Learning operators on labelled conditional distributions with applications to mean field control of non exchangeable systems

Samy Mekkaoui

Ecole Polytechnique, CMAP

Based on joint work with
Huyên PHAM Xavier WARIN

Stochastic Control and Games for Risk and Regulation

Hammamet, Tunisia, April 27-30, 2026

Outline

1. From exchangeable to heterogeneous mean-field models
2. Operator learning on a constrained Wasserstein space
3. Application to non-exchangeable mean field control

Motivation: large population of interacting agents

- In classical setting of interaction players/particles, one assumes
  - Homogeneous interaction $\leftrightarrow$ **exchangeability**
  - In the large population limit $\rightarrow$ one **single marginal law** μ_t describing law of the **representative agent**

$\rightarrow$ Mean-Field Game (MFG)/Mean-Field Control (MFC)

$\rightarrow$ Problem defined on Wasserstein space of probability measure

Emphasis

Exchangeability compresses the system into one law

Beyond exchangeability: heterogeneous population

- Instead of homogeneous interaction, consider **weighted interaction** e.g. via a graphon
- Labels / types / spatial indices
- Interaction depends on the **label** $u \in I$
- population described by $u \in I \mapsto \mu_t^u \in \mathcal{P}(\mathbb{R}^d)$

$\rightarrow$ The right state variable is no longer one law, but a labelled family of laws.

Some references: Caines, Huang (20), Jabin, Polato, Soler (21), Aurell, Carmona, Laurière (22), Bayraktar, Chakraporty, Wu (23), Lacker-Soret (23), Coppini, De Crescenzo, Pham (24), Crucianelli-Tangpi (24), etc

Goal of this work: develop numerical approach for problems (MFG/MFC) in the **non exchangeable** setting.

A constrained Wasserstein space

- I: compact label space, typically $[0,1]$
- λ: reference distribution of labels, e.g. uniform distribution
- admissible population laws keep the first marginal fixed

$$\mathcal{M}_\lambda = \{\mu \in \mathcal{P}_2(I \times \mathbb{R}^d) : \mathrm{pr}_1 \# \mu = \lambda\}.$$

By disintegration:

$$\mu(\mathrm{d}u, \mathrm{d}x) = \lambda(\mathrm{d}u)\, \mu^u(\mathrm{d}x).$$

Interpretation

The label marginal is fixed; only the conditional state laws vary. Equivalently, μ is a labelled family $(\mu_u)_{u \in I}$.

What operator are we trying to learn?

We study operators of the form

$$\vartheta : \mu \in \mathcal{M}_\lambda \longmapsto V(\cdot, \cdot, \mu) \in L^2(\mu).$$

for some $\mathbb{R}^q$-valued function V defined on $I \times \mathbb{R}^d \times \mathcal{M}_\lambda$. Typical outputs:

- Value function
- Feedback control.

Key issue

The input is infinite-dimensional and constrained by the fixed label marginal.

Main approximation question

Can we approximate continuous operators on $\mathcal{M}_\lambda$ by a trainable finite-dimensional architecture?

- respect the marginal constraint,
- compress the measure input,
- retain a universal approximation property.

Cylindrical features

Choose test functions $\phi_1, \dots, \phi_J \in C(I \times \mathbb{R}^d)$ and define

$$\mu \in \mathcal{M}_\lambda \longmapsto \Phi_J(\mu) = \big(\langle \phi_1, \mu \rangle, \dots, \langle \phi_J, \mu \rangle\big) \in \mathbb{R}^J.$$

- finite-dimensional summary of the measure input, e.g. moments:

$$\phi_j(u, x) = |x|^j + u^j, \quad j = 1, \dots, J.$$

- enough to separate measures on compact subsets in the proof.

DeepONetCyl architecture

Inspired by DeepOnet for operators on function (Karniadakis et al. 19), we consider:

$$(u, x, \mu) \longmapsto \sum_{k=1}^{r} \mathcal{T}_k(u, x)\, \mathcal{B}_k(\Phi_J(\mu)).$$

- trunk $\mathcal{T}_k : I \times \mathbb{R}^d \to \mathbb{R}$: local dependence on label and state,
- branch $\mathcal{B}_k : \mathbb{R}^J \to \mathbb{R}^q$: global dependence on the labelled law
- $k = 1, \ldots, r$ number of sensors

Interpretation

This is a finite-rank neural operator adapted to the measure space $\mathcal{M}_\lambda$.

Universal approximation theorem

Let ρ be a probability measure on $\mathcal{M}_\lambda$. For every continuous V with finite $L^2(\rho)$ norm and every $\varepsilon > 0$, there exist J, r, test functions $(\phi_j)_{1\leq j\leq J}$, trunk nets $(\mathcal{T}_k)_{1\leq k\leq r}$, and branch nets $(\mathcal{B}_k)_{1\leq k\leq r}$ such that

$$\int_{\mathcal{M}_\lambda} \mathbb{E}_{(U,X)\sim\mu}\left[\left|V(U,X,\mu) - \sum_{k=1}^{r} \mathcal{T}_k(U,X)\mathcal{B}_k(\Phi_J(\mu))\right|^2\right]\rho(\mathrm{d}\mu) \leq \varepsilon.$$

Meaning

Continuous operators on $\mathcal{M}_\lambda$ can be approximated in the same metric used for training.

Proof strategy in three steps

1. Cylindrical observables separate measures
2. Finite sums $f(\Phi_J(\mu))\,g(u,x)$ are dense by Stone–Weierstrass
3. Neural networks approximate the finite-dimensional factors

Core idea

Once the right observables are chosen, the proof reduces to a classical density argument plus finite-dimensional neural approximation.

Sampling training data in $\mathcal{M}_\lambda$

Take a non-atomic reference law ν and sample $(U, Y) \sim \lambda \otimes \nu$. Construct

$$\mu = \mathcal{L}(U, T(U, Y))$$

with $T(u, \cdot)\#\nu = \mu^u$. Two strategies:

- **S1**: randomize T, keep ν fixed,
- **S2**: fix T, randomize the base law ν.

For sampled measures $\mu^{(m)}$, minimize

$$\frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{(U,X)\sim\mu^{(m)}}\Big[\big|V(U, X, \mu^{(m)}) - \mathrm{DeepONetCyl}(U, X, \mu^{(m)})\big|^2\Big].$$

In the experiments, the cylindrical features are moment-type maps.

Remark

The marginal constraint is enforced by construction.

Non-exchangeable mean field system

- Controlled state equation:

$$\mathrm{d}X_t \quad = \quad b(U, X_t, \mathbb{P}_{(U,X_t)}, \alpha_t)\,\mathrm{d}t + \sigma(U, X_t, \mathbb{P}_{(U,X_t)}, \alpha_t)\,\mathrm{d}W_t,$$

where label $U \sim \lambda$.

- Cost functional:

$$J(\alpha) \quad = \quad \mathbb{E}\Big[\int_0^T f(U, X_t, \mathbb{P}_{(U,X_t)}, \alpha_t)\,\mathrm{d}t + g(U, X_T, \mathbb{P}_{(U,X_T)})\Big].$$

Bridge with the first part

Value functions, decoupling fields, and feedback maps in heterogeneous MFC are precisely operators on $\mathcal{M}_\lambda$.

Two routes to optimal control

- **Pontryagin maximum principle:** Kharroubi, Mekkaoui, Pham (25), and Cao, Laurière (25)
  - Hamiltonian minimization:

$$\hat{\alpha}_t = \inf_{a \in A} H(U, X_t, \mathbb{P}_{(U,X_t)}, Y_t, Z_t, a),$$

  - coupled FBSDE of MKV type with derivatives with respect to the labelled law:

$$\begin{cases} \mathrm{d}X_t &= b(U, X_t, \mathbb{P}_{(U,X_t)}, \hat{\alpha}_t)\mathrm{d}t + \sigma(U, X_t, \mathbb{P}_{(U,X_t)}, \hat{\alpha}_t)\mathrm{d}W_t, \\ \mathrm{d}Y_t &= -\partial_x H(U, X_t, \mathbb{P}_{(U,X_t)}, Y_t, Z_t, \hat{\alpha}_t)\mathrm{d}t - \tilde{\mathbb{E}}\big[\partial_{\tilde{x}} \frac{\delta}{\delta m} H(\tilde{U}, \tilde{X}_t, \mathbb{P}_{(U,X_t)}, \tilde{Y}_t, \tilde{Z}_t, \tilde{\hat{\alpha}}_t)(U, X_t)\big]\mathrm{d}t \\ & \quad + Z_t \mathrm{d}W_t \\ Y_T &= \partial_x g(U, X_T, \mathbb{P}_{(U,X_T)}) + \tilde{\mathbb{E}}\Big[\partial_{\tilde{x}} \frac{\delta}{\delta m} g(\tilde{U}, \tilde{X}_T, \mathbb{P}_{(U,X_T)})(U, X_T)\Big], \end{cases}$$

- **Dynamic programming**: De Crescenzo, Fuhrman, Kharroubi, Pham (25)
  - Bellman equation on $\mathcal{M}_\lambda$
  - Value function / master field depends on labelled conditional distributions

Benchmark: LQ graphon MFC

$$\mathrm{d}X_t \quad = \Big[A(U) + B(U)X_t + \tilde{\mathbb{E}}\big[G_B(U,\tilde{U})\tilde{X}_t\big] + C(U)\alpha_t\Big]\mathrm{d}t + \sigma(U)\mathrm{d}W_t$$

$$J(\alpha) = \mathbb{E}\Big[\int_0^T \Big(Q(U)\big(X_t - \tilde{\mathbb{E}}[\tilde{G}_Q(U,\tilde{U})\tilde{X}_t]\big)\cdot\big(X_t - \tilde{\mathbb{E}}[\tilde{G}_Q(U,\tilde{U})\tilde{X}_t]\big) + \alpha_t^{\intercal}N(U)\alpha_t\Big)\mathrm{d}t$$
$$+ H(U)\big(X_T - \tilde{\mathbb{E}}[\tilde{G}_H(U,\tilde{U})\tilde{X}_T]\big)\cdot\big(X_T - \tilde{\mathbb{E}}[\tilde{G}_H(U,\tilde{U})\tilde{X}_T]\big)\Big].$$

→ Optimal control (De Crescenzo, De Feo, Pham (25)) :

$$\hat{\alpha}_t = S_t(U)X_t + \tilde{\mathbb{E}}\Big[\bar{S}_t(U,\tilde{U})\tilde{X}_t\Big] + \Gamma_t(U), \quad 0 \le t \le T,$$

with

$$\begin{cases} S_t(U) & = -(N(U))^{-1}(C(U))^{\intercal}K_t(U), \\ \bar{S}_t(U,\tilde{U}) & = -(N(U))^{-1}(C(U)^{\intercal}\bar{K}_t(U,\tilde{U}) \\ \Gamma_t(U) & = -(N(U))^{-1}(C(U))^{\intercal}\Lambda_t(U) \end{cases}$$

where $K \in C^1\big([0,T]; L^{\infty}(I;\mathbb{S}^d_+)\big)$, $\bar{K} \in C^1\big([0,T], L^2(I\times I;\mathbb{R}^{d\times d})\big)$ and $\Lambda \in C^1\big([0,T]; L^2(I;\mathbb{R}^d)\big)$ are to be determined through infinite dimensional Riccati equations.

→ Riccati solver: sample $(u_i,\tilde{u}_j) \sim \lambda\otimes\lambda$ and compute $K_t(u_i)$ and $\bar{K}_t(u_i,u_j)$ by time discretization

Two implemented algorithms

1. **Deep graphon**: parametrize directly the control by a DeepOnetCyl and plug in the cost functional $\rightarrow$ trained by stochastic gradient descent
2. **Deep BSDE graphon** for solving the FBSDE from maximum principle: DeepOnetCyl for the initial value Y_0 and the Z component that are trained by minimizing the quadratic loss from the terminal condition of the BSDE.

Remark: Other possible algorithms can be implemented

Numerical experiments in the example of a systemic risk example with heterogeneous banks (extension of Carmona, Fouque, Sun 05).

Optimal trajectories: comparison with Riccati benchmark

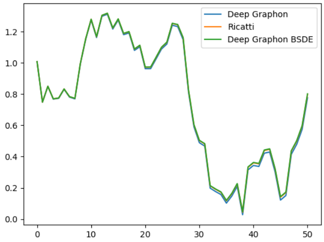

Figure: Optimal trajectory of X with $u = 0.708$

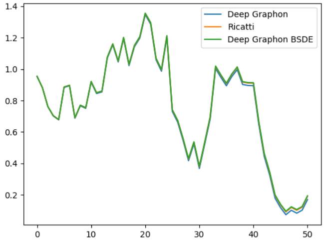

Figure: Optimal trajectory of X with $u = 0.599$

Figure: Comparison between NN solvers and the Riccati one with $G(u, v) = e^{-uv}$

Takeaway

Both neural solvers reproduce well the Riccati benchmark at the trajectory level.

Conclusion

- Heterogeneous mean-field systems naturally live on the constrained space $\mathcal{M}_\lambda$
- This leads to learning operators on labelled conditional laws
- A cylindrical DeepONet gives a practical architecture with universal approximation
- The framework extends neural MFC tools beyond exchangeability