal, Wonjun Lee, Samy Wu Fung, and Levon Nurbekyan. Random features for high-dimensional nonlocal mean-field games. *Journal of Computational Physics*, page 111136, 2022.

[4] Martin Burger, Jan-Frederik Pietschmann, and Marie-Therese Wolfram. Data assimilation in price formation. *Inverse Problems*, 36(6):064003, may 2020.

[5] Pierre Cardaliaguet and Charles-Albert Lehalle. Mean field game of controls and an application to trade crowding. *Math. Financ. Econ.*, 12(3):335–363, 2018.

[6] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[7] Marco Cirant. Multi-population mean field games systems with neumann boundary conditions. *Journal de Mathématiques Pures et Appliquées*, 103(5):1294–1315, 2015.

[8] Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *Set-valued and variational analysis*, 25(4):829–858, 2017.

[9] Lisang Ding, Wuchen Li, Stanley Osher, and Wotao Yin. A mean field game inverse problem. *arXiv preprint arXiv:2007.11551*, 2020.

[10] Diogo A Gomes, Levon Nurbekyan, and Edgard A Pimentel. *Economic models and mean-field games theory*. IMPA Mathematical Publications. Instituto Nacional de Matemática Pura e Aplicada (IMPA), Rio de Janeiro, 2015.

[11] Diogo A. Gomes, Edgard A. Pimentel, and Vardan Voskanyan. *Regularity theory for mean-field game systems*. SpringerBriefs in Mathematics. Springer, [Cham], 2016.

[12] Howard Heaton, Daniel McKenzie, Qiuwei Li, Samy Wu Fung, Stanley Osher, and Wotao Yin. Learn to predict equilibria via fixed point networks. *arXiv preprint arXiv:2106.00906*, 2021.

[13] M. Huang, P. E. Caines, and R. P. Malhamé. Large-population cost-coupled LQG problems with nonuniform agents: individual-mass behavior and decentralized ϵ-Nash equilibria. *IEEE Trans. Automat. Control*, 52(9):1560–1571, 2007.

[14] M. Huang, R. P. Malhamé, and P. E. Caines. Large population stochastic dynamic games: closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Commun. Inf. Syst.*, 6(3):221–251, 2006.

[15] Matt Jacobs, Flavien Léger, Wuchen Li, and Stanley Osher. Solving large-scale optimization problems with a convergence rate independent of grid size. *SIAM Journal on Numerical Analysis*, 57(3):1100–1123, 2019.

[16] Pushkin Kachroo, Shaurya Agarwal, and Shankar Sastry. Inverse problem for non-viscous mean field control: Example from traffic. *IEEE Transactions on Automatic Control*, 61(11):3412–3421, 2015.

[17] Jean-Michel Lasry and Pierre-Louis Lions. Jeux à champ moyen. I. Le cas stationnaire. *C. R. Math. Acad. Sci. Paris*, 343(9):619–625, 2006.

[18] Jean-Michel Lasry and Pierre-Louis Lions. Jeux à champ moyen. II. Horizon fini et contrôle optimal. *C. R. Math. Acad. Sci. Paris*, 343(10):679–684, 2006.

[19] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Jpn. J. Math.*, 2(1):229–260, 2007.

[20] Mathieu Laurière and Ludovic Tangpi. Convergence of large population games to mean field games with interaction through the controls. *arXiv preprint arXiv:2004.08351*, 2020.

[21] 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.

[22] Alex Tong Lin, Samy Wu Fung, Wuchen Li, Levon Nurbekyan, and Stanley J Osher. Alternating the population and control neural networks to solve high-dimensional stochastic mean-field games. *Proceedings of the National Academy of Sciences*, 118(31), 2021.

[23] Siting Liu, Matthew Jacobs, Wuchen Li, Levon Nurbekyan, and Stanley J. Osher. Computational methods for first-order nonlocal mean field games with applications. *SIAM Journal on Numerical Analysis*, 59(5):2639–2668, 2021.

[24] Siting Liu and Levon Nurbekyan. Splitting methods for a class of non-potential mean field games. *Journal of Dynamics & Games*, 8(4):467–486, 2021.

[25] Zhiyu Liu, Bo Wu, and Hai Lin. A mean field game approach to swarming robots control. In *2018 Annual American Control Conference (ACC)*, pages 4293–4298. IEEE, 2018.

[26] Levon Nurbekyan and J. Saúde. Fourier approximation methods for first-order nonlocal mean-field games. *Port. Math.*, 75(3-4):367–396, 2018.

[27] Xiaochun Rong. A bochner theorem and applications. *Duke mathematical journal*, 91(2):381–392, 1998.

[28] Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

[29] Zuoqiang Shi, Jian Sun, and Minghao Tian. Harmonic extension on the point cloud. *Multiscale Modeling & Simulation*, 16(1):215–247, 2018.

[30] Leonardo Stella and Dario Bauso. Mean-field games for bio-inspired collective decision-making in dynamical networks. *arXiv preprint arXiv:1802.03435*, 2018.

[31] Jiachen Yang, Xiaojing Ye, Rakshit Trivedi, Huan Xu, and Hongyuan Zha. Learning deep mean field games for modeling large population behavior. In *International Conference on Learning Representations (ICLR)*, 2018.

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

We denote by

$$\mathcal{L}(\cdot) := \mathcal{L}\bigg((\rho_{0,i}, g_i), (\rho_i, m_i), (a_i, \phi_i), (\kappa, \mu)\bigg),$$

the function $\mathcal{L}$ defined in Equation 3.3 for simplicity. The KKT conditions for the inverse mean-field game problem are then

$$\begin{cases} \Pi^*_{B,(\rho,m)}[\Pi_{B,(\rho,m)}(\rho_i, m_i) - \tilde{r}_{B,i}] + \partial^2_{(\rho_i,m_i),(\rho_i,m_i)}\mathcal{L}(\cdot)\lambda_{(\rho_i,m_i)} = 0, \quad i = 1, ..., N, \\ \Pi^*_{B,(a,\phi)}[\Pi_{B,(a,\phi)}(a_i, \phi_i) - \tilde{s}_{B,i}] - \partial^2_{(a_i,\phi_i),(a_i,\phi_i)}\mathcal{L}(\cdot)\lambda_{(a_i,\phi_i)} = 0, \quad i = 1, ..., N, \\ \partial_{(\kappa,\mu)}R_1(\kappa, \mu) + \partial_{(\kappa,\mu)}R_2(\kappa, \mu) + \sum_{i=1}^N \left\langle \partial_{(\kappa,\mu)}\partial_{(\rho_i,m_i)}\mathcal{L}(\cdot), \lambda_{(\rho_i,m_i)} \right\rangle, \\ -\sum_{i=1}^N \left\langle \partial_{(\kappa,\mu)}\partial_{(a_i,\phi_i)}\mathcal{L}(\cdot), \lambda_{(a_i,\phi_i)} \right\rangle = 0, \\ \partial_{(\rho_i,m_i)}\mathcal{L}(\cdot) = 0, \quad i = 1, ..., N, \\ -\partial_{(a_i,\phi_i)}\mathcal{L}(\cdot) = 0, \quad i = 1, ..., N. \end{cases}$$

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

$$\partial_{(\rho_i,m_i)}\mathcal{L} = \left(\partial_t\phi + \nu\Delta\phi - \frac{1}{2\kappa}\frac{|m|^2}{\rho^2} + a \cdot \zeta(\cdot; \mu, \omega), \nabla\phi + \frac{1}{\kappa}\frac{m}{\rho} \right)$$

$$\partial_{(a_i,\phi_i)}\mathcal{L} = \left(-a + \int_{\Omega'} \rho(y)\,\zeta(y; \mu, \omega)dx, -\partial_t\rho + \nu\Delta\rho - \nabla \cdot m \right)$$

$$\partial^2_{(\rho_i,m_i),(\rho_i,m_i)}\mathcal{L} = \begin{pmatrix} \frac{1}{\kappa}\frac{|m|^2}{\rho^3} & -\frac{1}{\kappa}\frac{m}{\rho^2} \\ -\frac{1}{\kappa}\frac{m}{\rho^2} & \frac{1}{\kappa\rho}I \end{pmatrix}$$

$$\partial^2_{(a_i,\phi_i),(a_i,\phi_i)}\mathcal{L} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}$$

$$\partial_{(\kappa,\mu)}\partial_{(\rho_i,m_i)}\mathcal{L} = \begin{pmatrix} \frac{1}{2\kappa^2(x)}\frac{|m|^2}{\rho^2} & -\frac{1}{\kappa^2(x)}\frac{m}{\rho} \\ \Lambda_1(\omega, \cdot)\,a & 0 \end{pmatrix}$$

$$\partial_{(\kappa,\mu)}\partial_{(a_i,\phi_i)}\mathcal{L} = \begin{pmatrix} 0 & 0 \\ \int_{\Omega'} \rho(y)\,\Lambda_1(\omega, y)dy & 0 \end{pmatrix}$$

where the variable $\Lambda_1(\omega, x)$ is defined in Section 4.1.2.

APPENDIX APPENDIX B   THE INVERSE ALGORITHM

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

---

Input: $(\rho_{0,i}, g_i, \tilde{r}_{B,i})$ for $i = 1, ..., N$, (κ_0, μ_0)

Output: (κ^n, μ^n) for $n = 1, ..., \mathcal{N}_{\max}$

while iteration $n < \mathcal{N}_{\text{maximal}}$ **do**

1.Update for the adjoint problem by computing $(\lambda_{\rho_i}^{n+1}, \lambda_{m_i}^{n+1,})$

$$
\begin{cases}
\begin{cases}
\Delta \eta_i^{n+1} = 0 \text{ in } \Omega \\
\eta_i^{n+1} = \rho_i^n - \text{pr}_1 \tilde{r}_{B,i} \text{ on } \partial\Omega^+ \\
\Delta \eta_i^{n+1} = 0 \text{ in } \Omega' \backslash \Omega \\
\eta_i^{n+1} = \rho_i^n - \text{pr}_1 \tilde{r}_{B,i} \text{ on } \partial\Omega^- \\
\eta_i^{n+1} = 0 \text{ on } \partial\Omega'
\end{cases} \\
\begin{cases}
\Delta \xi_i^{n+1} = 0 \text{ in } \Omega \\
\partial_n \xi_i^{n+1} = m_i^n \cdot n - \text{pr}_2 \tilde{r}_{B,i} \text{ on } \partial\Omega^+ \\
\Delta \xi_i^{n+1} = 0 \text{ in } \Omega' \backslash \Omega \\
\partial_n \xi_i^{n+1} = m_i^n \cdot n + \text{pr}_2 \tilde{r}_{B,i} \text{ on } \partial\Omega^- \\
\partial_n \xi_i^{n+1} = 0 \text{ on } \partial\Omega'
\end{cases} \\
\begin{cases}
\begin{pmatrix} \lambda_{\rho_i}^{n+1,\text{temp}} \\ \lambda_{m_i}^{n+1,\text{temp}} \end{pmatrix} = \\
\left[I + \alpha_{\lambda_{(\rho_i,m_i)}}^n \rho_i^n \begin{pmatrix} \frac{1}{\kappa^n} \frac{|m_i^n|^2}{(\rho_i^n)^3} - \frac{1}{\kappa^n} \frac{m_i^n}{(\rho_i^n)^2} \\ -\frac{1}{\kappa^n} \frac{m_i^n}{(\rho_i^n)^2} \frac{1}{\kappa^n \rho_i^n} I \end{pmatrix} \right]^{-1} \left(\begin{pmatrix} \lambda_{\rho_i}^n \\ \lambda_{m_i}^n \end{pmatrix} - \alpha_{\lambda_{(\rho_i,m_i)}}^n \rho_i^n \begin{pmatrix} \eta_i^{n+1} \\ \nabla\xi_i^{n+1} \end{pmatrix} \right) \\
\begin{pmatrix} \lambda_{\rho_i}^{n+1} \\ \lambda_{m_i}^{n+1} \end{pmatrix} = 2 \begin{pmatrix} \lambda_{\rho_i}^{n+1,\text{temp}} \\ \lambda_{m_i}^{n+1,\text{temp}} \end{pmatrix} - \begin{pmatrix} \lambda_{\rho_i}^{n,\text{temp}} \\ \lambda_{m_i}^{n,\text{temp}} \end{pmatrix}
\end{cases}
\end{cases}
$$

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

$$
\begin{cases}
\begin{cases}
\kappa^{n+1,\text{temp}} &= S_{\alpha_\kappa^n \gamma}\left(2\kappa^n - \tilde{\kappa}^n - \alpha_\kappa^n \sum_{i=1}^N \Lambda_\kappa(\kappa^n, m_i^n, \rho_i^n, \lambda_{\rho_i}^{n+1}, \lambda_{m_i}^{n+1}) - \kappa_0 \right) + \kappa_0 \\
\tilde{\kappa}^{n+1} &= \tilde{\kappa}^n + \mathcal{T}_{mask}(\kappa^{n+1,\text{temp}}, \kappa_0) - \kappa^n \\
\kappa^{n+1} &= \max\left\{ \varepsilon_1, \tilde{\kappa}^{n+1} * \psi \right\}
\end{cases} \\
\begin{cases}
\mu^{n+1,\text{temp}} &= S_{\alpha_\mu^n \gamma}\left(2\Pi_\mu^*(\mu^n) - \tilde{\mu}^n - \alpha_\mu^n \sum_{i=1}^N \Lambda_\mu(\lambda_{\rho_i}^{n+1}, a_i^n) - \Pi_\mu^*(\mu_0) \right) + \Pi_\mu^*(\mu_0) \\
\tilde{\mu}^{n+1} &= \tilde{\mu}^n + \mu^{n+1,\text{temp}} - \mu^n \\
\mu^{n+1} &= \Pi_\mu(\min\{\varepsilon_2, \tilde{\mu}^{n+1}\})
\end{cases}
\end{cases}
$$

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 $i = 1, ..., N$.

Apply the iterative Algorithm 3 given input $(\rho_{0,i}, g_i)$, $(\kappa^{n+1}, \mu^{n+1})$, e_{tol} with initial guess $(\rho_i^n, m_i^n, \phi_i^n, a_i^n)$, and assign

$$(\rho_i^{n+1}, m_i^{n+1}, \phi_i^{n+1}, a_i^{n+1}) := (\rho_i^*, m_i^*, \phi_i^*, a_i^*)$$

$n \leftarrow n + 1$

end while

APPENDIX APPENDIX C   THE FORWARD ALGORITHM FOR NONLOCAL MEAN-FIELD GAMES

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

---

Input: (ρ_0, g), (κ, μ), a set of initial guess $(\rho^0, m^0, \phi^0, a^0)$, e_{tol}, a set of stepsizes $(\alpha_\rho^j, \alpha_m^j, \alpha_\phi^j, \alpha_a^j)$

Output: $(\rho^*, m^*, \phi^*, a^*)$

while iteration $j < \mathcal{J}_{\max}$ and primal-dual gap $\mathrm{PD}(\rho^j, m^j, \phi^j, a^j) \geq e_{tol}$ **do**

$$
\begin{cases}
(\rho^{j+1}, m^{j+1}) & = \mathrm{argmin}_{(\rho,m)}\Big\{ \mathcal{L}\Big((\rho_0, g), (\rho, m), (a^j, \phi^j), (\kappa, \mu) \Big) \\
& \quad + \frac{1}{2\alpha_\rho^j}\|\rho^j - \rho\|_{L_x^2, t}^2 + \frac{1}{2\alpha_m^j}\|m^j - m\|_{L_x^2, t}^2 \Big\} \\
(\phi^{j+1,\text{temp}}, a^{j+1,\text{temp}}) & = \mathrm{argmin}_{(a,\phi)}\Big\{ -\mathcal{L}\Big((\rho_0, g), (\rho^{n+1}, m^{n+1}), (a, \phi), (\kappa, \mu) \Big) \\
& \quad + \frac{1}{2\alpha_\phi^j}\|\phi^j - \phi\|_{H_{x,t}^1}^2 + \frac{1}{2\alpha_a^j}\|a^j - a\|_{L_t^2}^2 \Big\} \\
(\phi^{j+1}, a^{j+1}) & = 2(\phi^{j+1,\text{temp}}, a^{j+1,\text{temp}}) - (\phi^j, a^j)
\end{cases}
$$

$\quad j \leftarrow j + 1$

end while

return $(\rho^j, m^j, \phi^j, a^j)$