MVNN: A MEASURE-VALUED NEURAL NETWORK FOR LEARNING MCKEAN-VLASOV DYNAMICS FROM PARTICLE DATA

Liyao Lyu
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90024, USA
lyuliyao@math.ucla.edu

Xinyue Yu
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90024, USA
tracy@math.ucla.edu

Hayden Schaeffer
Department of Mathematics
University of California, Los Angeles
Los Angeles, CA 90024, USA
hayden@math.ucla.edu

ABSTRACT

Collective behaviors that emerge from interactions are fundamental to numerous biological systems. To learn such interacting forces from observations, we introduce a measure-valued neural network that infers measure-dependent interaction (drift) terms directly from particle-trajectory observations. The proposed architecture generalizes standard neural networks to operate on probability measures by learning cylindrical features, using an embedding network that produces scalable distribution-to-vector representations. On the theory side, we establish well-posedness of the resulting dynamics and prove propagation-of-chaos for the associated interacting-particle system. We further show universal approximation and quantitative approximation rates under a low-dimensional measure-dependence assumption. Numerical experiments on first and second order systems, including deterministic and stochastic Motsch-Tadmor dynamics, two-dimensional attraction-repulsion aggregation, Cucker-Smale dynamics, and a hierarchical multi-group system, demonstrate accurate prediction and strong out-of-distribution generalization.

***K*eywords** mean field approximation · interacting particle models · data-driven modeling · McKean-Valsov SDE · Wasserstein space · approximation theory

1 Introduction

Interacting particle and agent systems are widely used to model collective dynamics in physics, biology, and the social sciences [1–5]. Since first-principles approaches to modeling interacting forces are often challenging, a key problem is to infer the underlying interaction mechanisms from trajectory observations. Recent data-driven methods address this by assuming a binary-interaction ansatz and estimating a pairwise interaction kernel via least-squares fitting of observed (finite-difference) velocities as a function of pairwise distances [6–11]. However, the pairwise assumption is often insufficient for complex, realistic systems. In many biological, social, and physical scenarios, the effective dynamics are governed by a drift of the mean field form, a nonlinear functional of the population distribution, rather than a

superposition of two-body forces. Popular examples include crowd dynamics governed by local density constraints rather than individual repulsion [12], vehicular traffic where speed is a nonlinear functional of the flow density [13], and cell migration influenced by chemical concentration fields [14]. Restricting the hypothesis class to pairwise interaction kernels can therefore limit the ability to capture emergent collective behaviors. Beyond modeling fidelity, computational efficiency is another major difficulty since direct simulation of interacting particle systems typically scales as $O(N^2)$ in the number of particles. Mean-field limits and related numerical techniques, such as the random batch method [15, 16], provide scalable approximations for large populations [17–19] and have been used in aggregation and swarming, Motsch-Tadmor dynamics, and optimization [20–28]. This motivates the development of drift models whose evaluation scales linearly with N, by utilizing the symmetry arising from agent indistinguishability.

In recent years, machine learning and optimization-based methods have attracted growing attention for discovering and modeling governing equations of ODEs [29–33], SDEs [34–39], and PDEs [40–49]; see also [50]. Extending these techniques to mean-field limits remains challenging because the drift is an operator on the space of probability measures rather than a function of finite-dimensional states. Neural operators such as DeepONet [41] and FNO [42] provide powerful tools for learning mappings between function spaces. More broadly, recent developments include graph neural operators [51], kernel-based operator learning frameworks [52, 53], discretization-free operators [54], and attention-based operator models [55], further advancing operator learning in infinite-dimensional settings. However, these approaches are typically designed for function-valued inputs defined on structured domains, and are not inherently suited to the Wasserstein setting considered here, where the input is a discrete empirical measure represented as an unordered point cloud. The emerging topic of PDE foundation models have been proposed to learn unified latent representations across multiple families of PDE datasets in a single, shared framework [55–68]. By capturing common structure across a wide range of systems, these models enable improved generalization to unseen regimes and more robust transfer across tasks. While these methods are applicable to a wide range of PDEs [55], their utilization for particle dynamics is limited. To the best of our knowledge, there are currently no end-to-end frameworks that can infer such measure-dependent drifts directly from particle trajectories while providing theoretical guarantees. To address these challenges, we propose a data-driven framework to learn the measure-dependent drift in mean-field dynamics directly from particle observation.

Unlike ordinary or stochastic differential equations, where the evolution of each particle depends explicitly on its relative positions or pairwise distances, the evolution of particles (or agents) in the mean-field limit depends only on the distribution of the system. The resulting dynamics take the form of a McKean–Vlasov stochastic differential equation for particles $(\mathbf{X}_t)$:

$$\mathrm{d}\mathbf{X}_t = \mathbf{b}(\mathbf{X}_t, f_t)\mathrm{d}t + \sigma(\mathbf{X}_t, f_t)\mathrm{d}\mathbf{B}_t,$$

where $f_t = \mathcal{L}(\mathbf{X}_t)$ describes the law of the $\mathbf{X}_t$ and $\mathbf{B}_t$ is a d-dimensional Wiener process. Numerically, the McKean–Vlasov dynamics can be approximated by a system of interacting particles $(\mathbf{X}_t^{i,N})_{1\le i\le N}$ with a sufficiently large number of particles N. Formally, the initial condition $(\mathbf{X}_0^{i,N})_{1\le i\le N}$ are independent and identically distributed with law μ_0 and each particle evolves according to:

$$\mathrm{d}\mathbf{X}_t^{i,N} = \mathbf{b}(\mathbf{X}_t^{i,N}, \mu_t^N)\mathrm{d}t + \sigma(\mathbf{X}_t^{i,N}, \mu_t^N)\mathrm{d}\mathbf{B}_t^{i,N}$$

where $\mathbf{B}_t^{i,N}$ are d-dimensional Wiener processes, $\mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{\mathbf{X}_t^{j,N}}$ denotes the empirical measure of the particle system, and δ is the Dirac measure. For simplicity, we only focus on the drift term $\mathbf{b} : \mathbb{R}^d \times \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}^d$ in this work, where $\mathcal{P}(\mathbb{R}^d)$ is the space of probability measures on $\mathbb{R}^d$. The diffusion term σ is treated as a known constant or omitted in the present work. Its extension to state- and measure-dependent cases will be explored in future work.

To represent such measure-dependent functions, we propose a *measure-valued neural network* (MVNN), which extends conventional neural networks to take probability measures as inputs in a permutation-invariant manner. The architecture is motivated by the cylindrical functional framework [69, 70], which characterizes functionals on spaces of probability measures via finitely many test-function integrals. We generalize this idea by learning the test functions and their subsequent interaction map with neural networks, yielding a scalable model that preserves permutation invariance and naturally extends to multi-group interactions.

We summarize our contributions as follows:

- We propose MVNN, a permutation-invariant architecture for learning measure-dependent drifts $b(X, \mu)$ directly from particle-trajectory data. In this setting, permutation-invariant means that the representation depends only on the empirical measure, i.e., on the unordered collection of particle states, so reordering the particles does not change the aggregated feature or the resulting drift.
- We establish well-posedness of the neural network induced McKean-Vlasov dynamics and prove propagation-of-chaos guarantees for the corresponding learned particle system.

- We prove a universal approximation result for MVNNs on $\mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d)$ and establish approximation rates under a low-dimensional (order-parameter) measure-dependence assumption.
- We demonstrate accurate forward simulations and out-of-distribution generalization on first and second order systems, including deterministic/stochastic Motsch-Tadmor dynamics, 2D attraction-repulsion aggregation, Cucker-Smale dynamics, and a hierarchical multi-group system.

2 Learning McKean–Vlasov Drifts from Particle Data

We consider a standard interacting particle/agent system as a motivating example. Note that the learning framework developed here does not rely on the pairwise interaction kernel and can be applied to more complex interactions. Consider N agents with states $\mathbf{X}_t^i \in \mathbb{R}^d$ and the (possibly stochastic) dynamics:

$$\mathrm{d}\mathbf{X}_t^i = \frac{1}{N}\sum_{j=1}^{N} \phi\Big(\|\mathbf{X}_t^j - \mathbf{X}_t^i\|\Big)\,\big(\mathbf{X}_t^j - \mathbf{X}_t^i\big)\,\mathrm{d}t + \sigma\,\mathrm{d}\mathbf{B}_t^i, \qquad i = 1,\dots,N, \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm, $\phi : \mathbb{R}_+ \to \mathbb{R}$ is the pairwise interaction kernel, and $\{\mathbf{B}_t^i\}_{i=1}^N$ are independent d-dimensional Wiener processes (the deterministic case corresponds to $\sigma = 0$). Let $\mathbf{X}_t = (\mathbf{X}_t^1, \dots, \mathbf{X}_t^N)$ and denote the empirical measure by:

$$\mu_t^N := \frac{1}{N}\sum_{j=1}^{N} \delta_{\mathbf{X}_t^j}.$$

In the mean-field limit, such interactions can be represented through a drift $\mathbf{b}(\mathbf{x}, \mu)$ acting on a single state $\mathbf{x}$ and the population distribution μ [71]. For instance, following Section 2.2.1 in [72], for the pairwise model in (1), the corresponding drift is

$$\mathbf{b}(\mathbf{x}, \mu) = \int \phi(\|\mathbf{y} - \mathbf{x}\|)(\mathbf{y} - \mathbf{x})\,\mu(\mathrm{d}\mathbf{y}).$$

In order to infer the drift, rather than deriving it from explicit particle-level interactions, we instead rely on observations of the agents, which are typically more accessible in practical settings. We observe M independent trajectories, at discrete time instances $0 = t_0 < t_1 < \cdots < t_L = T$:

$$\mathbf{X}_{\mathrm{tr}} := \{\mathbf{X}_{t_\ell,m}^i\}_{i=1,\dots,N;\,\ell=0,\dots,L;\,m=1,\dots,M},$$

and calculate the corresponding velocities by first-order finite differences:

$$\mathbf{V}_{t_\ell,m}^i := \frac{\mathbf{X}_{t_{\ell+1},m}^i - \mathbf{X}_{t_\ell,m}^i}{t_{\ell+1} - t_\ell}, \qquad \ell = 0,\dots,L-1.$$

Each trajectory, indexed by m, is initialized by independent and identically distributed samples drawn from an (a priori unknown) initial distribution $\mu_{0,m}$. Our goal is to infer a measure-dependent drift $\mathbf{b} : \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}^d$ directly from trajectory data, where $\mathcal{P}_2(\mathbb{R}^d)$ denotes the set of probability measures with finite second moment. In this work, we focus on learning the drift term; the diffusion coefficient is treated as constant (or zero). Extensions to state- and measure-dependent diffusion are left for future work.

2.1 Measure-Valued Neural Network

One way to represent functionals on Wasserstein space is via cylindrical dependence, i.e., a functional depending on a measure only through finitely many test-function integrals. Concretely, a cylindrical functional has the following form:

$$\mu \;\longmapsto\; f\big(\langle g_1, \mu\rangle, \dots, \langle g_n, \mu\rangle\big),$$

for some test functions g_i and a finite-dimensional map f (e.g. polynomial/smooth), see, e.g., [69, 70]. On compact subsets of Wasserstein space, such cylindrical classes are dense in appropriate smoothness spaces; see Lemma 3.12 and Definition 2.4 in [69] for a representative result. The key computational advantage is that $\langle g, \mu_t^N\rangle$ can be evaluated by averaging over particles, which is permutation invariant and scales linearly in N. Motivated by this formulation, we approximate the drift by a composition of two neural networks: (1) an embedding network and (2) an interaction network. We define the MVNN drift as

$$\mathbf{b}_\theta(\mathbf{x}, \mu) := \varphi_{\mathrm{int}}\big(\mathbf{x},\ \langle \varphi_{\mathrm{emb}}(\cdot;\theta_{\mathrm{emb}}),\ \mu\rangle;\ \theta_{\mathrm{int}}\big)\,, \tag{2}$$

where $\mu \in \mathcal{P}_2(\mathbb{R}^d)$, $\varphi_{\text{emb}}(\cdot;\theta_{\text{emb}}) : \mathbb{R}^d \to \mathbb{R}^k$ extracts feature representations, and $\varphi_{\text{int}}(\cdot,\cdot;\theta_{\text{int}}) : \mathbb{R}^d \times \mathbb{R}^k \to \mathbb{R}^d$ maps the local state and the global (measure) feature to the drift. Here $\langle \cdot, \cdot \rangle$ denotes integration with respect to μ:

$$\langle \varphi_{\text{emb}}, \mu \rangle := \int_{\mathbb{R}^d} \varphi_{\text{emb}}\left(\mathbf{x};\theta_{\text{emb}}\right)\, \mu(\mathrm{d}\mathbf{x}).$$

When $\mu = \mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{\mathbf{X}_t^{j,N}}$, we have:

$$\langle \varphi_{\text{emb}}, \mu_t^N \rangle = \frac{1}{N}\sum_{j=1}^N \varphi_{\text{emb}}\left(\mathbf{X}_t^{j,N};\theta_{\text{emb}}\right),$$

and thus:

$$\mathbf{b}_\theta(\mathbf{X}, \mu_t^N) = \varphi_{\text{int}}\left(\mathbf{X},\ \frac{1}{N}\sum_{j=1}^N \varphi_{\text{emb}}\left(\mathbf{X}_t^{j,N};\theta_{\text{emb}}\right);\ \theta_{\text{int}}\right). \tag{3}$$

Evaluating (3) requires a single forward pass of φ_{emb} per particle and a single forward pass of φ_{int} per particle. Consequently, the computational complexity of evaluating the learned drift scales linearly with the number of particles, i.e., $O(N)$. This is in contrast to explicit pairwise interaction models, which incur a quadratic computational cost of order $O(N^2)$.

The model jointly learns (i) a finite collection of test functions φ_{emb} that extract informative features from the measure, and (ii) an interaction map φ_{int} that aggregates these features with the local state to approximate the drift. The embedding part can be interpreted as learning an optimal, data-driven test function basis on the space of measures. The embedding network φ_{emb} plays the role of the functions g_i in the cylindrical functional representation, but unlike polynomial or finite-element bases, it is learned from data instead of fixed a priori. Given the MVNN drift (2), the associated particle system with N agents is governed by:

$$\mathrm{d}\mathbf{X}_t^{\theta,i,N} = \mathbf{b}_\theta\left(\mathbf{X}_t^{\theta,i,N}, \mu_t^{\theta,N}\right)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t^{i,N}, \tag{4}$$

where $\mu_t^{\theta,N} = \frac{1}{N}\sum_{i=1}^N \delta_{\mathbf{X}_t^{\theta,i,N}}$ is the empirical law of the interacting particles. Formally, the limiting McKean-Vlasov dynamics for a single representative particle $\mathbf{X}_t$ is given by:

$$\begin{aligned}\mathrm{d}\mathbf{X}_t^\theta =& \mathbf{b}_\theta\left(\mathbf{X}_t^\theta, f_t^\theta\right)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t,\\ =& \varphi_{\text{int}}\left(\mathbf{X}_t^\theta, \int \varphi_{\text{emb}}(\mathbf{x}) f_t^\theta(\mathbf{x})\mathrm{d}\mathbf{x}\right)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t,\end{aligned} \tag{5}$$

where $f_t^\theta = \text{Law}(\mathbf{X}_t^\theta)$ is the law of the random variable $\mathbf{X}_t^\theta$ at time t. The corresponding Fokker-Planck equation for $(f_t^\theta)_{t\geq 0}$ in weak form is

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left\langle f_t^\theta, \psi\right\rangle &= \left\langle f_t^\theta, \mathbf{b}_\theta(\mathbf{x}, f_t^\theta)\cdot\nabla\psi + \frac{1}{2}\sigma^2\Delta\psi\right\rangle\\ &= \left\langle f_t^\theta, \varphi_{\text{int}}\left(\mathbf{x}, \int \varphi_{\text{emb}}(\mathbf{x}) f_t^\theta(\mathbf{x})\mathrm{d}\mathbf{x}\right)\cdot\nabla\psi + \frac{1}{2}\sigma^2\Delta\psi\right\rangle,\end{aligned} \tag{6}$$

for all smooth test functions $\psi : \mathbb{R}^d \to \mathbb{R}$ with compact support [73].

Proposition 2.1 (Well-Posedness of MVNN-Induced McKean–Vlasov Dynamics)**.** *Assume that φ_{int} and φ_{emb} are globally Lipschitz: there exist $C_i, C_e > 0$, such that for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d, \mathbf{y}, \mathbf{y}' \in \mathbb{R}^k$, it holds that:*

$$\begin{aligned}\|\varphi_{\text{emb}}(\mathbf{x}) - \varphi_{\text{emb}}(\mathbf{x}')\| &\leq C_e\|\mathbf{x}-\mathbf{x}'\|,\\ \|\varphi_{\text{int}}(\mathbf{x},\mathbf{y}) - \varphi_{\text{int}}(\mathbf{x}',\mathbf{y}')\| &\leq C_i\left(\|\mathbf{x}-\mathbf{x}'\| + \|\mathbf{y}-\mathbf{y}'\|\right).\end{aligned}$$

Assume that $f_0 \in \mathcal{P}_2(\mathbb{R}^d)$, for any $T > 0$, the SDE (5) *has a unique strong solution on [0,T] and consequently, its law is the unique weak solution to the Fokker-Planck equation* (6)*.*

Although Proposition 2.1 is a standard consequence of Lipschitz continuity for McKean-Vlasov SDEs, it guarantees that our learned MVNN drift induces a well-posed nonlinear Fokker-Planck evolution at the continuum level. This establishes our method as a rigorous framework for learning continuous governing equations directly from discretized particle-level observations, without requiring any smoothing or filtering procedures, in contrast to existing approaches [74, 75]. However, to complete the theoretical justification, we must demonstrate that the learned N-particle system (4) converges to this identified mean-field model (5) as $N \to \infty$.

Proposition 2.2 (Mean-Field Convergence and Propagation of Chaos for the Learned Particle System)**.** *Let the assumptions of Proposition 2.1 hold. Let* $(\mathbf{X}_t^{\theta,i,N})_{i=1}^N$ *be the* N*-particle system solving* (4) *with* f_0*-chaotic initial data* $\mathbf{X}_0^{\theta,i,N} \sim f_0^\theta$*. Let* f_t^θ *be the unique solution to the mean-field Fokker-Planck equation* (6) *with initial condition* $f_0^\theta = f_0$*. Then, the* N*-particle system* (4) *converges to the mean-field model* (5) *as* $N \to \infty$*. That is, for any* $T > 0$*, the* N*-particle distribution* $f_t^{\theta,N} = Law(\mathbf{X}_t^{\theta,1,N}, \ldots, \mathbf{X}_t^{\theta,N,N})$ *is* f_t*-chaotic, satisfying:*

$$\lim_{N\to\infty} W_2\left(f_{[0,T]}^{1,\theta,N}, f_{[0,T]}^{\theta}\right) = 0$$

where $f_t^{1,\theta,N}$ *is the first marginal of* $f_t^{\theta,N}$*.*

By establishing the propagation of chaos, Proposition 2.2 rigorously bridges the gap between the microscopic and macroscopic descriptions. It confirms that the learned particle dynamics systematically converge to the proposed mean-field model in the large-scale limit. The network structure (2) is closely related to the mean-field neural network architecture studied in [76] for optimal control on Wasserstein space. In particular, a universal approximation theorem shows that cylindrical neural functionals of the form (2) can approximate any continuous drift $\mathbf{b}^\star : \mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}^d$ with arbitrarily small error.

Theorem 2.3 (Universal Approximation for Measure Valued Neural Network)**.** *Let* ζ *be a probability measure on* $\mathcal{P}_2(\mathbb{R}^d)$*, and* $\mathbf{b}^\star$ *be a continuous map from* $\mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d) \to \mathbb{R}^d$*, such that* $\|\mathbf{b}^\star\|_{L^2(\zeta)}^2 := \int_{\mathcal{P}_2(\mathbb{R}^d)} \int_{\mathbb{R}^d} \|\mathbf{b}^\star(\mathbf{x},\mu)\|^2 \mu(\mathrm{d}\mathbf{x})\zeta(\mathrm{d}\mu) \le \infty$ *. Then for all* $\epsilon > 0$*, there exist* $k \in \mathbb{N}$*,* φ_{int} *a neural network, mapping from* $\mathbb{R}^d \times \mathbb{R}^k$ *into* $\mathbb{R}^d$ *and* φ_{emb} *another neural network, mapping from* $\mathbb{R}^d$ *into* $\mathbb{R}^k$*, such that*

$$\|\mathbf{b}^\star(\cdot,\cdot) - \varphi_{\text{int}}(\cdot, \langle \varphi_{\text{emb}}, \cdot\rangle)\|_{L^2(\zeta)} \le \epsilon$$

The proof follows directly from Theorem 2.2 in [76], which establishes the universal approximation property for functions on measure spaces of the form $\mathbb{R}^d \times \mathcal{P}_2(\mathbb{R}^d)$ using neural network parameterizations of cylindrical form. This theorem is a direct specialization of Theorem 2.2 in [76] to the drift setting $p = d$, with the notational correspondence $V \leftrightarrow b^\star$, $\Psi \leftrightarrow \phi_{\text{int}}$, and $\phi \leftrightarrow \phi_{\text{emb}}$.

While Theorem 2.3 ensures the theoretical capability of MVNN, it does not address the efficiency of the approximation. In fact, without further structural assumptions, the number of parameters required may grow rapidly with dimension. Following [57, 77], we define a ReLU network $q : \mathbb{R}^{d_1} \to \mathbb{R}$ as:

$$q(\mathbf{x}) = W_L \cdot \mathrm{Relu}(W_{L-1} \cdots \mathrm{Relu}(W_1\mathbf{x} + b_1) + \cdots + b_{L-1}) + b_L, \tag{7}$$

where W_l are weight matrics, b_l are bias vectors, $\mathrm{ReLU}(a) = \max\{a, 0\}$. We define the nextwork class $\mathcal{F}_{NN} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$:

$$\begin{aligned}\mathcal{F}_{NN}(d_1, d_2, L, p, K, \kappa, R) = \{&[q_1, q_2, \cdots, q_{d_2}]^\top \in \mathbb{R}^{d_1} : \text{for each } k = 1, \cdots, d_2,\\ &q_k : \mathbb{R}^{d_1} \to \mathbb{R} \text{ is in the form of 7 with } L \text{ layers, width bounded by } p\\ &\|q_k\|_{L^\infty} \le R, \|W_l\|_{\infty,\infty} \le \kappa, \|b_l\|_\infty \le \kappa, \sum_{l=1}^{L} \|W_l\|_0 + \|b_l\|_0 \le K, \forall l\}\end{aligned}$$

where $\|q\|_{L^\infty(\Omega)} = \sup_{\mathbf{x}\in\Omega} |q(\mathbf{x})|$, $\|W_l\|_{\infty,\infty} = \max_{ij} |(W_l)_{i,j}|$, $\|b\|_\infty = \max_i |b_i|$ and $\|\cdot\|_0$ denotes the number of nonzero elements of its argument.

Theorem 2.4 (Quantitative Approximation via Finite-Dimensional Measure Embeddings)**.** *Assume that the empirical measure* $\mu \in U \subset \mathcal{P}_2(\Omega)$ *is supported in* $\Omega = [-\gamma_1, \gamma_1]^d \subset \mathbb{R}^d$ *for some* $\gamma_1 > 0$*. We further assume that the map* $f : \Omega \times U \to \mathbb{R}$ *is a Lipschitz continuous map in the sense that for any* $\mu, \nu \in U$ *and* $\mathbf{x}, \mathbf{y} \in \Omega$*, there exists a constant* L_f *such that*

$$|f(\mathbf{x},\mu) - f(\mathbf{y},\nu)| \le L_f\left(\|\mathbf{x} - \mathbf{y}\| + W_1(\mu,\nu)\right).$$

For any $\epsilon > 0$*, there exist a constant* C *depends on* γ_1 *and* L_f*, and* $\{\mathbf{c}_m\}_{m=1}^{C_\Omega} \subset \Omega$ *so that* $\{B_\delta(\mathbf{c}_m)\}_{m=0}^{C_\Omega}$ *is a cover of* Ω *for* $\delta = \frac{\epsilon}{2L_f}$ *and some* $C_\Omega = \epsilon^{-d_1}$*. Let* $H = C(C_\Omega + d)^{\frac{C_\Omega+d}{2}}(C_\Omega d)^{\frac{C_\Omega+d}{2}}\epsilon^{-C_\Omega - d}$ *and there are* H *ReLU neural networks* $\{q_k\}_{k=1}^H$*, where* $q_k \in \mathcal{F}_{NN}(d + C_\Omega, 1, L, p, K, \kappa, R)$ *with*

$$\begin{aligned}L &= O((d + C_\Omega)^2 \log(\epsilon^{-1}) + (d + C_\Omega)^2\sqrt{C_\Omega d}L_f + (d + C_\Omega)^2 \log(d + C_\Omega))) \quad && p = O(1)\\ K &= O((d + C_\Omega)^2 \log(\epsilon^{-1}) + (d + C_\Omega)^2\sqrt{C_\Omega d}L_f + (d + C_\Omega)^2 \log(d + C_\Omega))\\ \kappa &= O\left((d + C_\Omega)^{\frac{d+C_\Omega}{2}+1}\epsilon^{-d-C_\Omega-1}L_h^{d+C_\Omega}\right) && R = O(1),\end{aligned}$$

such that

$$\sup_{\mu\in U,\mathbf{x}\in\Omega}\left|f(\mathbf{x},\mu)-\sum_{k=1}^{H}a_k q_k(\mathbf{x},\mathbf{u})\right|\leq\epsilon,$$

where $\mathbf{u}=(\langle\omega_1,\mu\rangle,\cdots,\langle\omega_{C_\Omega},\mu\rangle)$, $\{\omega_m(\mathbf{x})\}_{m=1}^{C_\Omega}$ *is partition of unity subordinate to the cover* $\{\mathcal{B}_\delta(\mathbf{c}_m)\}_{m=1}^{C_\Omega}$ *and* a_k *depends on* f.

Theorem 2.4 provides a constructive finite-dimensional approximation result of the same form as MVNN. The partition-of-unity features $\big(\langle\omega_m,\mu\rangle\big)_{m=1}^{C_\Omega}$ provide a fixed embedding of the measure, whereas MVNN replaces them with a learned embedding $\mathbf{u}=\langle\varphi_{\text{emb}},\mu\rangle$, and the networks q_k correspond to the interaction network φ_{int}. For example, when the output dimension is one, the terms are $a_k q(x,\mathbf{u})=\varphi_{\text{int}}(x,\langle\varphi_{\text{emb}},\mu\rangle)$. Theorem 2.4 highlights a fundamental bottleneck: without structural constraints, the complexity of approximating functionals on the Wasserstein space lead to the curse-of-dimensionality, where the required network size grows quickly with the dimension d. This implies that a naive application of MVNNs would be computationally intractable for high-dimensional systems. However, in physical, biological, and social contexts, the effective mean-field interactions are unlikely to be arbitrary functionals of the full probability measure. Instead, collective behaviors, such as flocking, synchronization, or aggregation, are typically governed by a compact set of macroscopic summary statistics, often referred to as order parameters in statistical physics (e.g., local density, mean momentum, or polarity). These systems are embedded in a low-dimensional manifold within the infinite-dimensional space of measures. To formalize and utilize this intrinsic physical structure, we introduce the following assumption, which bridges the gap between the theoretical worst-case complexity and the practical efficiency observed in our experiments.

Assumption 1 (Finite-Dimensional Measure Dependency)**.** Let $U\subset\mathcal{P}_2(\mathbb{R}^d)$ be the space of measures under consideration. We assume there exists a fixed, finite set of r feature functions $\mathcal{F}=\{f_i:\mathbb{R}^d\to\mathbb{R}\}_{i=1}^r$ that fully characterizes the dependencies for all relevant functionals. Specifically, for any functional $V:U\to\mathbb{R}$, there exists a corresponding function $G:\mathbb{R}^r\to\mathbb{R}$ such that:

$$V(\mu)=G(\langle f_1,\mu\rangle,\ldots,\langle f_m,\mu\rangle)\quad\forall\mu\in U$$

Remark 2.5. Assumption 1 effectively positing that the measures of interest U lie on a finite-dimensional manifold embedded within $\mathcal{P}_2(\mathbb{R}^d)$. An example is the family of Gaussian distributions $\mathcal{N}(\nu,\Sigma)$, which are uniquely determined by their mean and covariance (moments). Thus, any functional V defined only on this family of measures can be written as $V(\mu)=G(\nu,\Sigma)$, which perfectly matches our assumed form with a finite $r=d+d(d+1)/2$.

Remark 2.6. A direct consequence of Assumption 1 is that our target drift function $\mathbf{b}^\star(\mathbf{X},\mu)$, which depends on both the state $\mathbf{X}$ and the measure μ, must also admit a finite-dimensional representation. This follows by treating the state $\mathbf{X}$ as a fixed parameter. For any given $\mathbf{X}\in\mathbb{R}^d$, the mapping $\mathbf{b}^\star(\mathbf{X},\cdot):U\to\mathbb{R}^d$ is a pure functional of μ. Assumption 1 then guarantees the existence of a corresponding function $G_X:\mathbb{R}^r\to\mathbb{R}^d$ such that:

$$\mathbf{b}^\star(\mathbf{X},\mu)=G_{\mathbf{X}}(\langle f_1,\mu\rangle,\ldots,\langle f_r,\mu\rangle).$$

Since this holds for all $\mathbf{X}$, we can construct a single, global function $G:\mathbb{R}^d\times\mathbb{R}^r\to\mathbb{R}^d$ by defining $G(\mathbf{X},\mathbf{z}):=G_X(\mathbf{z})$, where $\mathbf{z}\in\mathbb{R}^r$.

Theorem 2.7 (Approximation Rate of MVNN under Low-Dimensional Assumption)**.** *Let the state* $\mathbf{X}\in[0,1]^d$ *and assume that the measure* μ *is supported on* $[0,1]^d$*. Suppose that Assumption 1 holds on* U *for some* $r\in\mathbb{N}$ *and a collection of feature functions* $\mathbf{g}=(g_1,\ldots,g_r)$*. As established in Remark 2.6, this implies that there exists a function* $G:\mathbb{R}^d\times\mathbb{R}^r\to\mathbb{R}^d$ *such that*

$$\mathbf{b}^\star(\mathbf{X},\mu)=G(\mathbf{X},\langle\mathbf{g},\mu\rangle).$$

We further assume that the functions $\mathbf{g}$ *and* G *are Lipschitz continuous with constants* $L_{\mathbf{g}}$ *and* L_G*, respectively.*

Then for any $\epsilon>0$*, there exists deep ReLU networks* φ_{emb} *and* φ_{int}*, such that*

$$\sup_{\mathbf{X}\in[0,1]^d,\,\mu\in U}\|\mathbf{b}^\star(\mathbf{X},\mu)-\varphi_{\text{int}}(\mathbf{X},\langle\varphi_{\text{emb}},\mu\rangle)\|_{\mathbb{R}^d}\leq\epsilon,$$

where the interaction network φ_{int} *has depth* $L_{\text{i}}=O(\log(\epsilon^{-1}))$ *and width* $W_{\text{i}}=O(\epsilon^{-(d+r)})$ *and the embedding network* φ_{emb} *has depth* $L_{\text{e}}=O(\log(\epsilon^{-1}))$ *and width* $W_{\text{e}}=O\big(\epsilon^{-d}\big)$.

2.2 Learning Objective and Optimization

Given the observation dataset:

$$\mathcal{D}_{\text{obs}}=\big\{(\mathbf{X}^i_{t_l,m},\mathbf{V}^i_{t_l,m})\big\}_{i=1,l=1,m=1}^{N,L,M},$$

we approximate the empirical measure at time t_l in trajectory m by $\hat{\mu}_{t_l,m} = \frac{1}{N}\sum_{j=1}^{N}\delta_{\mathbf{X}^j_{t_l,m}}$. The mean-field drift predicted by the MVNN is then given by:

$$\hat{\mathbf{b}}_\theta(\mathbf{X}^i_{t_l,m}, \hat{\mu}_{t_l,m}) = \varphi_{\text{int}}\left(\mathbf{X}^i_{t_l,m},\ \frac{1}{N}\sum_{j=1}^{N}\varphi_{\text{emb}}(\mathbf{X}^j_{t_l,m};\theta_{\text{emb}});\theta_{\text{int}}\right).$$

The model parameters $\theta = (\theta_{\text{int}}, \theta_{\text{emb}})$ are learned by minimizing the discrepancy between the observed and predicted trajectory. Let $\mathbb{P}^\theta$ denote the path measure induced by the solution $(\mathbf{X}^\theta_s)_{s\in[0,t]}$ of the McKean-Vlasov SDE (5). Under standard regularity conditions ensuring absolute continuity of path measures, Girsanov's theorem yields the log-likelihood function (see [78])

$$\begin{aligned}\mathcal{L}_t(\theta) := \log\frac{\mathrm{d}\mathbb{P}^\theta}{\mathrm{d}\mathbb{P}^{\mathbf{b}}} = &\int_0^t \left\langle \hat{\mathbf{b}}_\theta(\mathbf{X}_s,\mu^\theta_s) - \mathbf{b}(\mathbf{X}_s,\mu_s),\ (\sigma(\mathbf{X}_s)\sigma(\mathbf{X}_s)^\top)^{-1}\,\mathrm{d}\mathbf{X}_s\right\rangle \\ &- \frac{1}{2}\int_0^t \left(\left\|\sigma(\mathbf{X}_s)^{-1}\hat{\mathbf{b}}_\theta(\mathbf{X}_s,\mu^\theta_s)\right\|^2 - \left\|\sigma(\mathbf{X}_s)^{-1}\mathbf{b}(\mathbf{X}_s,\mu_s)\right\|^2\right)\mathrm{d}s.\end{aligned} \tag{8}$$

Similar to Section 2.1.2 in [79], when we only observe discrete-time samples of M trajectories of an N-particle system at $t_l = l\Delta t$: $\{\mathbf{X}^i_{t_l,m}\}_{i=1}^N$ for $m = 1,\dots,M$ and $l = 0,\dots,L$. Approximating the stochastic integral by the Euler–Maruyama method, we obtain the discrete-time likelihood (up to θ-independent constants):

$$\begin{aligned}\hat{\mathcal{L}}(\theta) = \frac{1}{MLN}\sum_{m=1}^{M}\sum_{l=0}^{L-1}\sum_{i=1}^{N}\Bigg[&\left\langle \hat{\mathbf{b}}_\theta(\mathbf{X}^i_{t_l,m},\hat{\mu}_{t_l,m}) - \mathbf{b}(\mathbf{X}^i_{t_l,m},\hat{\mu}_{t_l,m}), (\sigma\sigma^\top)^{-1}\Delta\mathbf{X}^i_{t_l,m}\right\rangle \\ &- \frac{1}{2}\Big(\|\sigma^{-1}\hat{\mathbf{b}}_\theta(\mathbf{X}^i_{t_l,m},\hat{\mu}_{t_l,m})\|^2 - \|\sigma^{-1}\mathbf{b}(\mathbf{X}^i_{t_l,m},\hat{\mu}_{t_l,m})\|^2\Big)\Delta t\Bigg],\end{aligned} \tag{9}$$

where $\Delta\mathbf{X}^i_{t_l,m} := \mathbf{X}^i_{t_{l+1},m} - \mathbf{X}^i_{t_l,m}$. We further assume $\sigma(x) \equiv \sigma I$ with a constant scalar $\sigma > 0$. Then, dropping all terms independent of θ, (9) reduces to:

$$\hat{\mathcal{L}}(\theta) = -\frac{\Delta t}{2\sigma^2}\,\frac{1}{MLN}\sum_{m=1}^{M}\sum_{l=0}^{L-1}\sum_{i=1}^{N}\left\|\hat{\mathbf{b}}_\theta(\mathbf{X}^i_{t_l,m},\hat{\mu}_{t_l,m}) - \mathbf{V}^i_{t_l,m}\right\|^2 \ +\ C. \tag{10}$$

Therefore, maximizing $\hat{\mathcal{L}}(\theta)$ is equivalent (up to a positive scaling) to minimize the mean-squared regression loss:

$$\theta^\star \in \arg\min_\theta\ \frac{1}{MLN}\sum_{m=1}^{M}\sum_{l=0}^{L-1}\sum_{i=1}^{N}\left\|\mathbf{V}^i_{t_l,m} - \hat{\mathbf{b}}_\theta(X^i_{t_l,m},\hat{\mu}_{t_l,m})\right\|^2. \tag{11}$$

While our objective function (11) is derived heuristically from the discretization of the path measure, its theoretical validity in the mean-field limit is supported by recent results on parameter estimation for McKean-Vlasov SDEs [79]. They proved that under standard regularity conditions, the estimator obtained by maximizing the particle likelihood is consistent in the limit as $N \to \infty$ (Theorem 1.1 in [79]). Furthermore, the estimator exhibits asymptotic normality with a convergence rate of $\mathcal{O}(N^{-1/2})$ (Theorem 1.2 in [79]). These results guarantee that minimizing our regression loss $\hat{\mathcal{L}}(\theta)$, which is equivalent to maximizing the log-likelihood, will recover the true mean-field drift $\mathbf{b}(\mathbf{x},\mu)$ as the number of particles increases, provided the neural network has sufficient capacity. We optimize (11) using Adam [80] with mini-batches to improve computational efficiency and training stability. All gradients are obtained via automatic differentiation in JAX [81].

2.3 Numerical Result

We present numerical experiments that validate the accuracy of the learned model and its ability to generalize to previously unseen initial configurations. Unless otherwise stated, we compare the learned mean-field model against reference particle simulations.

2.3.1 1D Motsch-Tadmor Dynamics

We begin by validating our framework on the Motsch-Tadmor model [24, 82]. Unlike standard alignment models with additive forces, the Motsch-Tadmor dynamics feature a normalized interaction mechanism, where the influence of

neighbors is weighted by their relative distance and normalized by the total interaction strength. The evolution of N agents with scalar states $X_t^i \in \mathbb{R}$ is governed by:

$$\dot{X}_t^i = \frac{1}{\sum_{k=1}^N \phi\left(|X_t^k - X_t^i|\right)} \sum_{j=1}^N \phi\left(|X_t^j - X_t^i|\right)\left(X_t^j - X_t^i\right), \qquad i = 1, \ldots, N.$$

Here we set the population size $N = 16,000$ and the time horizon $T = 2$. The interaction kernel is chosen as a Gaussian function $\phi(r) = \exp\left(-(r/\ell)^2\right)$ with a characteristic length $\ell = 0.5$. The normalization term in the denominator introduces a non-trivial dependency on the global empirical measure, making the macroscopic drift strictly non-pairwise and highly nonlinear. We discretize the system using a forward Euler scheme with timestep $\Delta t = 1 \times 10^{-2}$ to generate $M = 100$ independent trajectories $\{X_{tl,m}^i\}$, $l = 0, \cdots, 200, m = 1, \cdots, 100$. For each realization m, initial distributions $X_{0,m}^i$ are sampled from a randomly generated multimodal distribution μ_0^m. Specifically, each μ_0^m is constructed as a mixture of 2-8 Gaussian components whose mean uniformly distributed in $[0, 3]$ and variance is 0.25. The mixture weights of different Gaussian is sampled from a symmetric Dirichlet distribution, producing a diverse set of initializations. The dataset is split into 97 trajectories for training and 3 for testing. The learned mean-field dynamics are simulated via the McKean-Vlasov equation driven by the trained MVNN:

$$\dot{X}_t^{i,N} = \hat{b}_\theta\left(X_t^{i,N}, \hat{\mu}_t^N\right) \qquad i = 1, \ldots, N,$$

where the empirical measure is defined as $\hat{\mu}_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{X_t^{j,N}}$. Unlike pairwise interaction estimation, which aims to recover the microscopic interaction kernel, our evaluation focuses on the system's macroscopic behavior. Specifically, we compare the predicted and true population densities over time to assess the quality of the learned mean-field dynamics. Figure 1 compares the time evolution of the densities predicted by our model against the ground truth for three unseen initial conditions. The results demonstrate that the MVNN successfully captures the clustering and consensus formation inherent in the dynamics. Despite the complex normalization factor, the learned model accurately reproduces the merging of clusters and the preservation of density peaks without explicit knowledge of the microscopic interaction form.

Comparisons with Gaussian Process: Additionally, we compare the learned MVNN dynamics to predictions generated from a Gaussian process model with a Matérn kernel [83]. For the comparison, the Gaussian process model was trained on $N = 16$ agents and $M = 9$ independent trajectories, and the system was discretized using a forward Euler scheme with $L = 20$ timesteps of size $\Delta t = 5 \times 10^{-2}$ on the interval $[0, 1]$. The MVNN model was trained on $N = 16,000$ agents and $M = 100$ trajectories, and the system was discretized using a forward Euler scheme with $L = 200$ timesteps of size $\Delta t = 10^{-2}$ on the interval $[0, 2]$. The Gaussian process model is restricted to smaller training set sizes relative to the MVNN model due to the construction of a kernel matrix that scales with the total size of the training set. The initial distributions of each trajectory for both models are sampled from a Gaussian mixture model with 2-8 components, whose mean is uniformly distributed in $[0, 3]$, variance is 0.25, and mixture weights are sampled from a symmetric Dirichlet distribution. Figure 2 compares the time evolution of the densities predicted by the MVNN and Gaussian process models against the ground truth for $N = 16,000$ agents and three unseen initial conditions. Figure 3 shows the L^2 error between the true density and the predictions using the Gaussian process model, the MVNN model trained on 16 agents, 9 trajectories, and 20 timesteps (the same training set as the Gaussian process model), and the MVNN model trained on 16,000 agents, 100 trajectories, and 200 timesteps. The results show that MVNN matches the reference dynamics better than the Gaussian process model and achieves lower test error even when trained on a small number of agents and trajectories. MVNN is able to handle a large number of agents, which may be needed in the mean-field limit, and can also accurately account for the normalization factor in the Motsch-Tadmor model. In contrast, the Gaussian process model is limited to a smaller number of agents, and it cannot accurately handle the normalization factor, since it requires the normalized interaction kernel to be symmetric. Figure 4 compares the average simulation times against the number of agents N for the MVNN and Gausian process models over 10 trials. The simulation times for MVNN remain approximately constant as N increases, whereas the simulation times for the Gaussian process model increase significantly as N increases.

2.3.2 1D Stochastic Motsch-Tadmor Dynamics

We extend our validation to the stochastic variant of the system, where the dynamics are driven by:

$$\mathrm{d}X_t^i = \frac{1}{\sum_{k=1}^N \phi\left(|X_t^k - X_t^i|\right)} \sum_{j=1}^N \phi\left(|X_t^j - X_t^i|\right)\left(X_t^j - X_t^i\right)\mathrm{d}t + \sigma\,\mathrm{d}B_t^i, \qquad i = 1, \ldots, N,$$

with noise strength $\sigma = 0.1$. For the stochastic case, we employ the Euler–Maruyama method with time step $\Delta t = 10^{-2}$, whereas the remaining parameters are kept identical to the deterministic configuration. Consistent with our problem

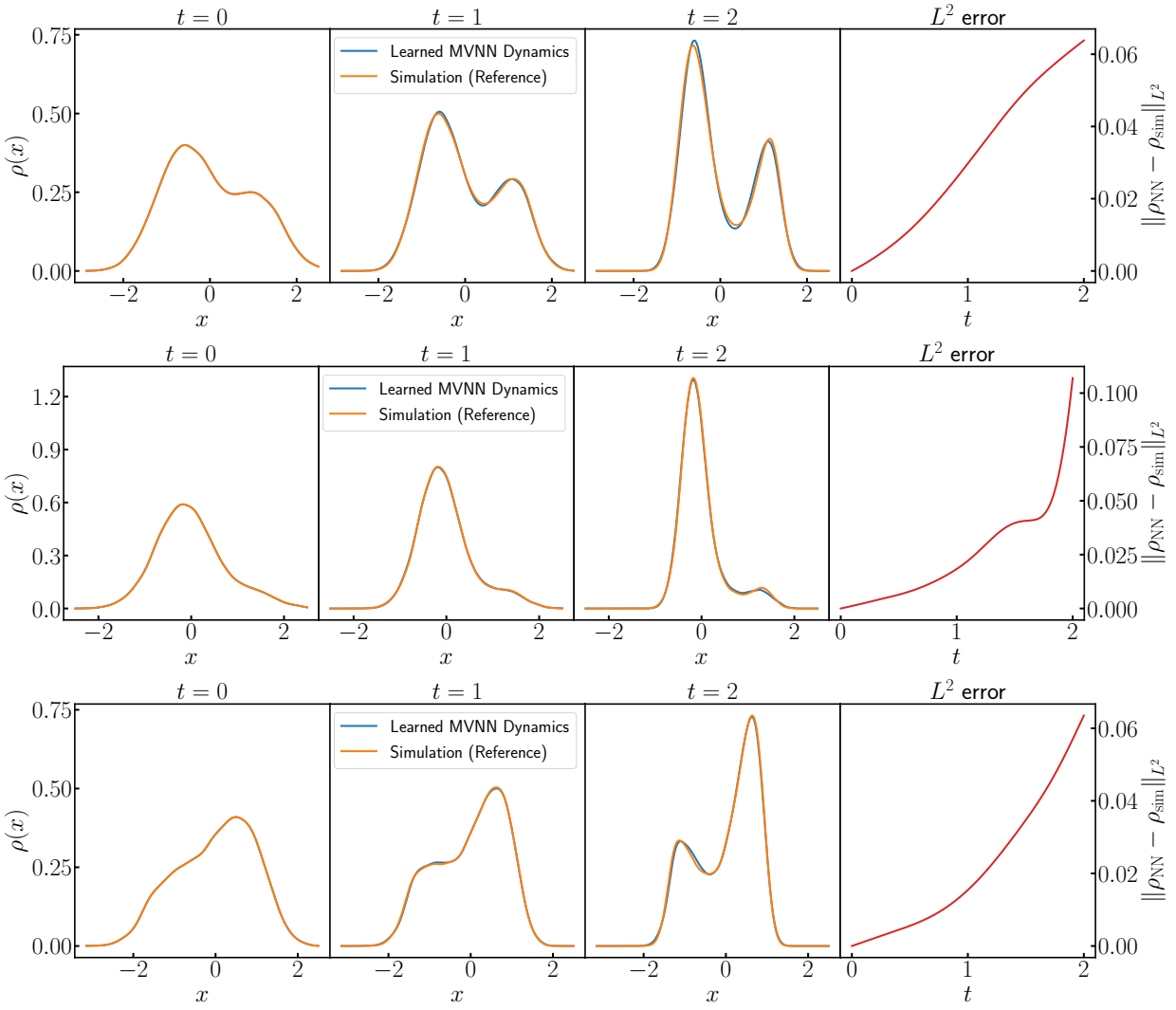


Figure 1: **1D Motsch-Tadmor dynamics**: Empirical density $\rho(x, t)$ for the 1D Motsch-Tadmor dynamics: comparison between the reference N-particle simulation (orange) and the MVNN-learned mean-field model (blue). Columns show $t = 0, 1, 2$; rows correspond to three unseen initial distributions. Densities are estimated using Gaussian kernel density estimation. The L^2 error is computed between the KDE-smoothed densities.

setup, the diffusion coefficient σ is treated as a known constant, and the learning task focuses solely on identifying the effective drift field.The learned macroscopic dynamics are simulated via the corresponding McKean-Vlasov SDE driven by the trained MVNN:

$$\mathrm{d}X_t^{i,N} = \hat{b}_\theta\left(X_t^{i,N},\, \hat{\mu}_t^N\right)\mathrm{d}t + \sigma \mathrm{d}B_t^{i,N}, \qquad i = 1, \ldots, N,$$

where $\hat{\mu}_t^N$ denotes the empirical measure defined above. Figure 5 shows the time dynamics of the agent density under stochastic forcing and a comparison with the true simulation.

2.3.3 2D Aggregation Dynamics with Attraction-Repulsion

We further evaluate the framework on a two-dimensional first-order swarm model governed by attractive-repulsive interactions. Unlike the consensus models discussed previously, this system exhibits rich spatial pattern formation, such as rings and clumps. The dynamics of the N agents are described by:

$$\dot{\mathbf{X}}_t^i = \frac{1}{N}\sum_{j=1}^{N} \phi\left(\|\mathbf{X}_t^j - \mathbf{X}_t^i\|\right)(\mathbf{X}_t^j - \mathbf{X}_t^i), \qquad i = 1, \ldots, N,$$

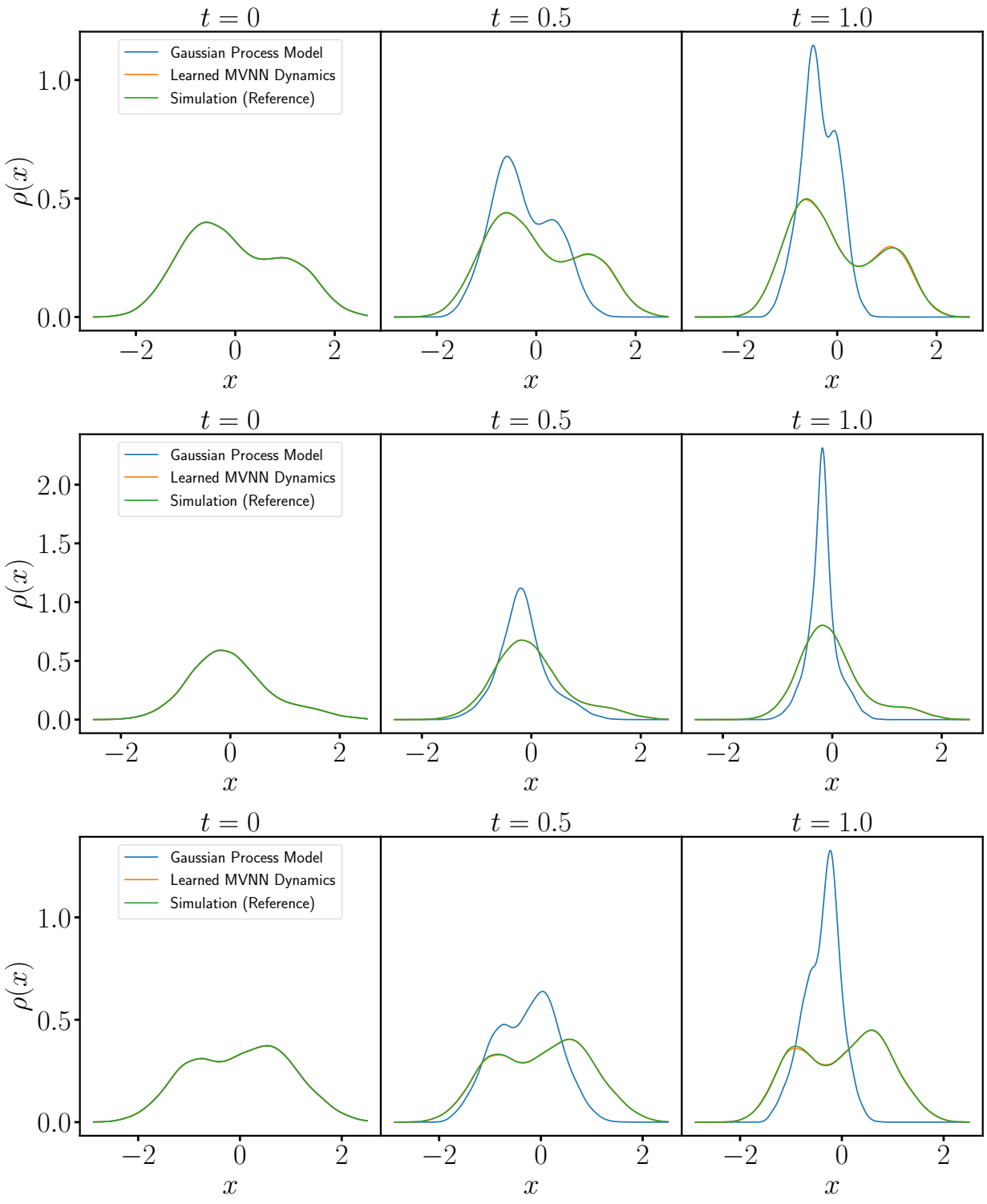


Figure 2: **Comparisons on 1D Motsch-Tadmor dynamics**: Empirical density $\rho(x, t)$ for the 1D Motsch-Tadmor dynamics: comparison between the reference N-particle simulation (green), the MVNN-learned mean-field model (orange), and the prediction from the Gaussian process model [83] (blue). Columns show $t = 0, 0.5, 1$; rows correspond to three unseen initial distributions. Densities are estimated using Gaussian kernel density estimation.

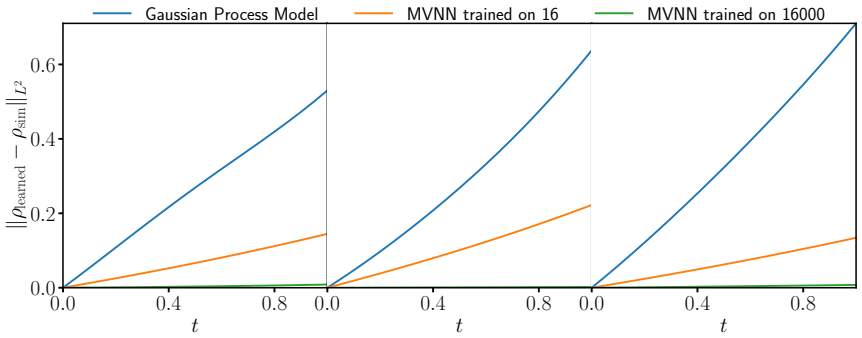


Figure 3: **Comparisons on 1D Motsch-Tadmor dynamics**: L^2 error for the 1D Motsch-Tadmor dynamics: comparison of the L^2 error for the Gaussian process model (blue), the MVNN model trained on 16 agents, 9 trajectories, and 20 timesteps (orange), and the MVNN model trained on $16,000$ agents, 100 trajectories, and 200 timesteps (green). The L^2 error is computed between the KDE-smoothed densities. Columns correspond to three unseen initial distributions.

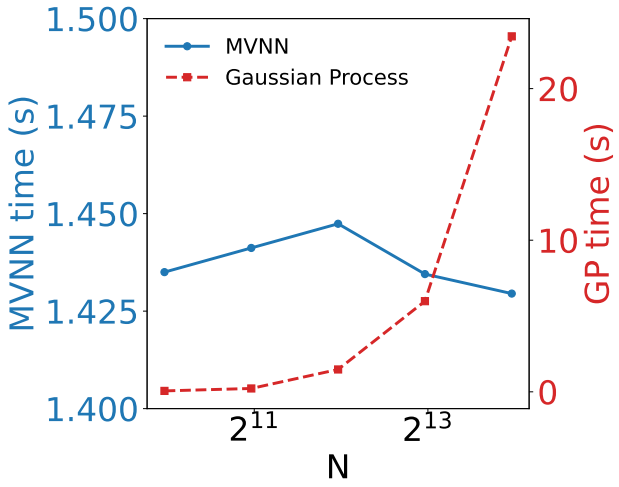


Figure 4: **Simulation Time Comparison**: Comparison of average simulation times (seconds) against number of agents N for the MVNN and Gaussian process models over 10 trials.

where $\mathbf{X}_t^i \in \mathbb{R}^2$ denotes the position of agent i. The interaction kernel ϕ combines short-range repulsion and long-range attraction, modeled by a sum of Gaussians:

$$\phi(r) = c_{\text{rep}} \exp\left(-(r/\ell_{\text{rep}})^2\right) - c_{\text{att}} \exp\left(-(r/\ell_{\text{att}})^2\right).$$

We use the parameters $c_{\text{rep}} = 1.0$, $\ell_{\text{rep}} = 0.5$, $c_{\text{att}} = 0.7$, and $\ell_{\text{att}} = 2.0$. The simulation is performed with $N = 16{,}000$ agents for 200 steps using a time step $\Delta t = 10^{-2}$. To test the model's ability to learn and represent complex geometric structures, we construct the training set using noisy annulus initial configurations. Specifically, for each trajectory, the initial position of each agent i is then sampled via polar coordinates with additive Gaussian noise:

$$\Theta_i \sim \mathcal{U}(0, 2\pi), \quad \rho_i \sim \mathcal{U}\big(R_0 - \tfrac{W}{2},\, R_0 + \tfrac{W}{2}\big),$$
$$\mathbf{X}_0^i = \rho_i\big(\cos\Theta_i, \sin\Theta_i\big) + \varepsilon_i, \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(\mathbf{0},\, \sigma_0^2 I_2),$$

where I_2 denotes the identity matrix. We generate 100 distinct initial distributions to form the dataset. Figure 6 visualizes the evolution of the system initialized from a ring distribution. The learned mean-field dynamics successfully reproduce the stability of the ring structure and the correct contraction rate, demonstrating that the MVNN can capture effective potentials that support metastable geometric patterns.

To assess the generalization capability of the learned MVNN, we evaluate the model on initial distributions with topological structures and density profiles distinct from the training set. Specifically, we consider three test cases: a double-ring, a uniform disk, and a binary distribution exhibiting spatial heterogeneity (asymmetric density with a

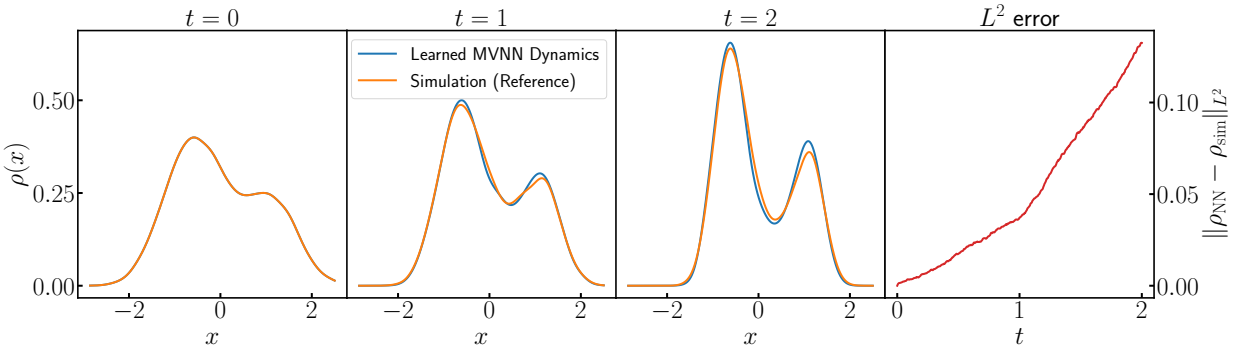


Figure 5: **Stochastic Motsch-Tadmor dynamics** ($\sigma = 0.1$): density evolution. Empirical density $\rho(x, t)$ from the reference interacting-particle simulation (orange) and from the MVNN-learned McKean-Vlasov model (blue), shown at $t = 0, 1, 2$ for an unseen initial distribution. Densities are estimated via Gaussian KDE, and the reported L^2 error is computed between the kernel-smoothed densities.

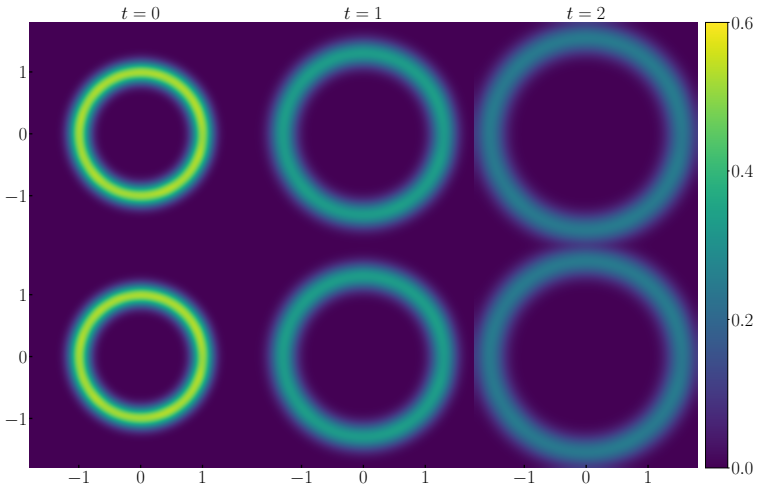


Figure 6: **2D aggregation model with ring-shaped initialization.** The upper row displays the ground truth particle trajectories, while the lower row shows the evolution predicted by the learned mean-field dynamics. The model accurately preserves the topological structure of the ring over time.

low-density region on the left and a high-density region on the right). The results are presented in Figures 7, 8, and 9, respectively. In all cases, the learned mean-field dynamics (bottom rows) yield excellent agreement with the ground truth particle simulations (top rows). These results confirm that the MVNN has successfully learned the intrinsic interaction operator rather than merely overfitting to the geometry of the training data, thus providing robust predictions on unseen configurations.

3 Extensions to Systems with Heterogeneous Agent Groups

Our method extends naturally to a wide class of interacting agent systems that arise in different applications, including systems involving multiple groups of agents. Let the agents be divided into K distinct groups, with the k-th group consisting of N_k agents whose states are denoted by $\mathbf{X}^{i,k}$, for $i = 1, \ldots, N_k$ and $k = 1, \ldots, K$. The dynamics of the system can then be described by:

$$\dot{\mathbf{X}}_t^{i,k} = \sum_{l=1}^{K} \frac{1}{N_l - \delta_{k,l}} \sum_{\substack{j=1 \\ (i,k)\neq(j,l)}}^{N_l} \phi_{k,l}\Big(\|\mathbf{X}_t^{i,k} - \mathbf{X}_t^{j,l}\|\Big)\,\big(\mathbf{X}_t^{i,k} - \mathbf{X}_t^{j,l}\big), \qquad i = 1, \ldots, N_k, \;\; k = 1, \ldots, K,$$

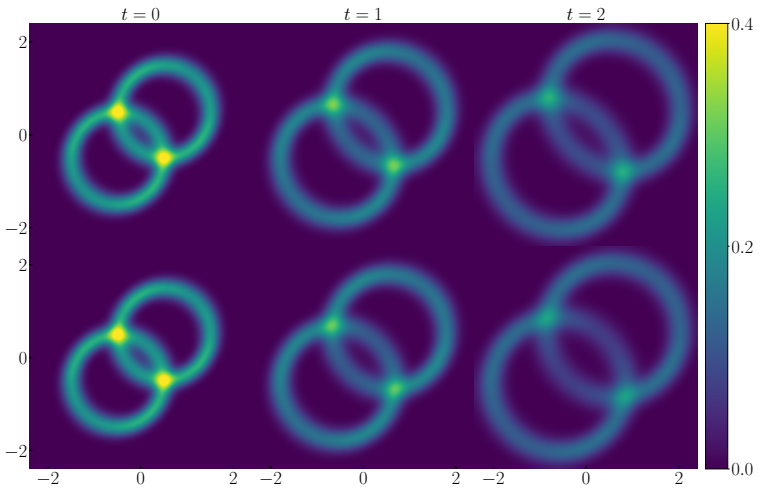


Figure 7: **2D aggregation model with double-ring initialization.** Comparison between the ground truth particle system (upper row) and the learned mean-field dynamics (lower row). The model correctly reproduces the contraction of both rings despite never seeing this topology during training.

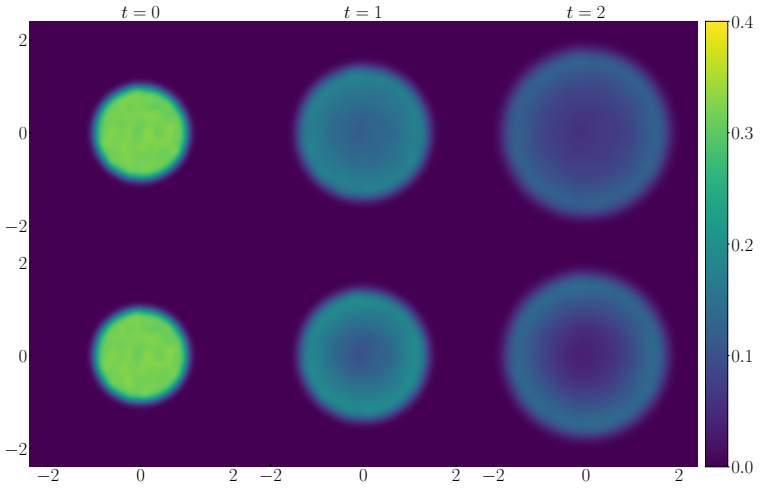


Figure 8: **2D aggregation model with disk-shaped initialization.** The learned dynamics (lower row) accurately capture the collapse of the uniform disk, matching the ground truth (upper row).

where $\phi_{k,l}$ represents the interaction kernel between agents in group k and group l. The corresponding mean-field dynamics can be written in the form:

$$\dot{\mathbf{X}}_t^{i,k} = \mathbf{b}_k\Big(\mathbf{X}_t^{i,k},\, \mu_1, \ldots, \mu_K\Big), \qquad i = 1, \ldots, N_k, \;\; k = 1, \ldots, K,$$

where each $\mu_k \in \mathcal{P}(\mathbb{R}^d)$ denotes the probability distribution of agents of group k. The observed data consist of the positions of agents from different groups along multiple trajectories:

$$\mathbf{X}_{\text{tr}} := \{\mathbf{X}_{t_\ell,m}^{i,k}\}_{i=1,\ell=1,m=1,k=1}^{N_k,L,M,K},$$

where $0 = t_1 < \cdots < t_L = T$ denote the observation times, m indexes the M independent trajectories, and $k = 1, \ldots, K$ indexes the agent groups. For each group k, the corresponding velocities $\mathbf{V}_{t_\ell,m}^{i,k}$ are approximated by finite differences. Each trajectory m is initialized from a collection of independent probability measures $\mu_{0,1}^m, \ldots, \mu_{0,K}^m$, representing the initial distributions of the K groups, yielding different realizations of the interacting multi-group system. Our objective is to infer the mean-field interaction drifts $\mathbf{b}_k(\mathbf{X}, \mu_1, \ldots, \mu_K),\ k = 1, \ldots, K$, from these observations.

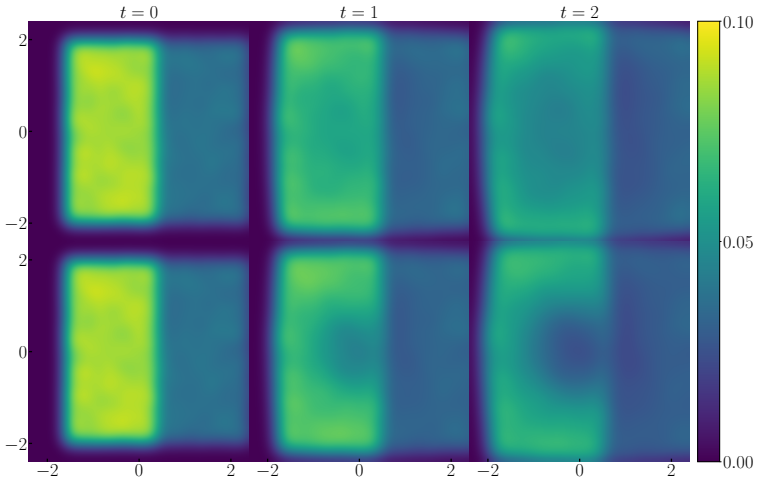


Figure 9: **2D aggregation model with binary asymmetric initialization.** Evolution of a system initialized with heterogeneous densities (low density left, high density right). The learned model (lower row) preserves the density gradient and correctly predicts the asymmetric aggregation process.

3.1 Multi-Group Measure-Valued Neural Network

We extend the proposed framework to heterogeneous systems composed of multiple interacting agent groups. We introduce the *Multi-Group Measure-Valued Neural Network* (MG-MVNN) to capture the complex inter-group coupling. Consider a system with K distinct groups, where the k-th group is characterized by its population distribution $\mu_k \in \mathcal{P}(\mathbb{R}^d)$. The effective mean-field dynamics for an agent i in group k are governed by a drift function $\mathbf{b}_k$ dependent on the state and the distributions of all groups:

$$\dot{\mathbf{X}}_t^{i,k} = \mathbf{b}_k\Big(\mathbf{X}_t^{i,k},\, \mu_1, \ldots, \mu_K\Big)\,.$$

To allow for efficient learning of these high-dimensional dependencies, we parametrize each drift $\mathbf{b}_k$ using a composite neural operator. This architecture maps the local agent state $\mathbf{X}$ and the aggregated population features from all groups to the target drift. Specifically, for each groups $l \in \{1, \ldots, K\}$, we employ a group-specific embedding network $\varphi_{\text{emb},l} : \mathbb{R}^d \to \mathbb{R}^{r_l}$ to extract latent representations. The collective state of group l is then summarized by the moment vector:

$$\mathbf{z}_l(\mu_l) := \big\langle \varphi_{\text{emb},l}(\cdot;\theta_{\text{emb},l}),\, \mu_l \big\rangle = \int \varphi_{\text{emb},l}(\mathbf{x})\, \mu_l(\mathrm{d}\mathbf{x}).$$

The drift for group k is approximated by an interaction network $\varphi_{\text{int},k}$ that consumes the local state and the concatenated global features of all groups:

$$\mathbf{b}_k(\mathbf{X}, \mu_1, \ldots, \mu_K) \;\approx\; \varphi_{\text{int},k}\Big(\mathbf{X},\; \mathbf{z}_1(\mu_1), \ldots, \mathbf{z}_K(\mu_K);\; \theta_{\text{int},k}\Big).$$

This design enforces permutation invariance within each group while allowing for complex, asymmetric interactions between different groups. Given a multi-group dataset $\mathcal{D}_{\text{obs}} = \{(\mathbf{X}_{t_\ell,m}^{i,k}, \mathbf{V}_{t_\ell,m}^{i,k})\}$, we replace the theoretical moments $\mathbf{z}_l(\mu_l)$ with their Monte Carlo estimates based on the empirical measures $\hat{\mu}_{t_\ell,m}^l = \frac{1}{N_l}\sum_{j=1}^{N_l} \delta_{\mathbf{X}_{t_\ell,m}^{j,l}}$. The empirical population feature for group l becomes:

$$\hat{\mathbf{z}}_{t_\ell,m}^l = \frac{1}{N_l}\sum_{j=1}^{N_l} \varphi_{\text{emb},l}(\mathbf{X}_{t_\ell,m}^{j,l}; \theta_{\text{emb},l}).$$

Consequently, the predicted mean-field drift is given by:

$$\hat{\mathbf{b}}_{\theta,k} = \varphi_{\text{int},k}\Big(\mathbf{X}_{t_\ell,m}^{i,k},\; \hat{\mathbf{z}}_{t_\ell,m}^1, \ldots, \hat{\mathbf{z}}_{t_\ell,m}^K;\; \theta_{\text{int},k}\Big)\,.$$

The network parameters $\Theta = \{\theta_{\text{int},k}, \theta_{\text{emb},k}\}_{k=1}^K$ are jointly learned by minimizing the global trajectory matching error:

$$\mathcal{L}(\Theta) = \frac{1}{ML}\sum_{m=1}^{M}\sum_{\ell=1}^{L}\sum_{k=1}^{K}\frac{1}{N_k}\sum_{i=1}^{N_k}\big\|\mathbf{V}_{t_\ell,m}^{i,k} - \hat{\mathbf{b}}_{\theta,k}\big\|^2.$$

The optimization is performed using Adam [80] within the `JAX` framework [81], utilizing automatic differentiation to compute gradients through the empirical averages. To ensure scalability and training stability, we employ randomized mini-batching over the agent-time indices.

3.2 Numerical Result for MG-MVNN

We evaluate the multi-group framework on a hierarchical system with asymmetric interactions, designed to mimic a stratified communication network. The system comprises $K = 3$ groups with population sizes $(N_1, N_2, N_3) = (16{,}000,\ 4{,}000,\ 200)$, labeled as Group 1, Group 2 , and Group 3. Each pair of group (k, l) interacts through a group-dependent influence kernel:

$$\phi_{k,l}(r) = D_{k,l}\ \exp\left(1 - \frac{1}{1 - |r/R_l|^{10}}\right) \mathbf{1}_{(-1,1)}\left(\frac{r}{R_l}\right),$$

where $\mathbf{1}_{(-1,1)}$ denotes the indicator function on $(-1, 1)$. The coefficients $D_{k,l}$ quantify the influence strength of group l on group k, while R_l specifies the interaction radius of group l. In this hierarchical setting, the influence matrix $(D_{k,l})$ is given by:

$$D = \begin{pmatrix} 5 & 10 & 0 \\ 0 & 2 & 5 \\ 0 & 0 & 1 \end{pmatrix}, \qquad (R_1, R_2, R_3) = (1.0,\ 2.5,\ 5.0).$$

The upper-triangular structure of D enforces a directional flow of influence from higher-ranking group to lower ones, while the diagonal terms represent intra-group cohesion. Notably, the interaction radii increase with the hierarchy level ($R_3 > R_2 > R_1$), so Group 3 acts over the broadest spatial range, whereas Group 1 interacts most locally.

The training dataset consists of $M = 100$ trajectories initialized from random Gaussian mixtures, as described in the single-group experiments. To assess the model's ability to capture complex cross-group coupling, we test on two specific out-of-distribution scenarios where the groups are initially spatially separated.Figures 10 and 11 display the evolution of the groups densities. The MVNN accurately predicts the hierarchical entrainment process: Group 3 moves independently to form a consensus; Group 2 is pulled towards Group 3; and Group 1 subsequently clusters around Group 2. This sequential locking of dynamics confirms that our multi-group architecture correctly learns the asymmetric causal structure encoded in the influence matrix.

4 Second-Order Dynamics

The proposed framework can be applied to second-order interacting particle/agent systems. Consider the second-order McKean–Vlasov stochastic differential equation for particles involving position $(\mathbf{X}_t)$ and velocity $(\mathbf{V}_t)$:

$$\begin{aligned} \mathrm{d}\mathbf{X}_t &= \mathbf{V}_t \mathrm{d}t \\ \mathrm{d}\mathbf{V}_t &= \mathbf{b}(\mathbf{X}_t, \mathbf{V}_t, f_t)\mathrm{d}t + \sigma(\mathbf{X}_t, \mathbf{V}_t, f_t)\mathrm{d}\mathbf{B}_t, \end{aligned}$$

where $f_t = \mathcal{L}(\mathbf{X}_t, \mathbf{V}_t)$ describes the law of the pair $(\mathbf{X}_t, \mathbf{V}_t)$, and $\mathbf{B}_t$ is a d-dimensional Wiener process. We can approximate the second-order McKean-Vlasov dynamics by a system of N stochastic differential equations involving the position-velocity pairs $\left(\mathbf{X}_t^{i,N}, \mathbf{V}_t^{i,N}\right)_{1\le i\le N}$, where N is sufficiently large. The initial conditions $\left(\mathbf{X}_0^{i,N}, \mathbf{V}_0^{i,N}\right)_{1\le i\le N}$ are independent and identically distributed with law μ_0 and each particle evolves according to:

$$\begin{aligned} \mathrm{d}\mathbf{X}_t^{i,N} &= \mathbf{V}_t^{i,N}\mathrm{d}t \\ \mathrm{d}\mathbf{V}_t^{i,N} &= \mathbf{b}(\mathbf{X}_t^{i,N}, \mathbf{V}_t^{i,N}, \mu_t^N)\mathrm{d}t + \sigma(\mathbf{X}_t^{i,N}, \mathbf{V}_t^{i,N}, \mu_t^N)\mathrm{d}\mathbf{B}_t^{i,N}, \end{aligned}$$

where $\mathbf{B}_t^i$ are d-dimensional Wiener processes, $\mu_t^N = \frac{1}{N}\sum_{j=1}^N \delta_{\left(\mathbf{X}_t^{j,N}, \mathbf{V}_t^{j,N}\right)}$ denotes the empirical measure of the particle system, and δ is the Dirac measure. We again focus on the drift term defined here as $\mathbf{b} : \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d) \to \mathbb{R}^d$ and treat the diffusion σ as constant or zero. Note that $\mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)$ is the space of probability measures on $\mathbb{R}^d \times \mathbb{R}^d$.

Our observations are positions and velocities along M trajectories at the times $0 = t_0 < t_1 < \cdots < t_L = T$:

$$(\mathbf{X}_{\text{tr}}, \mathbf{V}_{\text{tr}}) := \left(\{\mathbf{X}_{t_l,m}^i\}_{i=1,l=0,m=1}^{N,L,M}, \{\mathbf{V}_{t_l,m}^i\}_{i=1,l=0,m=1}^{N,L,M}\right).$$

The accelerations $\mathbf{A}_{t_l,m}^i$ are approximated using first-order finite differences:

$$\mathbf{A}_{t_l,m}^i := \frac{\mathbf{V}_{t_{\ell+1},m}^i - \mathbf{V}_{t_\ell,m}^i}{t_{\ell+1} - t_\ell}, \qquad \ell = 0, \cdots, L-1.$$

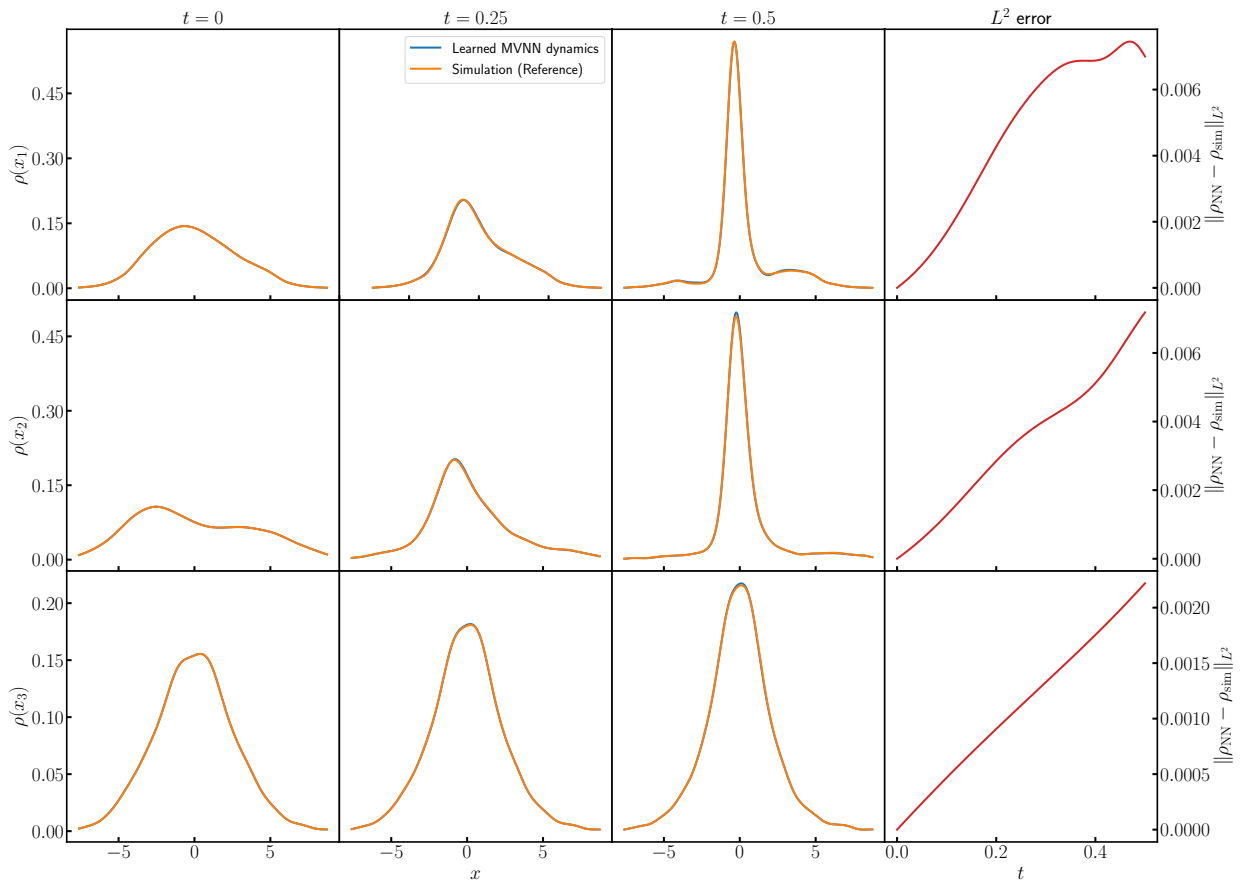


Figure 10: **Hierarchical dynamics:** Initial condition 1. Evolution of the multi-group system initialized with spatially separated populations. The rows correspond to Group 1, Group 2, and Group 3. The learned model (blue) faithfully reproduces the reference particle dynamics (orange), capturing the directional information flow from Group 3 down to Group 1.

For each trajectory m, the initial conditions $\left(\mathbf{X}_{0,m}^i, \mathbf{V}_{0,m}^i\right)$ are independently sampled from the probability measure $\mu_{0,m}$ for $i = 1, \cdots, N$. Our goal is to learn the mean field interaction kernels $\mathbf{b}(x, v, \mu)$ from the observations.

4.1 Measure-Valued Neural Network

Similar to the first order dynamics, we consider approximating $\mathbf{b}$ by a composition of two neural networks, φ_{emb} and φ_{int}:

$$\mathbf{b}_\theta(\mathbf{x}, \mathbf{v}, \mu) \approx \varphi_{\text{int}}\left(\mathbf{x}, \mathbf{v}, \left\langle \varphi_{\text{emb}}(\cdot, \cdot; \theta_{\text{emb}}), \mu \right\rangle; \theta_{\text{int}}\right),$$

where $\mu \in \mathcal{P}_2(\mathbb{R}^d \times \mathbb{R}^d)$, $\varphi_{\text{emb}}(\cdot, \cdot; \theta_{\text{emb}}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^k$ maps particle positions and velocities to their feature representations, and $\varphi_{\text{int}}(\cdot, \cdot, \cdot; \theta_{\text{int}}) : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^k \to \mathbb{R}^d$ learns the drift as a function of the position, velocity, and feature embedding $\langle \varphi_{\text{emb}}, \mu \rangle$. Here $\mathcal{P}_2(\mathbb{R}^d \times \mathbb{R}^d)$ is the space of probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with finite second moments:

$$\mathcal{P}_2(\mathbb{R}^d \times \mathbb{R}^d) = \left\{ \mu \text{ probability measure on } \mathbb{R}^d \times \mathbb{R}^d \;\middle|\; \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 d\mu(\mathbf{x}, \mathbf{y}) < \infty \right\}.$$

We have that for the empirical measure:

$$\langle \varphi_{\text{emb}}, \mu_t^N \rangle = \frac{1}{N} \sum_{j=1}^{N} \varphi_{\text{emb}}\left(\mathbf{X}_t^{j,N}, \mathbf{V}_t^{j,N}; \theta_{\text{emb}}\right),$$

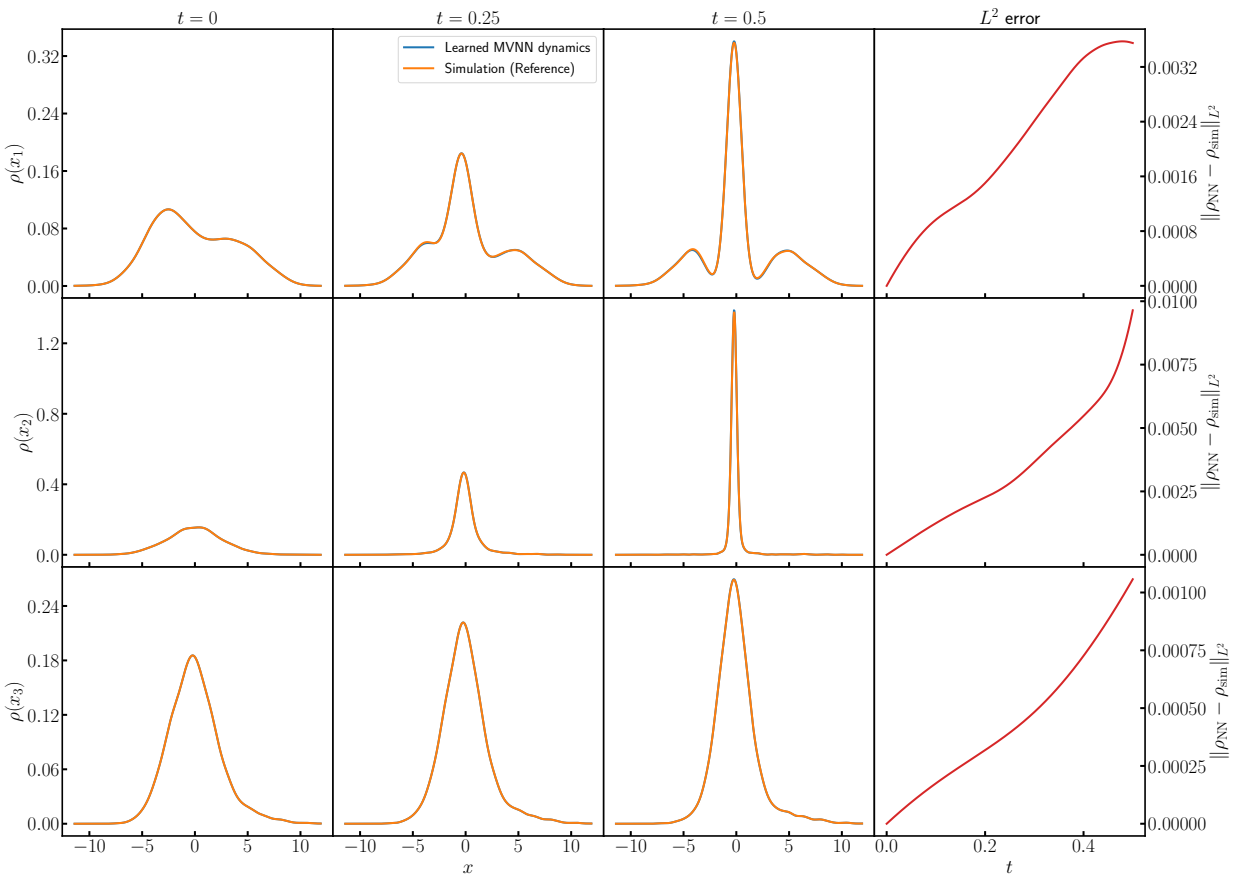


Figure 11: **Hierarchical dynamics:** Initial condition 2. Evolution of the multi-group system initialized with spatially separated populations. The rows correspond to Group 1, Group 2, and Group 3. The learned model (blue) accurately reproduces the reference particle dynamics (orange), capturing the directional information flow from Group 3 down to Group 1.

hence,

$$\mathbf{b}_\theta(\mathbf{X}, \mathbf{V}, \mu_t^N) \approx \varphi_{\text{int}}\left(\mathbf{X}, \mathbf{V}, \frac{1}{N}\sum_{j=1}^{N} \varphi_{\text{emb}}(\mathbf{X}_t^{j,N}, \mathbf{V}_t^{j,N}; \theta_{\text{emb}}); \theta_{\text{int}}\right).$$

The dynamics of the corresponding N-particle system are described by:

$$\begin{aligned}
\mathrm{d}\mathbf{X}_t^{\theta,i,N} &= \mathbf{V}_t^{\theta,i,N} dt \\
\mathrm{d}\mathbf{V}_t^{\theta,i,N} &= \mathbf{b}_\theta\left(\mathbf{X}_t^{\theta,i,N}, \mathbf{V}_t^{\theta,i,N,}, \mu_t^{\theta,N}\right) \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t^{i,N},
\end{aligned}$$

where $\mu_t^{\theta,N} = \frac{1}{N}\sum_{i=1}^{N} \delta_{(\mathbf{X}_t^{\theta,i,N}, \mathbf{V}_t^{\theta,i,N})}$ is the empirical measure of the particle system. The limiting McKean-Vlasov stochastic differential equation for a single representative particle with position $\mathbf{X}_t^\theta$ and velocity $\mathbf{V}_t^\theta$ is:

$$\begin{aligned}
\mathrm{d}\mathbf{X}_t^\theta &= \mathbf{V}_t^\theta \mathrm{d}t \\
\mathrm{d}\mathbf{V}_t^\theta &= \mathbf{b}_\theta\left(\mathbf{X}_t^\theta, \mathbf{V}_t^\theta, f_t^\theta\right) \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \\
&= \varphi_{\text{int}}\left(\mathbf{X}_t^\theta, \mathbf{V}_t^\theta, \int \varphi_{\text{emb}}(\mathbf{x}, \mathbf{v}) f_t^\theta(\mathbf{x}, \mathbf{v}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}\right) \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t,
\end{aligned} \tag{12}$$

where $f_t^\theta = \mathcal{L}\left(\mathbf{X}_t^\theta, \mathbf{V}_t^\theta\right)$ is the law of the position-velocity pair $\left(\mathbf{X}_t^\theta, \mathbf{V}_t^\theta\right)$ at time t. The Fokker-Planck equation for $(f_t^\theta)_{t\geq 0}$ in weak form is given by:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\langle f_t^\theta, \psi\rangle &= \left\langle f_t^\theta, \mathbf{v}\cdot\nabla_{\mathbf{x}}\psi + \mathbf{b}_\theta(\mathbf{x},\mathbf{v},f_t)\cdot\nabla_{\mathbf{v}}\psi + \frac{1}{2}\sigma^2\Delta_{\mathbf{v}}\psi\right\rangle \\ &= \left\langle f_t^\theta, \mathbf{v}\cdot\nabla_{\mathbf{x}}\psi + \varphi_{\text{int}}\left(\mathbf{x},\mathbf{v},\int\varphi_{\text{emb}}(\mathbf{x},\mathbf{v})f_t^\theta(\mathbf{x},\mathbf{v})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{v}\right)\cdot\nabla_{\mathbf{v}}\psi + \frac{1}{2}\sigma^2\Delta_{\mathbf{v}}\psi\right\rangle,\end{aligned}$$

for any smooth test function $\psi : \mathbb{R}^d \to \mathbb{R}$ with compact support.

4.2 Learning Objective and Optimization

Given the observations

$$\mathcal{D}_{\text{obs}} = \left\{\left(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \mathbf{A}_{t_l,m}^i\right)\right\}_{i=1,l=0,m=1}^{N,L,M},$$

the predicted mean-field drift from the MVNN is:

$$\hat{\mathbf{b}_\theta}\left(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \hat{\mu}_{t_l,m}\right) = \varphi_{\text{int}}\left(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \frac{1}{N}\sum_{j=1}^N \varphi_{\text{emb}}(\mathbf{X}_{t_l,m}^j, \mathbf{V}_{t_l,m}^j; \theta_{\text{emb}}); \theta_{\text{int}}\right).$$

We learn the model parameters $\theta = (\theta_{\text{int}}, \theta_{\text{emb}})$ by minimizing the differences between the predicted and observed trajectories. Let $\mathbb{P}^\theta$ be the measure induced by the solution $\left(\mathbf{X}_s^\theta, \mathbf{V}_s^\theta\right)_{s\in[0,t]}$ of the McKean-Vlasov SDE (12). By Girsanov's theorem, assuming the usual regularity conditions, we have the log-likelihood function

$$\begin{aligned}\mathcal{L}_t(\theta) := \log\frac{\mathrm{d}\mathbb{P}^\theta}{\mathrm{d}\mathbb{P}^{\mathbf{b}}} &= \int_0^t \left\langle \hat{\mathbf{b}}_\theta\left(\mathbf{X}_s, \mathbf{V}_s, \mu_s^\theta\right) - \mathbf{b}\left(\mathbf{X}_s, \mathbf{V}_s, \mu_s\right), \left(\sigma(\mathbf{X}_s,\mathbf{V}_s)\sigma(\mathbf{X}_s,\mathbf{V}_s)^\top\right)^{-1}\mathrm{d}\mathbf{V}_s\right\rangle \\ &\quad - \frac{1}{2}\int_0^t \left(\|\sigma(\mathbf{X}_s,\mathbf{V}_s)^{-1}\hat{\mathbf{b}}_\theta(\mathbf{X}_s,\mathbf{V}_s,\mu_s^\theta)\|^2 - \|\sigma(\mathbf{X}_s,\mathbf{V}_s)^{-1}\mathbf{b}(\mathbf{X}_s,\mathbf{V}_s,\mu_s)\|^2\right)\mathrm{d}s.\end{aligned}$$

We assume $\sigma(x,v) \equiv \sigma\mathbf{I}$, where $\sigma > 0$ is a constant, and approximate the stochastic integral using the Euler-Maruyama method to get the discrete log likelihood:

$$\begin{aligned}\hat{\mathcal{L}}(\theta) =& \frac{1}{MLN}\sum_{m=1}^M\sum_{l=0}^{L-1}\sum_{i=1}^N\Bigg[\left\langle\hat{\mathbf{b}}_\theta(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \hat{\mu}_{t_l,m}) - \mathbf{b}(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \hat{\mu}_{t_l,m}), (\sigma\sigma^\top)^{-1}\Delta\mathbf{V}_{t_l,m}^i\right\rangle \\ &- \frac{1}{2}\left(\|\sigma^{-1}\hat{\mathbf{b}}_\theta(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \hat{\mu}_{t_l,m})\|^2 - \|\sigma^{-1}\mathbf{b}(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \hat{\mu}_{t_l,m})\|^2\right)\Delta t\Bigg],\end{aligned}$$

where $\hat{\mu}_{t_l,m} = \frac{1}{N}\sum_{j=1}^N \delta_{\left(\mathbf{X}_{t_l,m}^j, \mathbf{V}_{t_l,m}^j\right)}$ is the empirical measure for trajectory m at time t_l, and $\Delta\mathbf{V}_{t_l,m}^i := \mathbf{V}_{t_{l+1},m}^i - \mathbf{V}_{t_l,m}^i$. Dropping terms independent of θ yields:

$$\hat{\mathcal{L}}(\theta) = -\frac{\Delta t}{2\sigma^2}\frac{1}{MLN}\sum_{m=1}^M\sum_{l=0}^{L-1}\sum_{i=1}^N\left\|\hat{\mathbf{b}}_\theta(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \hat{\mu}_{t_l,m}) - \mathbf{A}_{t_l,m}^i\right\|^2 + C.$$

Hence, maximizing the log likelihood $\hat{\mathcal{L}}(\theta)$ is equivalent to minimizing the mean-squared error loss:

$$\theta^* \in \arg\min_\theta \frac{1}{MLN}\sum_{m=1}^M\sum_{l=0}^{L-1}\sum_{i=1}^N\left\|\mathbf{A}_{t_l,m}^i - \hat{\mathbf{b}}_\theta(\mathbf{X}_{t_l,m}^i, \mathbf{V}_{t_l,m}^i, \hat{\mu}_{t_l,m})\right\|^2. \tag{13}$$

The numerical optimization of (13) is performed using Adam [80] with mini-batches.

4.3 Numerical Result

We present numerical experiments that validate the accuracy of our MVNN model for second-order systems and show its ability to generalize to unseen initial position configurations. We compare the dynamics from the learned mean-field model to the true dynamics from reference particle simulations.

4.3.1 Second-Order Attraction-Repulsion Swarming Model

We consider the second-order attraction repulsion swarming model in two dimensions:

$$\ddot{\mathbf{X}}_t^i = \frac{1}{N}\sum_{j=1}^{N} \phi\left(\|\mathbf{X}_t^j - \mathbf{X}_t^i\|\right)(\mathbf{X}_t^j - \mathbf{X}_t^i), \qquad i = 1, \ldots, N,$$

where $\mathbf{X}_t^i \in \mathbb{R}^2$ denotes the position of agent i, and:

$$\phi(r) = c_{\text{rep}} \exp(-(r/\ell_{\text{rep}})^2) - c_{\text{att}} \exp(-(r/\ell_{\text{att}})^2)$$

with $c_{\text{rep}} = 1.0$, $\ell_{\text{rep}} = 0.5$, $c_{\text{att}} = 0.7$, and $\ell_{\text{at}} = 2.0$. We use the forward Euler method to simulate 100 trajectories with $N = 16{,}000$ agents and 200 steps with time step $\Delta t = 10^{-2}$. We construct our training set with 80% of our initial positions sampled from a noisy annulus with random radius and width and the remaining 20% of the initial positions generated from a noisy double annuli with identical random radius and width. In particular, for each trajectory $m \in \{1, \cdots, 80\}$, the initial position of agent i is sampled using polar coordinates with additive Gaussian noise:

$$\Theta_i \sim \mathcal{U}(0, 2\pi), \quad \rho_i \sim \mathcal{U}\big(R_0 - \tfrac{W}{2},\, R_0 + \tfrac{W}{2}\big),$$
$$\mathbf{X}_0^i = \rho_i\big(\cos\Theta_i, \sin\Theta_i\big) + \varepsilon_i, \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(\mathbf{0},\, \sigma_0^2 I_2),$$

where $\mathbf{I}_2$ is the 2×2 identity matrix. For each trajectory $m \in \{81, \cdots, 100\}$, the initial position of half the agents, i.e. agents $i \in \{1, \cdots, 8000\}$, is generated from:

$$\Theta_i \sim \mathcal{U}(0, 2\pi), \quad \rho_i \sim \mathcal{U}\big(R_0 - \tfrac{W}{2},\, R_0 + \tfrac{W}{2}\big),$$
$$\mathbf{X}_0^i = (0.5, 0.5) + \rho_i\big(\cos\Theta_i, \sin\Theta_i\big) + \varepsilon_i, \quad \text{with} \quad \varepsilon_i \sim \mathcal{N}(\mathbf{0},\, \sigma_0^2 I_2),$$

and the initial position of agents $i \in \{8001, \cdots, 16000\}$ takes the value:

$$\mathbf{X}_0^i = (-0.5, -0.5) + \rho_{i-800}\big(\cos\Theta_{i-800}, \sin\Theta_{i-800}\big) + \varepsilon_{i-800}.$$

All initial velocities for the training samples are drawn from $\mathcal{N}(\mathbf{0}, 0.25\mathbf{I}_2)$. Figure 12 displays the evolution of the particle system with initial position generated from a ring distribution and initial velocity drawn from a Gaussian distribution, with a comparison between the true dynamics and the learned mean-field dynamics. Figure 13 shows the dynamics of the system with initial position drawn from a double ring distribution and Gaussian initial velocity, again comparing the true versus learned dynamics.

We also evaluate the generalization capability of the learned MVNN by testing the model on initial position distributions that differ from those in the training set. We consider test cases with initial positions generated from a uniform disk and a binary distribution with heterogeneous density. We still sample the initial velocities from a Gaussian distribution. The true evolution of the particle system compared to the evolution predicted by the learned MVNN is shown in Figures 14 and 15.

4.3.2 Second-Order Cucker Smale Model

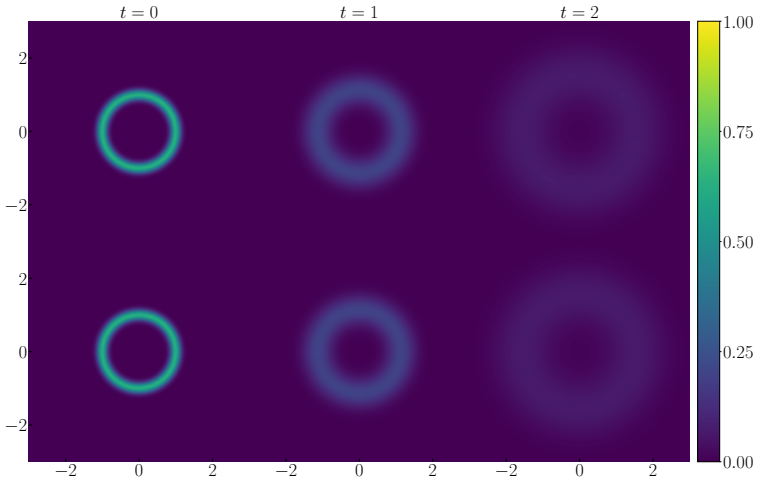
We also consider the two-dimensional second-order Cucker-Smale model, which involves velocity alignment rather than distance-based interactions:

$$\ddot{\mathbf{X}}_t^i = \frac{1}{N}\sum_{j=1}^{N} \phi\left(\|\mathbf{X}_t^j - \mathbf{X}_t^i\|\right)(\dot{\mathbf{X}}_t^j - \dot{\mathbf{X}}_t^i), \qquad i = 1, \ldots, N,$$

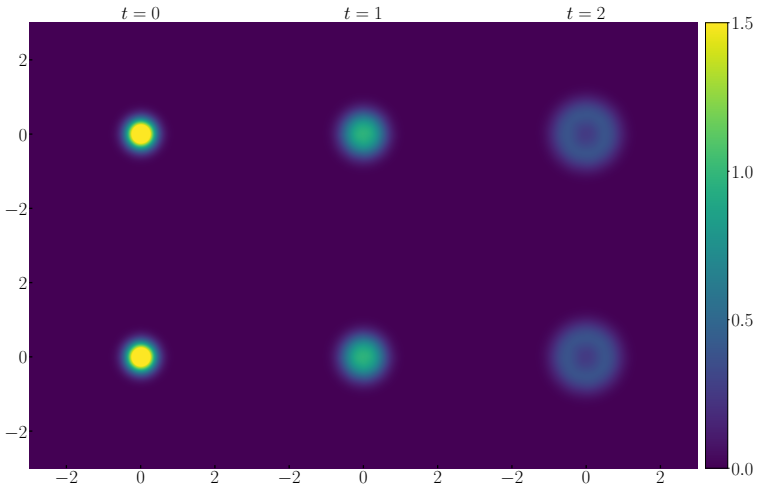
where $\mathbf{X}_t^i \in \mathbb{R}^2$ denotes the position of agent i, and:

$$\phi(r) = c_{\text{rep}} \exp(-(r/\ell_{\text{rep}})^2) - c_{\text{att}} \exp(-(r/\ell_{\text{att}})^2)$$

with $c_{\text{rep}} = 1.0$, $\ell_{\text{rep}} = 0.5$, $c_{\text{att}} = 0.7$, and $\ell_{\text{at}} = 2.0$. For our simulations, we use forward Euler to generate 100 trajectories for $N = 16{,}000$ particles with 200 time steps and $\Delta t = 10^{-2}$. Our training set is formed by sampling 80% of the initial positions from a Gaussian distribution with random scaling and 20% of the initial positions from a two-component Gaussian mixture that is also randomly scaled. Specifically, for trajectories $m \in \{1, \cdots, 80\}$, the initial positions are generated from $\mathcal{N}(\mu, (s_m\sigma)^2\mathbf{I}_2)$, where $s_m \sim \mathcal{U}(s_{\min}, s_{\max})$. For trajectories $m \in \{81, \cdots, 100\}$, the initial position for half the agents is drawn from $\mathcal{N}(\mu_1, (s_{m,1}\sigma_1)^2\mathbf{I}_2)$ with $s_{m,1} \sim \mathcal{U}(s_{\min}, s_{\max})$ and the initial position for the other half is sampled from $\mathcal{N}(\mu_2, (s_{m,2}\sigma_2)^2\mathbf{I}_2)$, where $s_{m,2} \sim \mathcal{U}(s_{\min}, s_{\max})$. All initial training velocities are drawn from $\mathcal{N}(\mathbf{0}, 0.25\mathbf{I}_2)$. Figure 16 visualizes the dynamics of the particle system with Gaussian initial position and velocity, and figure 17 shows the evolution of the system with initial position generated from a two-component Gaussian mixture and initial velocity drawn from a Gaussian distribution .



(a) Upper row: true positions, lower row: learned positions



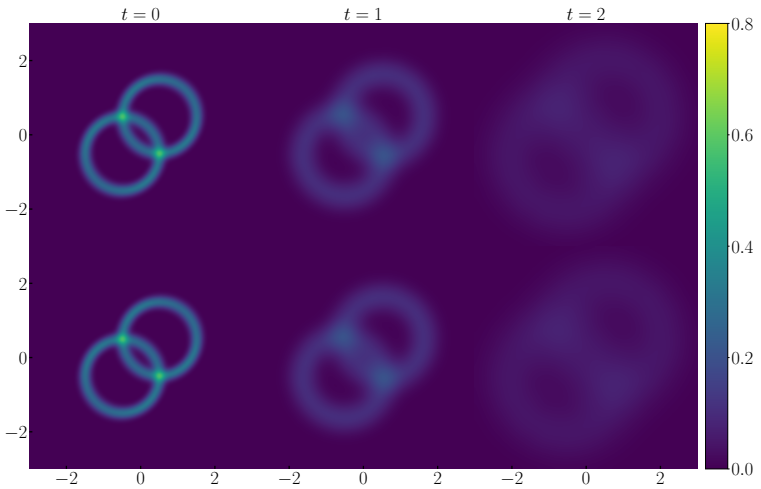
(b) Upper row: true velocities, lower row: learned velocities

Figure 12: **Second-order attraction-repulsion**: Evolution of the second-order attraction-repulsion model initialized with ring-shaped initial position and Gaussian initial velocity. The upper rows display ground truth particle positions and velocities, while the lower rows show the positions and velocities predicted by the learned MVNN.

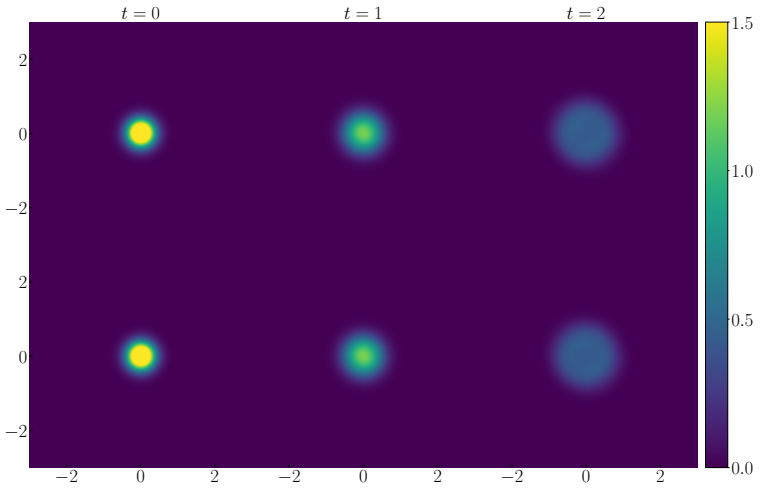
5 Discussion

We proposed a measure-valued neural network (MVNN) framework for estimating mean-field dynamics directly from particle-level observations of interacting agent systems. The approach generalizes neural network architectures to operate on probability measures, enabling the learning of measure-dependent drift terms that arise in the mean-field limit of large-scale interacting particle systems. Unlike mean-field approximations, our method provides a data-driven alternative that can efficiently infer complex mean-field interactions from empirical trajectory data without requiring explicit functional forms. We demonstrated through several representative examples, including Motsch-Tadmor dynamics, Cucker–Smale flocking, and hierarchical multi-group systems, that the proposed framework accurately captures the emergent collective behavior and generalizes well to unseen initial configurations. In particular, the multi-group extension of MVNN successfully recovers asymmetric and hierarchical communication structures among heterogeneous groups of agents. We also show that our framework can be extended to second-order systems. Comparisons with Gaussian process models demonstrate that our approach achieves greater efficiency and improved accuracy in predicting complex dynamics.

Our model can be viewed as a weak-form variant of operator learning [32, 49], conceptually related to but distinct from existing architectures such as DeepONet [41] and the Fourier Neural Operator [84]. DeepONet learns operators by



(a) Upper row: true positions, lower row: learned positions

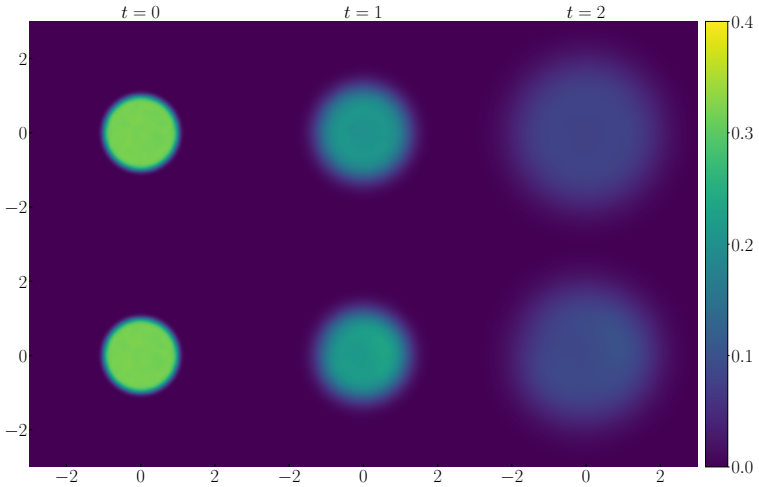


(b) Upper row: true velocities, lower row: learned velocities

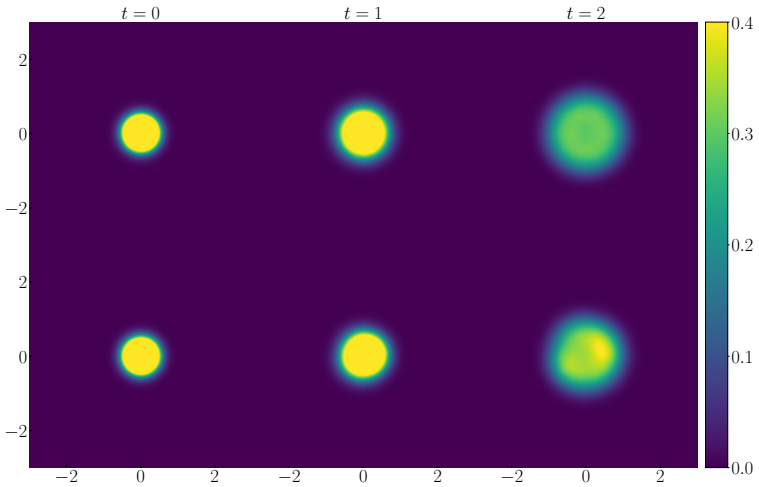
Figure 13: **Second-order attraction-repulsion**: Evolution of the second-order attraction-repulsion model initialized with double ring-shaped initial position and Gaussian initial velocity. The upper rows display ground truth particle positions and velocities, while the lower rows show the positions and velocities predicted by the learned MVNN.

mapping discrete pointwise evaluations of input functions to outputs, similar to finite difference schemes that rely on pointwise discretizations of differential operators. FNO, in contrast, represents input functions through their Fourier spectra, resembling spectral methods that approximate operators in a global frequency basis. Our proposed approach can be viewed as using the weak formulation to learn operators through integral constraints with test functions rather than pointwise values, akin to finite element methods in numerical analysis.

In future work, we note that the mean-field approximation is not always sufficient to capture the full range of interactions in complex systems. For instance, in plasma physics and dense particle systems, higher-order correlations play a significant role, and models such as the BBGKY hierarchy or kinetic closures are required to describe the coupled evolution of multi-particle distributions. Extending the proposed framework to learn higher-order interaction terms or reduced representations of two-particle distributions could therefore provide a promising pathway toward the data-driven discovery of beyond–mean-field dynamics. Such extensions would help bridge the gap between microscopic simulations and macroscopic kinetic models, potentially enabling efficient modeling of systems in which mean-field assumptions break down. This direction may also contribute to the development of foundation models for partial differential equations, capable of capturing multiscale structure and transferring across a broad class of dynamical systems.



(a) Upper row: true positions, lower row: learned positions



(b) Upper row: true velocities, lower row: learned velocities

Figure 14: **Second-order attraction-repulsion**: Evolution of the second-order attraction-repulsion model initialized with disk-shaped initial position and Gaussian initial velocity. The upper rows display ground truth particle positions and velocities, while the lower rows show the positions and velocities predicted by the learned MVNN.

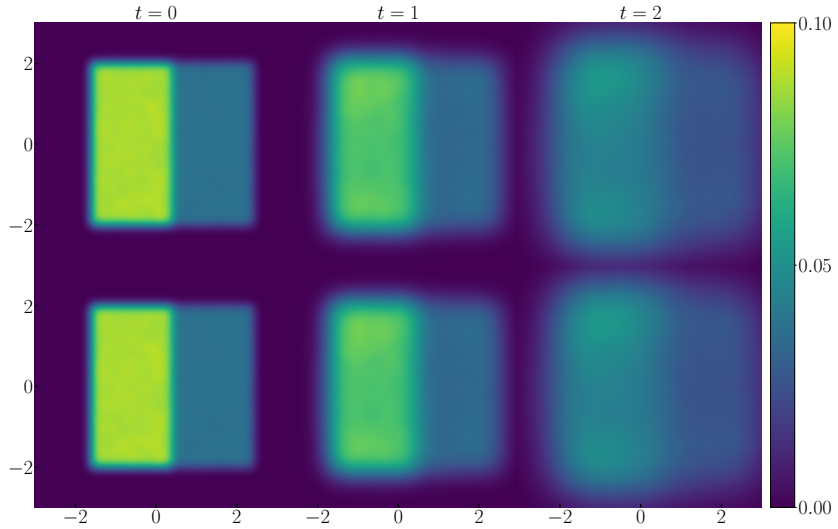
Acknowledgments

We would like to thank Adrien Weihs for the insightful discussions. This work used Anvil at Purdue through allocation MTH260007 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program[85], which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

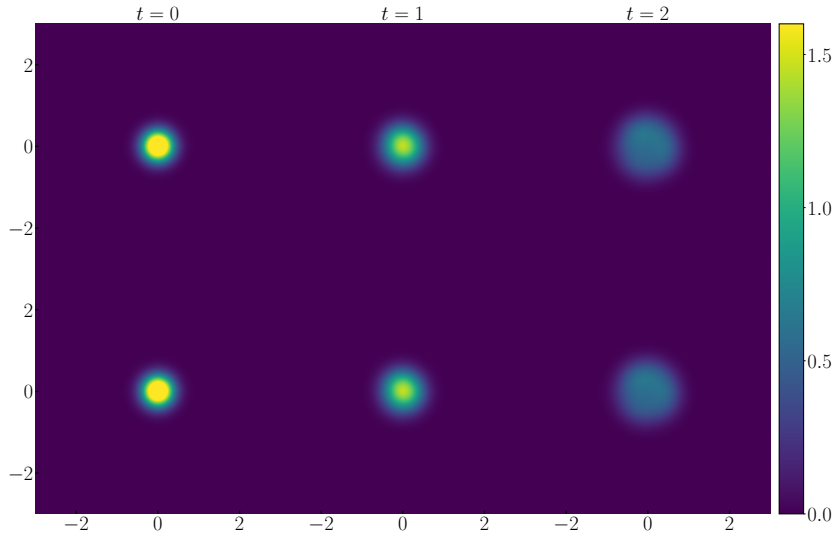
References

[1] José A Carrillo, Young-Pil Choi, and Sergio P Perez. A review on attractive–repulsive hydrodynamics for consensus in collective behavior. *Active Particles, Volume 1: Advances in Theory, Models, and Applications*, pages 259–298, 2017.

[2] Giacomo Albi, Nicola Bellomo, Luisa Fermo, S-Y Ha, Jeongho Kim, Lorenzo Pareschi, David Poyato, and Juan Soler. Vehicular traffic, crowds, and swarms: From kinetic theory and multiscale methods to applications and research perspectives. *Mathematical Models and Methods in Applied Sciences*, 29(10):1901–2005, 2019.



(a) Upper row: true positions, lower row: learned positions



(b) Upper row: true velocities, lower row: learned velocities

Figure 15: **Second-order attraction-repulsion**: Evolution of the second-order attraction-repulsion model initialized with binary asymmetric initial position (low density left, high density right) and Gaussian initial velocity. The upper rows display ground truth particle positions and velocities, while the lower rows show the positions and velocities predicted by the learned MVNN.

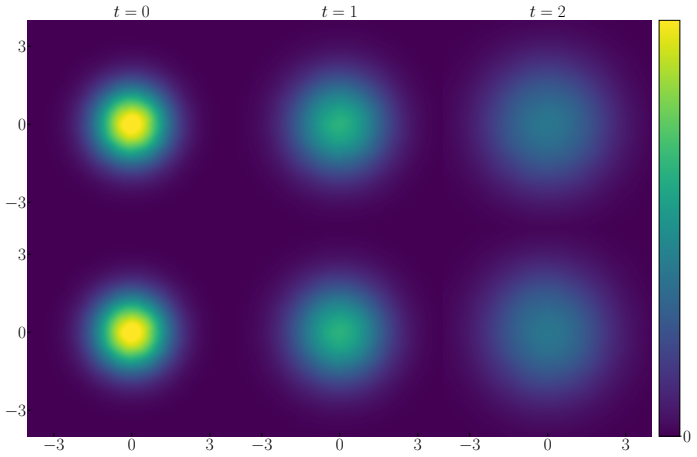
[3] Theodore Kolokolnikov, Hui Sun, David Uminsky, and Andrea L Bertozzi. Stability of ring patterns arising from two-dimensional particle interactions. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 84(1):015203, 2011.

[4] Tamás Vicsek and Anna Zafeiris. Collective motion. *Physics reports*, 517(3-4):71–140, 2012.

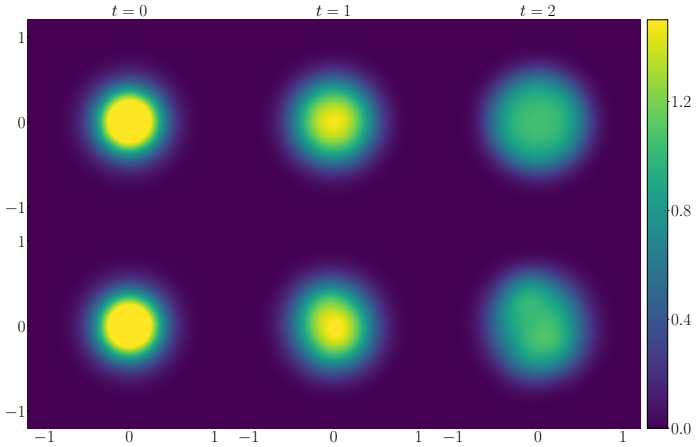
[5] Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[6] Fei Lu, Ming Zhong, Sui Tang, and Mauro Maggioni. Nonparametric inference of interaction laws in systems of agents from trajectory data. *Proceedings of the National Academy of Sciences*, 116(29):14424–14433, 2019.

[7] Yuxuan Liu, Scott G McCalla, and Hayden Schaeffer. Random feature models for learning interacting dynamical systems. *Proceedings of the Royal Society A*, 479(2275):20220835, 2023.



(a) Upper row: true positions, lower row: learned positions



(b) Upper row: true positions, lower row: learned positions

Figure 16: **Second-order Cucker-Smale**: Evolution of the second-order Cucker-Smale model initialized with Gaussian initial position and Gaussian initial velocity. The upper rows display ground truth particle positions and velocities, while the lower rows show the positions and velocities predicted by the learned MVNN.

[8] Quanjun Lang and Fei Lu. Learning interaction kernels in mean-field equations of first-order systems of interacting particles. *SIAM Journal on Scientific Computing*, 44(1):A260–A285, 2022.

[9] Mengyang Gu, Xinyi Fang, and Yimin Luo. Data-driven model construction for anisotropic dynamics of active matter. *PRX Life*, 1(1):013009, 2023.

[10] Weiqi Chu, Qin Li, and Mason A Porter. Inference of interaction kernels in mean-field models of opinion dynamics. *SIAM Journal on Applied Mathematics*, 84(3):1096–1115, 2024.

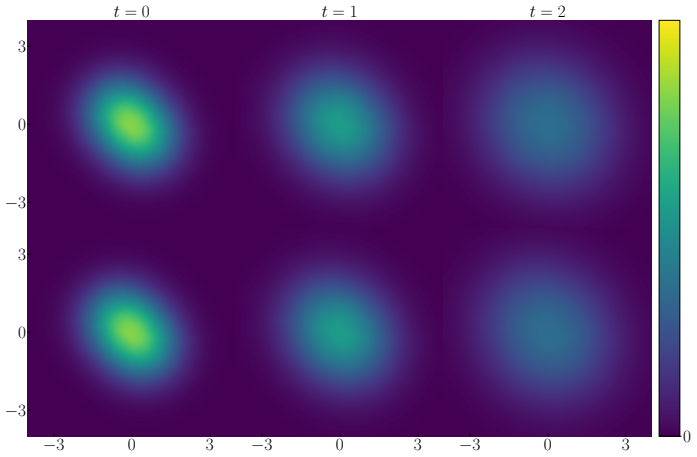
[11] Mauro Maggioni, Jason J Miller, Hongda Qiu, and Ming Zhong. Learning interaction kernels for agent systems on riemannian manifolds. In *International Conference on Machine Learning*, pages 7290–7300. PMLR, 2021.

[12] Bertrand Maury, Aude Roudneff-Chupin, and Filippo Santambrogio. A macroscopic crowd motion model of gradient flow type. *Mathematical Models and Methods in Applied Sciences*, 20(10):1787–1821, 2010.

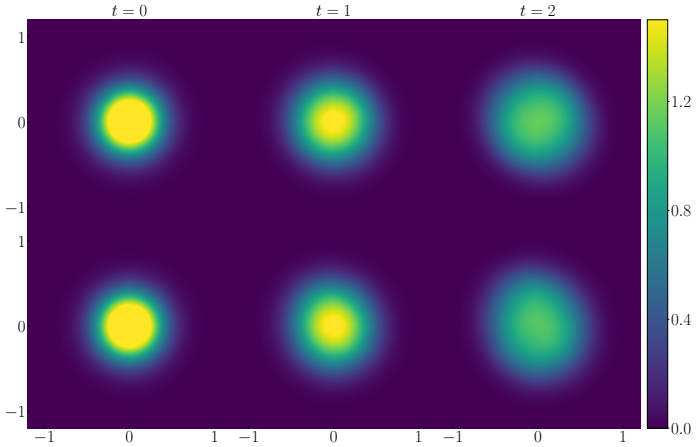
[13] Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proceedings of the royal society of london. series a. mathematical and physical sciences*, 229(1178):317–345, 1955.

[14] Evelyn F Keller and Lee A Segel. Initiation of slime mold aggregation viewed as an instability. *Journal of theoretical biology*, 26(3):399–415, 1970.

[15] Shi Jin, Lei Li, and Jian-Guo Liu. Random batch methods (rbm) for interacting particle systems. *Journal of Computational Physics*, 400:108877, 2020.



(a) Upper row: true positions, lower row: learned positions



(b) Upper row: true velocities, lower row: learned velocities

Figure 17: **Second-order Cucker-Smale**: Evolution of the second-order Cucker-Smale model initialized with a two-component Gaussian mixture initial position and Gaussian initial velocity. The upper rows display ground truth particle positions and velocities, while the lower rows show the positions and velocities predicted by the learned MVNN.

[16] Zecheng Gan, Xuanzhao Gao, Jiuyang Liang, and Zhenli Xu. Random batch ewald method for dielectrically confined coulomb systems. *SIAM Journal on Scientific Computing*, 47(4):B846–B874, 2025.

[17] Leo P Kadanoff. More is the same; phase transitions and mean field theories. *Journal of Statistical Physics*, 137(5):777–797, 2009.

[18] François Golse. The mean-field limit for the dynamics of large particle systems. *Journées équations aux dérivées partielles*, pages 1–47, 2003.

[19] Herbert Spohn. *Large scale dynamics of interacting particles*. Springer Science & Business Media, 2012.

[20] Didier Bresch, Pierre-Emmanuel Jabin, and Zhenfu Wang. Mean field limit and quantitative estimates with singular attractive kernels. *Duke Mathematical Journal*, 172(13):2591–2641, 2023.

[21] José A Carrillo, Massimo Fornasier, Giuseppe Toscani, and Francesco Vecil. Particle, kinetic, and hydrodynamic models of swarming. In *Mathematical modeling of collective behavior in socio-economic and life sciences*, pages 297–336. Springer, 2010.

[22] José A Carrillo, Massimo Fornasier, Jesús Rosado, and Giuseppe Toscani. Asymptotic flocking dynamics for the kinetic cucker–smale model. *SIAM Journal on Mathematical Analysis*, 42(1):218–236, 2010.

[23] Giacomo Albi, Lorenzo Pareschi, and Mattia Zanella. Boltzmann-type control of opinion consensus through leaders. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 372(2028):20140138, 2014.

[24] Sebastien Motsch and Eitan Tadmor. Heterophilious dynamics enhances consensus. *SIAM review*, 56(4):577–621, 2014.

[25] José A Carrillo, Young-Pil Choi, Claudia Totzeck, and Oliver Tse. An analytical framework for consensus-based global optimization method. *Mathematical Models and Methods in Applied Sciences*, 28(06):1037–1066, 2018.

[26] José A Carrillo, Shi Jin, Lei Li, and Yuhua Zhu. A consensus-based global optimization method for high dimensional machine learning problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 27:S5, 2021.

[27] Hui Huang and Jinniao Qiu. On the mean-field limit for the consensus-based optimization. *Mathematical Methods in the Applied Sciences*, 45(12):7814–7831, 2022.

[28] Liyao Lyu and Jingrun Chen. Consensus based stochastic optimal control. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[30] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[31] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[32] Hayden Schaeffer and Scott G McCalla. Sparse model selection via integral terms. *Physical Review E*, 96(2):023302, 2017.

[33] Hayden Schaeffer, Giang Tran, and Rachel Ward. Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78(6):3279–3295, 2018.

[34] Pinchen Xie, Roberto Car, and Weinan E. Ab initio generalized langevin equation. *Proceedings of the National Academy of Sciences*, 121(14):e2308668121, 2024.

[35] Liyao Lyu and Huan Lei. Construction of coarse-grained molecular dynamics with many-body non-markovian memory. *Physical Review Letters*, 131(17):177301, 2023.

[36] Pei Ge, Zhongqiang Zhang, and Huan Lei. Data-driven learning of the generalized langevin equation with state-dependent memory. *Physical Review Letters*, 133(7):077301, 2024.

[37] Yuan Chen and Dongbin Xiu. Learning stochastic dynamical system via flow map operator. *Journal of Computational Physics*, 508:112984, 2024.

[38] Yanfang Liu, Yuan Chen, Dongbin Xiu, and Guannan Zhang. A training-free conditional diffusion model for learning stochastic dynamical systems. *SIAM Journal on Scientific Computing*, 47(5):C1144–C1171, 2025.

[39] Kyongmin Yeo, Hyomin Shin, Heechang Kim, and Minseok Choi. Model-free learning of random dynamical system from noisy observations. *Journal of Computational Physics*, page 114474, 2025.

[40] Haoyang Zheng and Guang Lin. Les-sindy: Laplace-enhanced sparse identification of nonlinear dynamical systems. *Journal of Computational Physics*, page 114443, 2025.

[41] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[42] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*.

[43] Hayden Schaeffer, Russel Caflisch, Cory D Hauck, and Stanley Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, 110(17):6634–6639, 2013.

[44] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International conference on machine learning*, pages 3208–3216. PMLR, 2018.

[45] Victor Churchill, Yuan Chen, Zhongshu Xu, and Dongbin Xiu. Dnn modeling of partial differential equations with incomplete data. *Journal of Computational Physics*, 493:112502, 2023.

[46] Yifan Sun, Linan Zhang, and Hayden Schaeffer. Neupde: Neural network based ordinary and partial differential equations for modeling time-dependent data. In *Mathematical and Scientific Machine Learning*, pages 352–372. PMLR, 2020.

[47] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197), 2017.

[48] Hayden Schaeffer, Giang Tran, Rachel Ward, and Linan Zhang. Extracting structured dynamical systems using sparse optimization with very few samples. *Multiscale Modeling & Simulation*, 18(4):1435–1461, 2020.

[49] Daniel A Messenger and David M Bortz. Weak sindy for partial differential equations. *Journal of Computational Physics*, 443:110525, 2021.

[50] Junfeng Chen, Kailiang Wu, and Dongbin Xiu. Due: A deep learning framework and library for modeling unknown equations. *SIAM Review*, 67(4):873–902, 2025.

[51] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020.

[52] Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496:112549, 2024.

[53] Xinyue Yu and Hayden Schaeffer. Regularized random fourier features and finite element reconstruction for operator learning in sobolev space. *arXiv preprint arXiv:2512.17884*, 2025.

[54] Zecheng Zhang, Leung Wing Tat, and Hayden Schaeffer. Belnet: Basis enhanced learning, a mesh-free neural operator. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 479(2276), 2023.

[55] Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equations: Multioperator learning and extrapolation. *Physical Review E*, 111(3):035304, 2025.

[56] Min Zhu, Jingmin Sun, Zecheng Zhang, Hayden Schaeffer, and Lu Lu. Pi-mfm: Physics-informed multimodal foundation model for solving partial differential equations. *arXiv preprint arXiv:2512.23056*, 2025.

[57] Adrien Weihs, Jingmin Sun, Zecheng Zhang, and Hayden Schaeffer. A deep learning framework for multi-operator learning: Architectures and approximation theory. *arXiv preprint arXiv:2510.25379*, 2025.

[58] Hamda Hmida, Hsiu-Wen Chang Joly, and Youssef Mesri. Compno: A novel foundation model approach for solving partial differential equations. *Applied Sciences*, 16(2):972, 2026.

[59] Elisa Negrini, Yuxuan Liu, Liu Yang, Stanley J Osher, and Hayden Schaeffer. A multimodal pde foundation model for prediction and scientific text descriptions. *arXiv preprint arXiv:2502.06026*, 2025.

[60] Zhanhong Ye, Xiang Huang, Leheng Chen, Hongsheng Liu, Zidong Wang, and Bin Dong. Pdeformer: Towards a foundation model for one-dimensional partial differential equations. *arXiv preprint arXiv:2402.12652*, 2024.

[61] Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Prose: Predicting multiple operators and symbolic expressions using multimodal transformers. *Neural Networks*, 180:106707, 2024.

[62] Yadi Cao, Yuxuan Liu, Liu Yang, Rose Yu, Hayden Schaeffer, and Stanley Osher. Vicon: Vision in-context operator networks for multi-physics fluid dynamics prediction. *arXiv preprint arXiv:2411.16063*, 2024.

[63] Rudy Morel, Jiequn Han, and Edouard Oyallon. Disco: learning to discover an evolution operator for multi-physics-agnostic prediction. *arXiv preprint arXiv:2504.19496*, 2025.

[64] Liu Yang, Siting Liu, Tingwei Meng, and Stanley J Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023.

[65] Derek Jollie, Jingmin Sun, Zecheng Zhang, and Hayden Schaeffer. Time-series forecasting, knowledge distillation, and refinement within a multimodal pde foundation model. *arXiv preprint arXiv:2409.11609*, 2024.

[66] Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. Prose-fd: A multimodal pde foundation model for learning multiple operators for forecasting fluid dynamics. *arXiv preprint arXiv:2409.09811*, 2024.

[67] Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: Pde-conditional transformers towards universal neural pde solvers. *arXiv preprint arXiv:2405.17527*, 2024.

[68] Zhanhong Ye, Zining Liu, Bingyang Wu, Hongjie Jiang, Leheng Chen, Minyan Zhang, Xiang Huang, Qinghe Meng Zou, Hongsheng Liu, Bin Dong, et al. Pdeformer-2: A versatile foundation model for two-dimensional partial differential equations. *arXiv preprint arXiv:2507.15409*, 2025.

[69] Xin Guo, Huyên Pham, and Xiaoli Wei. Itô's formula for flows of measures on semimartingales. *Stochastic Processes and their applications*, 159:350–390, 2023.

[70] Philip E. Protter. *Stochastic Differential Equations*, pages 249–361. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[71] Alain-Sol Sznitman. Topics in propagation of chaos. In *Ecole d'été de probabilités de Saint-Flour XIX—1989*, pages 165–251. Springer, 2006.

[72] Louis-Pierre Chaintron and Antoine Diez. Propagation of chaos: a review of models, methods and applications. ii. applications. *arXiv preprint arXiv:2106.14812*, 2021.

[73] Henry P McKean et al. Propagation of chaos for a class of non-linear parabolic equations. *Stochastic Differential Equations (Lecture Series in Differential Equations, Session 7, Catholic Univ., 1967)*, pages 41–57, 1967.

[74] Daniel A Messenger and David M Bortz. Learning mean-field equations from particle data using wsindy. *Physica D: Nonlinear Phenomena*, 439:133406, 2022.

[75] Benjamin G Cohen, Burcu Beykal, and George M Bollas. Physics-informed genetic programming for discovery of partial differential equations from scarce and noisy data. *Journal of Computational Physics*, 514:113261, 2024.

[76] Huyên Pham and Xavier Warin. Mean-field neural networks: learning mappings on wasserstein space. *Neural Networks*, 168:380–393, 2023.

[77] Hao Liu, Zecheng Zhang, Wenjing Liao, and Hayden Schaeffer. Neural scaling laws of deep relu and deep operator network: A theoretical study. *arXiv preprint arXiv:2410.00357*, 2024.

[78] Jianghui Wen, Xiangjun Wang, Shuhua Mao, and Xinping Xiao. Maximum likelihood estimation of mckean–vlasov stochastic differential equation and its application. *Applied Mathematics and Computation*, 274:237–246, 2016.

[79] Louis Sharrock, Nikolas Kantas, Panos Parpas, and Grigorios A Pavliotis. Parameter estimation for the mckean-vlasov stochastic differential equation. *arXiv preprint arXiv:2106.13751*, 2021.

[80] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[81] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[82] Sebastien Motsch and Eitan Tadmor. A new model for self-organized dynamics and its flocking behavior. *Journal of Statistical Physics*, 144(5):923, 2011.

[83] Jinchao Feng, Charles Kulick, Yunxiang Ren, and Sui Tang. Learning particle swarming models from data with gaussian processes. *Mathematics of Computation*, 93, 11 2023.

[84] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *JMLR*, 24(1), 2023.

[85] Timothy J Boerner, Stephen Deems, Thomas R Furlani, Shelley L Knuth, and John Towns. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and experience in advanced research computing 2023: Computing for the common good*, pages 173–176. 2023.

[86] René Carmona. *Lectures on BSDEs, stochastic control, and stochastic differential games with financial applications*. SIAM, 2016.

[87] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural networks*, 94:103–114, 2017.

A Proof of Proposition 2.1

Proof. The proof is based on Theorem 1.7 in [86]. For simplicity, we denote the embedding and interaction networks by φ_{e} and φ_{i}, respectively. We only need to show that for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ and $\mu, \mu' \in \mathcal{P}$, we have:

$$|\mathbf{b}_\theta(\mathbf{x}, \mu) - \mathbf{b}_\theta(\mathbf{x}', \mu')| \leq C(|\mathbf{x} - \mathbf{x}'| + W_2(\mu, \mu')),$$

where W_2 denotes the Wasserstein-2 distance. Notice that, by the triangle inequality, we have:

$$|\mathbf{b}_\theta(\mathbf{x}, \mu) - \mathbf{b}_\theta(\mathbf{x}', \mu')| \leq |\mathbf{b}_\theta(\mathbf{x}, \mu) - \mathbf{b}_\theta(\mathbf{x}, \mu')| + |\mathbf{b}_\theta(\mathbf{x}, \mu') - \mathbf{b}_\theta(\mathbf{x}', \mu')|.$$

The first term is bounded by:

$$\begin{aligned}|\mathbf{b}_\theta(\mathbf{x}, \mu) - \mathbf{b}_\theta(\mathbf{x}, \mu')| =& \left|\varphi_{\mathrm{int}}\left(\mathbf{x}, \int \varphi_{\mathrm{e}}(\mathbf{y})\mu(\mathrm{d}\mathbf{y})\right) - \varphi_{\mathrm{int}}\left(\mathbf{x}, \int \varphi_{\mathrm{e}}(\mathbf{y})\mu'(\mathrm{d}\mathbf{y})\right)\right| \\ \leq& C_i \left|\int \varphi_{\mathrm{e}}(\mathbf{y})\mu(\mathrm{d}\mathbf{y}) - \int \varphi_{\mathrm{e}}(\mathbf{y})\mu'(\mathrm{d}\mathbf{y})\right|.\end{aligned}$$

For the second term, keeping the measure argument fixed at μ' and using the Lipschitz continuity of ϕ_{int} in its first variable, we obtain:

$$\begin{aligned}|\mathbf{b}_\theta(\mathbf{x},\mu') - \mathbf{b}_\theta(\mathbf{x}',\mu')| =& \left|\varphi_{\text{int}}\left(\mathbf{x}, \int \varphi_{\text{e}}(\mathbf{y})\mu'(\mathrm{d}\mathbf{y})\right) - \varphi_{\text{i}}\left(\mathbf{x}', \int \varphi_{\text{e}}(\mathbf{y})\mu'(\mathrm{d}\mathbf{y})\right)\right| \\ \leq& C_i\left|\mathbf{x}-\mathbf{x}'\right|.\end{aligned}$$

For the first term, we notice that for any coupling $\pi \in \Pi(\mu,\mu')$, whose marginals are μ and μ', i.e.

$$\begin{aligned}\int_{\mathbb{R}^d\times\mathbb{R}^d} \phi(x)\,\pi(\mathrm{d}x,\mathrm{d}y) &= \int_{\mathbb{R}^d} \phi(x)\,\mu(\mathrm{d}x), \\ \int_{\mathbb{R}^d\times\mathbb{R}^d} \psi(y)\,\pi(\mathrm{d}x,\mathrm{d}y) &= \int_{\mathbb{R}^d} \psi(y)\,\mu'(\mathrm{d}y).\end{aligned}$$

For all π, we have

$$\begin{aligned}\left|\int \varphi_{\text{e}}(\mathbf{y})\mu(\mathrm{d}\mathbf{y}) - \int \varphi_{\text{e}}(\mathbf{y})\mu'(\mathrm{d}\mathbf{y})\right| =& \left|\int \varphi_{\text{e}}(\mathbf{y}) - \varphi_{\text{e}}(\mathbf{y}')\pi(\mathrm{d}\mathbf{y},\mathrm{d}\mathbf{y}')\right| \\ \leq& \left(\int \left(\varphi_{\text{e}}(\mathbf{y}) - \varphi_{\text{e}}(\mathbf{y}')\right)^2 \pi(\mathrm{d}\mathbf{y},\mathrm{d}\mathbf{y}')\right)^{\frac{1}{2}} \\ \leq& C_e\left(\int \left(\mathbf{y}-\mathbf{y}'\right)^2 \pi(\mathrm{d}\mathbf{y},\mathrm{d}\mathbf{y}')\right)^{\frac{1}{2}}.\end{aligned}$$

Taking the π that minimize the $\int |\mathbf{y}-\mathbf{y}'|^2\,\pi(\mathrm{d}\mathbf{y},\mathrm{d}\mathbf{y}')$, yields:

$$|\mathbf{b}_\theta(\mathbf{x},\mu) - \mathbf{b}_\theta(\mathbf{x},\mu')| \leq C_i C_e W_2(\mu,\mu').$$

Therefore, we obtain a Lipschitz bound on the drift:

$$|\mathbf{b}_\theta(\mathbf{x},\mu) - \mathbf{b}_\theta(\mathbf{x}',\mu')| \leq C(|\mathbf{x}-\mathbf{x}'| + W_2(\mu,\mu')).$$

□

B Proof of Proposition 2.2

Proof. The proof is based on a synchronous coupling argument [71]. We construct the N-particle system and a set of N independent "ideal" mean-field processes driven by the same Brownian motions and show that their L^2 distance vanishes as $N\to\infty$. Let us define N independent processes $\bar{\mathcal{X}}_t^{\theta,N} = \left(\bar{\mathbf{X}}_t^{\theta,1,N},\cdots,\bar{\mathbf{X}}_t^{\theta,N,N}\right)$ defined as the solution of N SDEs:

$$\begin{aligned}\mathrm{d}\bar{\mathbf{X}}_t^{\theta,i,N} =& \mathbf{b}_\theta(\bar{\mathbf{X}}_t^{\theta,i,N})\mathrm{d}t + \sigma\mathrm{d}\mathbf{B}_t^i \\ =& \varphi_{\text{int}}\left(\bar{\mathbf{X}}_t^{\theta,i,N}, \int \varphi_{\text{e}}(\mathbf{y}) f_t^\theta(\mathrm{d}\mathbf{y})\right)\mathrm{d}t + \sigma\mathrm{d}\mathbf{B}_t^i, \quad i\in\{1,\cdots,N\},\end{aligned}$$

where $(\mathbf{B}_t^i)$ is the same Brownian motion and where we call that $f_t^\theta = \text{Law}(\mathbf{X}_t^{\theta,i})$. Since the Brownian motions are independent, the law f_t^θ is independent of index i. We will show that:

$$\frac{1}{N}\sum_{i=1}^N \mathbb{E}\left[\sup_{t\leq T}\left|\mathbf{X}_t^{\theta,i,N} - \bar{\mathbf{X}}_t^{\theta,i,N}\right|^2\right] \leq \epsilon(N,T). \tag{14}$$

Fix $i=1$ and define the path-space coupling

$$\pi_N := \text{Law}\Big((\mathbf{X}_t^{\theta,1,N})_{t\in[0,T]}, (\bar{\mathbf{X}}_t^{\theta,1,N})_{t\in[0,T]}\Big).$$

Its marginals are $f_{[0,T]}^{1,\theta,N}$ and $f_{[0,T]}^\theta$, respectively. Therefore, by the definition of the Wasserstein distance on path space,

$$\begin{aligned}W_2^2\left(f_{[0,T]}^{1,\theta,N}, f_{[0,T]}^\theta\right) \leq& \int \sup_{0\leq t\leq T}|\mathbf{x}_t - \mathbf{y}_t|^2\,\pi_N(\mathrm{d}\mathbf{x},\mathrm{d}\mathbf{y}) \\ =& \mathbb{E}\left[\sup_{0\leq t\leq T}\left|\mathbf{X}_t^{\theta,1,N} - \bar{\mathbf{X}}_t^{\theta,1,N}\right|^2\right].\end{aligned}$$

Since the right-hand side of Equation (14) tends to 0 as $N \to \infty$, we conclude that

$$W_2\Big(f^{1,\theta,N}_{[0,T]}, f^{\theta}_{[0,T]}\Big) \to 0.$$

Moreover, by exchangeability, the same estimate holds for any fixed finite collection of particles, which yields $f^{\theta}_{[0,T]}$-chaoticity of the learned particle system. Using the Ito formula, and since the stochastic terms cancel due to the synchronous coupling:

$$\left|\mathbf{X}^{\theta,i,N}_t - \bar{\mathbf{X}}^{\theta,i,N}_t\right|^2 = 2\int_0^t \left\langle \mathbf{X}^{\theta,i,N}_s - \bar{\mathbf{X}}^{\theta,i,N}_s, \mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right\rangle \mathrm{d}s,$$

where $e^i_s = \mathbf{X}^{\theta,i,N}_s - \bar{\mathbf{X}}^{\theta,i,N}_s$. Taking the supremum and then expectation:

$$\begin{aligned}
&\mathbb{E}\left[\sup_{t\le T}\left|\mathbf{X}^{\theta,i,N}_t - \bar{\mathbf{X}}^{\theta,i,N}_t\right|^2\right] \\
&= \mathbb{E}\left[\sup_{t\le T}\left|2\int_0^t \left\langle \mathbf{X}^{\theta,i,N}_s - \bar{\mathbf{X}}^{\theta,i,N}_s, \mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right\rangle \mathrm{d}s\right|\right] \\
&\le 2\int_0^T \mathbb{E}\left[\left|\left\langle \mathbf{X}^{\theta,i,N}_s - \bar{\mathbf{X}}^{\theta,i,N}_s, \mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right\rangle\right|\right] \mathrm{d}s \\
&\le \int_0^T \mathbb{E}\left[\left|\mathbf{X}^{\theta,i,N}_s - \bar{\mathbf{X}}^{\theta,i,N}_s\right|^2\right] \mathrm{d}s + \int_0^T \mathbb{E}\left[\left|\mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right|^2\right] \mathrm{d}s \\
&\le \int_0^T \mathbb{E}\left[\sup_{r\le s}\left|\mathbf{X}^{\theta,i,N}_r - \bar{\mathbf{X}}^{\theta,i,N}_r\right|^2\right] \mathrm{d}s + \int_0^T \mathbb{E}\left[\left|\mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right|^2\right] \mathrm{d}s.
\end{aligned}$$

The drift term can be split into two terms as follows:

$$\begin{aligned}
&\mathbb{E}\left[\left|\mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right|^2\right] \\
&\quad\le 2\mathbb{E}\left[\left|\mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, \bar{\mu}^{\theta,N}_s)\right|^2\right] + 2\mathbb{E}\left[\left|\mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, \bar{\mu}^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right|^2\right],
\end{aligned}$$

where $\bar{\mu}^{\theta,N}_t = \frac{1}{N}\sum_{i=1}^N \delta_{\bar{\mathbf{X}}^{\theta,i,N}_t}$. For the first term, we have:

$$\begin{aligned}
\mathbb{E}\left[\left|\mathbf{b}_\theta(\mathbf{X}^{\theta,i,N}_s, \mu^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, \bar{\mu}^{\theta,N}_s)\right|^2\right] &\le 2C_i^2\mathbb{E}\left[\left(\mathbf{X}^{\theta,i,N}_s - \bar{\mathbf{X}}^{\theta,i,N}_s\right)^2\right] + 2C_i^2C_e^2\mathbb{E}\left[\left(W_2\left(\mu^{\theta,N}_s, \bar{\mu}^{\theta,N}_s\right)\right)^2\right] \\
&\le 2C_i^2(1+C_e^2)\mathbb{E}\left[\left|\mathbf{X}^{\theta,i,N}_s - \bar{\mathbf{X}}^{\theta,i,N}_s\right|^2\right],
\end{aligned}$$

and for the second term:

$$\begin{aligned}
&\mathbb{E}\left[\left|\mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, \bar{\mu}^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right|^2\right] \\
&\le (C_iC_e)^2\mathbb{E}\left[\left|\int \varphi_e(\mathbf{x})\bar{\mu}^{\theta,N}_s(\mathrm{d}\mathbf{x}) - \int \varphi_e(\mathbf{x}) f^{\theta}_s(\mathbf{x})\mathrm{d}\mathbf{x}\right|^2\right] \\
&\le (C_iC_e)^2\mathbb{E}\left[\left|\frac{1}{N}\sum_{j=1}^N \varphi_e(\bar{\mathbf{X}}^{\theta,j,N}_s) - \mathbb{E}[\varphi_e(\bar{\mathbf{X}}^{\theta,j,N}_s)]\right|^2\right].
\end{aligned}$$

Since the $\bar{\mathbf{X}}^{\theta,j,N}_s$ are i.i.d. random variables with law f_s, the term inside the expectation is the squared error of a sample mean estimate. This is equal to the variance of the sample mean:

$$\mathbb{E}\left[\left|\mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, \bar{\mu}^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right|^2\right] \le \frac{C_i^2(1+C_e^2)}{N}\mathrm{Var}\left(\varphi_e(\bar{\mathbf{X}}^{\theta,j,N}_s)\right)$$

Since φ_e is Lipschitz and f_s has finite second moments (guaranteed by linear growth and Lipschitz continuous of $\mathbf{b}$), the variance is finite and bounded by some constant C_V. Therefore, the difference is bounded by:

$$\mathbb{E}\left[\left|\mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, \bar{\mu}^{\theta,N}_s) - \mathbf{b}_\theta(\bar{\mathbf{X}}^{\theta,i,N}_s, f^{\theta}_s)\right|^2\right] \le \frac{(C_iC_e)^2C_V}{N}.$$

Let $Y(t) = \mathbb{E}\left[\sup_{r\le t}\left|\mathbf{X}_r^{\theta,i,N} - \bar{\mathbf{X}}_r^{\theta,i,N}\right|^2\right]$ and using the Itô inequality, we have:

$$\begin{aligned}
Y(T) &\le \int_0^T Y(s)\mathrm{d}s + \int_0^T \mathbb{E}\left[\left|\mathbf{b}_\theta(\mathbf{X}_s^{\theta,i,N}, \mu_s^{\theta,N}) - \mathbf{b}_\theta(\bar{\mathbf{X}}_s^{\theta,i,N}, f_s^\theta)\right|^2\right]\mathrm{d}s \\
&\le \int_0^T Y(s)\mathrm{d}s + \int_0^T \left(2(2C_i^2 + 2(C_iC_e)^2)\mathbb{E}\left[|\mathbf{X}_s^{\theta,i,N} - \bar{\mathbf{X}}_s^{\theta,i,N}|^2\right] + 2\frac{(C_iC_e)^2C_V}{N}\right)\mathrm{d}s \\
&\le \frac{2TC_B}{N} + \left(1 + 4C_i^2 + 4(C_iC_e)^2\right)\int_0^T Y(s)\mathrm{d}s.
\end{aligned}$$

Thus, the following holds:

$$\mathbb{E}\left[\sup_{t\le T}\left|\mathbf{X}_t^{\theta,i,N} - \bar{\mathbf{X}}_t^{\theta,i,N}\right|^2\right] \le 2T\int_0^T C_i^2(1+C_e^2)\mathbb{E}\left[\sup_{t\le T}\left|\mathbf{X}_s^{\theta,i,N} - \bar{\mathbf{X}}_s^{\theta,i,N}\right|^2\right] + \frac{(C_iC_e)^2C_V}{N}\mathrm{d}s.$$

The conclusion follows by the Gronwall lemma. □

C Proof of Theorem 2.4

First, we provide an approximation result for Lipschitz functions by deep neural network. It will be the same arguments as in [57, 77], but we need to include the dependence of Lipschitz constant explicitly here. We keep the dependence on the Lipschitz constant L_h explicit because, in the proof of Theorem 2.4, the theorem is applied to an auxiliary finite-dimensional map whose Lipschitz constant depends on L_f and the covering complexity; this dependence must therefore be tracked in the final approximation-rate estimate.

Theorem C.1. *Let $d_1 \in \mathbb{N}$, $\gamma_1 > 0$, and $\Omega_h = [-\gamma_1, \gamma_1]^{d_1}$. Let $h : \Omega_h \to \mathbb{R}$ be L_h-Lipschitz, i.e.*

$$|h(x) - h(y)| \le L_h\|x - y\|_2, \qquad \forall x, y \in \Omega_h,$$

and bounded, i.e., $\|h\|_{L^\infty(\Omega_h)} \le \beta_h$. We have that there exist constants C dependent on γ_1 such that the following holds: for any $\epsilon > 0$, set $N = CL_h\sqrt{d_1}\epsilon^{-1}$. Let $\{\mathbf{c}_k\}_{k=1}^{N^{d_1}}$ be a uniform grid on Ω_h with spacing $\frac{2\gamma_1}{N-1}$ along each dimension. There exist neural networks architecture $\mathcal{F}_{NN}(d_1, 1, L, p, K, \kappa, R)$ and networks $\{\tilde{q}_k\}_{k=1}^{N^{d_1}}$ with $\tilde{q}_k \in \mathcal{F}_{NN}(d_1, 1, L, p, K, \kappa, R)$, for $k = 1, \cdots, N^{d_1}$, such that for any h satisfies the above assumption, we have

$$\left\| h - \sum_{k=1}^{N^{d_1}} h(\mathbf{c}_k)\tilde{q}_k \right\|_{L^\infty(\Omega_h)} \le \epsilon, \tag{15}$$

where

$$\begin{aligned}
L &= O(d_1^2\log(\epsilon^{-1}) + d_1^2L_h + d_1^2\log(d_1)) \\
p &= O(1) \\
K &= O(d_1^2\log(\epsilon^{-1}) + d_1^2L_h + d_1^2\log(d_1)) \\
\kappa &= O\left(d_1^{\frac{d_1}{2}+1}\epsilon^{-d_1-1}L_h^{d_1}\right) \\
R &= O(1).
\end{aligned}$$

Proof. We partition Ω_h into N^{d_1} subcubes for some N to be specific later. We are going to approximate h on each cube by a constnat function and then assemble them together to get an approximation of h on Ω_h. Denote the centers of the subcubes by $\{\mathbf{c}_k\}_{k=1}^{N^{d_1}}$ with $\mathbf{c}_k = [c_{k,1}, \cdots, c_{k,d_1}]^\top$. Let $\{\mathbf{c}_k\}_{k=1}^{N^{d_1}}$ be a unifom grid on Ω_h so that each $\mathbf{c}_k \in \left\{-\gamma_1, -\gamma_1 + \frac{2\gamma_1}{N-1}, \cdots, \gamma_1\right\}^{d_1}$ for each k. Define

$$\psi(a) = \begin{cases} 1, & |a| \le 1, \\ 0, & |a| > 2, \\ 2 - |a|, & 1 \le |a| \le 2, \end{cases} \tag{16}$$

with $a \in \mathbb{R}$, and

$$\phi_{\mathbf{c}_k}(\mathbf{x}) = \prod_{j=1}^{d_1}\psi\left(\frac{3(N-1)}{2\gamma_1}(x_j - c_{k,j})\right).$$

For any h, we construct a piecewise constant approximation of h as

$$\bar{h}(\mathbf{x}) = \sum_{k=1}^{N^{d_1}} h(\mathbf{c}_k)\phi_{\mathbf{c}_k}(\mathbf{x}).$$

By utilizing the partition of unity property given by $\sum_{k=1}^{N^{d_1}} \phi_{\mathbf{c}_k}(\mathbf{x}) = 1$, it follows that for any $\mathbf{x} \in \Omega_h$, we have

$$\begin{aligned}
|h(\mathbf{x}) - \bar{h}(\mathbf{x})| &= \left| \sum_{k=1}^{N^{d_1}} \phi_{\mathbf{c}_k}(\mathbf{x}) \left(h(\mathbf{x}) - h(\mathbf{c}_k)\right) \right| \\
&\leq \sum_{k=1}^{N^{d_1}} \phi_{\mathbf{c}_k}(\mathbf{x}) \left|h(\mathbf{x}) - h(\mathbf{c}_k)\right| \\
&\leq \sum_{k:\|\mathbf{c}_k - \mathbf{x}\|_\infty \leq \frac{2\gamma_1}{N-1}} \phi_{\mathbf{c}_k}(\mathbf{x}) \left|h(\mathbf{x}) - h(\mathbf{c}_k)\right| \\
&\leq \max_{k:\|\mathbf{c}_k - \mathbf{x}\|_\infty \leq \frac{2\gamma_1}{N-1}} \left|h(\mathbf{x}) - h(\mathbf{c}_k)\right| \left(\sum_{k:\|\mathbf{c}_k - \mathbf{x}\|_\infty \leq \frac{2\gamma_1}{N-1}} \phi_{\mathbf{c}_k}(\mathbf{x}) \right) \\
&\leq \max_{k:\|\mathbf{c}_k - \mathbf{x}\|_\infty \leq \frac{2\gamma_1}{N-1}} \left|h(\mathbf{x}) - h(\mathbf{c}_k)\right| \leq \frac{2\sqrt{d_1}\gamma_1 L_h}{N-1},
\end{aligned} \tag{17}$$

Setting $N = \lceil \frac{4\sqrt{d_1}\gamma_1 L_h}{\epsilon} \rceil + 1$ yields:

$$|h(\mathbf{x}) - \bar{h}(\mathbf{x})| \leq \frac{\epsilon}{2}, \quad \forall x \in \Omega_h.$$

Then we show that $\phi_{\mathbf{c}_k}$ can be approximated by a network with arbitrary accuracy. Note $\phi_{\mathbf{c}_k}$ is a product of d_1 functions, and each of them can be a piecewise linear function and can be realized by 4-layer ReLU networks.

Lemma C.2. *Given $M > 0$ and $\epsilon > 0$, there is a ReLU network $\tilde{f} : \mathbb{R}^d \to \mathbb{R}$ in $\mathcal{F}_{NN}(2, 1, L, p, K, \kappa, R)$ such that for any $|x| \leq M$ and $|y| \leq M$, we have*

$$\left| \tilde{f}(x, y) - f(x, y) \right| < \epsilon,$$

where $L = O(\log \epsilon^{-1})$, $p = 6$, $K = O(\log \epsilon^{-1})$, $\kappa = O(\epsilon^{-1})$, $R = M^2$. The constant hidden in O depends on M.

Let $\tilde{f}$ be the network defined in Lemma C.2 with accuracy δ. We approximate $\phi_{\mathbf{c}_k}$ by $\tilde{q}_k$ defined as:

$$\tilde{q}_k(\mathbf{x}) = \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_1 - c_{k,1})\right), \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_2 - c_{k,2})\right), \cdots\right)\right).$$

For each k, $\tilde{q}_k \in \mathcal{F}_{NN}(d_1, 1, L, p, K, \kappa, R)$ with

$$L = O(d_1 \log \delta^{-1}), p = O(1), K = O(d_1 \log \delta^{-1}), \kappa = O(\delta^{-1} + N), R = 1.$$

For any $\mathbf{x} \in \Omega_h$, we have that

$$\begin{aligned}
&|\tilde{q}_k(\mathbf{x}) - \phi_{\mathbf{c}_k}(\mathbf{x})| \\
\leq &\left| \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_1 - c_{k,1})\right), \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_2 - c_{k,2})\right), \cdots\right)\right) - \phi_{\mathbf{c}_k}(\mathbf{x}) \right| \\
\leq &\left| \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_1 - c_{k,1})\right), \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_2 - c_{k,2})\right), \cdots\right)\right) \right. \\
&\left. - \psi\left(\frac{2(N-1)}{2\gamma_1}(x_1 - c_{k,1})\right) \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_2 - c_{k,2})\right), \cdots\right) \right| \\
&+ \left| \psi\left(\frac{2(N-1)}{2\gamma_1}(x_1 - c_{k,1})\right) \tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_2 - c_{k,2})\right), \cdots\right) - \phi_{\mathbf{c}_k}(\mathbf{x}) \right| \\
\leq &\delta + \mathcal{E}_2
\end{aligned}$$

where

$$\begin{aligned}\mathcal{E}_2 &= \left|\psi\left(\frac{2(N-1)}{2\gamma_1}(x_1-c_{k,1})\right)\tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_2-c_{k,2})\right),\cdots\right)-\phi_{\mathbf{c}_k}(\mathbf{x})\right| \\ &= \left|\psi\left(\frac{2(N-1)}{2\gamma_1}(x_1-c_{k,1})\right)\right|\left|\tilde{f}\left(\psi\left(\frac{2(N-1)}{2\gamma_1}(x_2-c_{k,2})\right),\cdots\right)-\prod_{j=2}^{d_1}\psi\left(\frac{2(N-1)}{2\gamma_1}(x_j-c_{k,j})\right)\right|.\end{aligned}$$

Repeat this process to estimate $\mathcal{E}_2, \mathcal{E}_3, \cdots, \mathcal{E}_{d+1}$, where $\mathcal{E}_{d_1+1}=0$. This implies that $\|\phi_{\mathbf{c}_k}-\tilde{q}_k\|_{L^\infty(\Omega_h)} \le d_1\delta$. It follows that,

$$\left\|\sum_{k=1}^{N^{d_1}} h(\mathbf{c}_k)\tilde{q}_k-\bar{h}\right\|_{L^\infty(\Omega_h)} = \left\|\sum_{k=1}^{N^{d_1}} h(\mathbf{c}_k)\tilde{q}_k-\sum_{k=1}^{N^{d_1}} h(\mathbf{c}_k)\tilde{\phi}_{\mathbf{c}_k}\right\|_{L^\infty(\Omega_h)} \le \sum_{k=1}^{N^{d_1}}|h(\mathbf{c}_k)|\,\|\tilde{q}_k-\phi_{\mathbf{c}_k}\|_{L^\infty(\Omega_h)} \le d_1 N^{d_1}\beta_h\delta$$

and setting $\delta=\frac{\epsilon}{2d_1N^{d_1}\beta_h}$ yields:

$$\left\|h-\sum_{k=1}^{N^{d_1}} h(\mathbf{c}_k)\tilde{q}_k\right\|_{L^\infty(\Omega_h)} \le \epsilon$$

Therefore

$$\begin{aligned}L &= O\left(d_1\log\left(\frac{2d_1N^{d_1}\beta_h}{\epsilon}\right)\right) = O\left(d_1\log(\epsilon^{-1})+d_1^2\log(N)\right) = O(d_1^2\log(\epsilon^{-1})+d_1^2L_h+d_1^2\log(d_1)) \\ p &= O(1) \\ K &= O(d_1^2\log(\epsilon^{-1})+d_1^2L_h+d_1^2\log(d_1)) \\ \kappa &= O\left(\frac{d_1\left(\frac{\sqrt{d_1}L_h}{\epsilon}\right)^{d_1}}{\epsilon}\right) = O\left(d_1^{\frac{d_1}{2}+1}\epsilon^{-d_1-1}L_h^{d_1}\right) \\ R &= O(1).\end{aligned}$$

□

Proof of Theorem 2.4. By Lemma 2 in [77], there exist a cover $\{\mathcal{B}_\delta(\mathbf{c}_m)\}_{m=1}^{C_\Omega}$ of Ω by C_Ω Euclidean balls with $C_\Omega \le C\delta^{-d}$. There exists a partition of unity $\{\omega_m(\mathbf{x})\}_{m=1}^{C_\Omega}$ subordinate to the cover $\{\mathcal{B}_\delta(\mathbf{c}_m)\}_{m=1}^{C_\Omega}$ such that $0\le\omega_m\in C^\infty(\Omega)$ and $\sum_{m=1}^{C_\Omega}\omega(\mathbf{x})=1$ for all $\mathbf{x}\in\Omega$. For any $\mu\in U$, we define $\mathbf{u}=(\langle\omega_1,\mu\rangle,\cdots,\langle\omega_{C_\Omega},\mu\rangle)$. We note that:

$$\sum_{m=1}^{C_\Omega}\mathbf{u}_m=\int_\Omega\sum_{m=1}^{C_\Omega}\omega_m(\mathbf{x})\mu(\mathrm{d}\mathbf{x})=1.$$

and $\mathbf{u}_m\ge 0$ for all m. Therefore, we construct the approximate measure:

$$\mu_\omega=\sum_{m=1}^{C_\Omega}\mathbf{u}_m\delta_{\mathbf{c}_m}(\mathrm{d}\mathbf{x}),$$

The W_2 error estimation of the approximation is given by:

$$\begin{aligned}W_2^2(\mu,\mu_\omega) &= \inf_{\pi\in\Pi(\mu,\mu_\Omega)}\int_{\Omega\times\Omega}|\mathbf{x}-\mathbf{y}|^2\mathrm{d}\pi(\mathbf{x},\mathbf{y}) \\ &\le \int_{\Omega\times\Omega}|\mathbf{x}-\mathbf{y}|^2\mathrm{d}\pi^*(\mathbf{x},\mathbf{y}) \quad \left(\mathrm{d}\pi^*(\mathbf{x},\mathbf{y})=\left(\sum_{m=1}^{C_\Omega}\omega_m(\mathbf{x})\delta_{\mathbf{c}_m}(\mathrm{d}y)\right)\mathrm{d}\mu(\mathbf{x})\right) \\ &= \sum_{m=1}^{C_\Omega}\int_{\Omega\times\Omega}|\mathbf{x}-\mathbf{c}_m|^2\omega_m(\mathbf{x})\mathrm{d}\mu(\mathbf{x}) \\ &= \sum_{m:\|\mathbf{x}-\mathbf{c}_m\|_2\le\delta}\int_{\Omega\times\Omega}|\mathbf{x}-\mathbf{c}_m|^2\omega_m(\mathbf{x})\mathrm{d}\mu(\mathbf{x}) \\ &\le \sum_{m:\|\mathbf{x}-\mathbf{c}_m\|_2\le\delta}\delta^2\mathbf{u}_m\le\delta^2\end{aligned}$$

Setting $\delta = \frac{\epsilon}{2L_f}$ and using the Lipschitz property of f, we have:

$$\begin{aligned}|f(\mathbf{x},\mu) - f(\mathbf{x},\mu_\omega)| \leq & L_f W_1(\mu,\mu_\omega)\\ \leq & L_f W_2(\mu,\mu_\omega)\\ \leq & \frac{\epsilon}{2}.\end{aligned}$$

We also have that $C_\Omega \leq C\epsilon^{-\epsilon}$. We next define a function $g : \mathbb{R}^d \times \mathbb{R}^{C_\Omega} \to \mathbb{R}$ such that $g(\mathbf{x},\mathbf{u}) = f(\mathbf{x},\mu_\omega)$. We claim that g is Lipschitz in the following sense: for any $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$, and for any $\mu,\nu \in \mathcal{P}_2(\Omega)$, define μ_ω and ν_ω as before with coefficients $\mathbf{u}$ and $\mathbf{v}$ respectively. We have:

$$\begin{aligned}|g(\mathbf{x},\mathbf{u}) - g(\mathbf{y},\mathbf{v})| = & |f(\mathbf{x},\mu_\omega) - f(\mathbf{y},\nu_\omega)|\\ \leq & L_f\left(|\mathbf{x}-\mathbf{y}|_2 + W_1(\mu_\omega,\nu_\omega)\right)\end{aligned}$$

Notice that

$$\begin{aligned}W_1(\mu_\omega,\nu_\omega) &= \sup_{\mathrm{Lip}(t)\leq 1}\left\{\int_\Omega t(x)d\mu_\omega(x) - \int_\Omega t(x)d\nu_\omega(x)\right\}\\ &= \sup_{\mathrm{Lip}(t)\leq 1}\left\{\int_\Omega t(\mathbf{x})\sum_{m=1}^{C_\Omega}(\mathbf{u}_m - \mathbf{v}_m)\delta_{\mathbf{c}_m}(\mathrm{d}\mathbf{x})\right\}\\ &= \sup_{\mathrm{Lip}(t)\leq 1}\left\{\sum_{m=1}^{C_\Omega} t(\mathbf{c}_m)(\mathbf{u}_m - \mathbf{v}_m)\right\}\end{aligned}$$

Leting $\mathbf{w}_m^+ = \max\{\mathbf{u}_m - \mathbf{v}_m, 0\}$ and $\mathbf{w}_m^- = -\min\{\mathbf{u}_m - \mathbf{v}_m, 0\}$ yields:

$$W_1(\mu_\omega,\nu_\omega) = \sup_{\mathrm{Lip}(t)\leq 1}\left\{\sum_{m=1}^{C_\Omega} t(\mathbf{c}_m)\mathbf{w}_m^+ - t(\mathbf{c}_m)\mathbf{w}_m^-\right\}.$$

Next, define $M = \sum_{m=1}^{C_\Omega}\mathbf{w}_m^+ = \sum_{m=1}^{C_\Omega}\mathbf{w}_m^- = \frac{1}{2}|\mathbf{u}-\mathbf{v}|_1$ and thus the following bound holds:

$$\begin{aligned}W_1(\mu_\omega,\nu_\omega) \leq & \sup_{\mathrm{Lip}(t)\leq 1}\left\{\sum_{m=1}^{C_\Omega} t_{\max}\mathbf{w}_m^+ - t_{\min}\mathbf{w}_m^-\right\}\\ \leq & M(t_{\max} - t_{\min}) \leq 2MR\sqrt{d} = 2R\sqrt{d}|\mathbf{u}-\mathbf{v}|_1 \leq 2R\sqrt{C_\Omega d}|\mathbf{u}-\mathbf{v}|_2.\end{aligned}$$

Therefore, we have:

$$\begin{aligned}|g(\mathbf{x},\mathbf{u}) - g(\mathbf{y},\mathbf{v})| \leq & L_f\left(|\mathbf{x}-\mathbf{y}|_2 + 2R\sqrt{C_\Omega d}\|\mathbf{u}-\mathbf{v}\|_2\right)\\ \leq & 2R\sqrt{C_\Omega d}L_f\sqrt{|\mathbf{x}-\mathbf{y}|_2^2 + |\mathbf{u}-\mathbf{v}|_2^2}\end{aligned}$$

Therefore, by Theorem C.1, for $\epsilon > 0$, we set $H = N^{d_1}$, where $N = \lfloor\frac{4\sqrt{d_1}L_U\gamma_1}{\epsilon}\rfloor + 1$ and $L_U = 2R\sqrt{C_\Omega d}L_f, d_1 = d + C_\Omega, \gamma_1 = \max\{1, R\}$. Thus:

$$H \lesssim \left(\frac{C\,\gamma_1\sqrt{d+C_\Omega}\,R\sqrt{C_\Omega d}\,L_f}{\epsilon}\right)^{d+C_\Omega} \leq C(C_\Omega + d)^{\frac{C_\Omega+d}{2}}(C_\Omega d)^{\frac{C_\Omega+d}{2}}\epsilon^{-C_\Omega-d}.$$

where C depending on R, L_f, then there exist H neural networks $\{q_k\}_{k=1}^H \in \mathcal{F}_{NN}(d+C_\Omega, 1, L, p, K, \kappa, R)$ with

$$\begin{aligned}L &= O((d+C_\Omega)^2\log(\epsilon^{-1}) + (d+C_\Omega)^2\sqrt{C_\Omega d}L_f + (d+C_\Omega)^2\log(d+C_\Omega)))\\ p &= O(1)\\ K &= O((d+C_\Omega)^2\log(\epsilon^{-1}) + (d+C_\Omega)^2\sqrt{C_\Omega d}L_f + (d+C_\Omega)^2\log(d+C_\Omega))\\ \kappa &= O\left((d+C_\Omega)^{\frac{d+C_\Omega}{2}+1}\epsilon^{-d-C_\Omega-1}L_h^{d+C_\Omega}\right)\\ R &= O(1),\end{aligned}$$

such that

$$\sup_{\mu\in U,\mathbf{x}\in\Omega}\left|g(\mathbf{x},\mathbf{u}) - \sum_k^H a_k q_k(\mathbf{x},\mathbf{u})\right| \leq \frac{\epsilon}{2},$$

where a_k are constant dependent on f. We have for any $\mu \in \mathcal{P}_2(\Omega)$

$$\begin{aligned}\sup_{\mu\in U,\mathbf{x}\in\Omega}\left|f(\mathbf{x},\mu)-\sum_k^H a_k q_k(\mathbf{x},\mathbf{u})\right| \leq& \sup_{\mu\in U,\mathbf{x}\in\Omega}|f(\mathbf{x},\mu)-g(\mathbf{x},\mathbf{u})|+\sup_{\mu\in U,\mathbf{x}\in\Omega}\left|g(\mathbf{x},\mathbf{u})-\sum_k^H a_k q_k(\mathbf{x},\mathbf{u})\right| \\ =&\sup_{\mu}|f(\mathbf{x},\mu)-f(\mathbf{x},\mu_\omega)|+\sup_{\mu}\left|g(\mathbf{x},\mathbf{u})-\sum_k^H a_k q_k(\mathbf{x},\mathbf{u})\right| \\ \leq&\epsilon\end{aligned}$$

□

D Proof of Theorem 2.7

Proof. By Assumption 1 and Remark 2, there exists an L_G-Lipschitz function $G : \mathbb{R}^d \times \mathbb{R}^r \to \mathbb{R}^d$ such that $\mathbf{b}^\star(\mathbf{X},\mu) = G(\mathbf{X},\langle \mathbf{g},\mu\rangle)$, where $\mathbf{g} = (g_1,\ldots,g_r)$ is the vector of $L_{\mathbf{g}}$-Lipschitz feature functions. Since $\mathbf{g}$ is Lipschitz and the domain $[0,1]^d$ is compact, each g_j is bounded. Let:

$$M_{\mathbf{g}} := \sup_{\mathbf{Y}\in[0,1]^d}\|\mathbf{g}(\mathbf{Y})\|_\infty < \infty,$$

then, for any probability measure μ supported on $[0,1]^d$:

$$\|\langle \mathbf{g},\mu\rangle\|_\infty \leq \int \|\mathbf{g}(\mathbf{Y})\|_\infty \,\mu(d\mathbf{Y}) \leq M_{\mathbf{g}}.$$

Our goal is to construct an MVNN of the form: $\mathbf{b}_\theta(\mathbf{X},\mu) = \varphi_{\text{int}}\big(\mathbf{X},\langle\varphi_{\text{e}},\mu\rangle\big)$ such that:

$$\sup_{\mathbf{X}\in[0,1]^d,\,\mu\in U}\big\|\mathbf{b}^\star(\mathbf{X},\mu)-\mathbf{b}_\theta(\mathbf{X},\mu)\big\|_{\mathbb{R}^d} \leq \epsilon.$$

For brevity, denote $\mathbf{z}_{\mathbf{g}}(\mu) := \langle \mathbf{g},\mu\rangle \in \mathbb{R}^r, \mathbf{z}_{\text{e}}(\mu) := \langle \varphi_{\text{e}}^{(1:r)},\mu\rangle \in \mathbb{R}^r$, where $\varphi_{\text{e}}^{(1:r)}$ denotes the first r coordinates of φ_{e}. We decompose the total error as:

$$\begin{aligned}&\sup_{\mathbf{X},\mu}\big\|G(\mathbf{X},\langle\mathbf{g},\mu\rangle)-\varphi_{\text{i}}(\mathbf{X},\langle\varphi_{\text{e}}^{(1:r)},\mu\rangle)\big\|_{\mathbb{R}^d} \\ &\leq \sup_{\mathbf{X},\mu}\big\|G(\mathbf{X},\langle\mathbf{g},\mu\rangle)-G(\mathbf{X},\langle\varphi_{\text{e}}^{(1:r)},\mu\rangle)\big\|_{\mathbb{R}^d} \\ &\quad+\sup_{\mathbf{X},\mu}\big\|G(\mathbf{X},\langle\varphi_{\text{e}}^{(1:r)},\mu\rangle)-\varphi_{\text{i}}(\mathbf{X},\langle\varphi_{\text{e}}^{(1:r)},\mu\rangle)\big\|_{\mathbb{R}^d}\end{aligned}$$

For the first term, since G is L_G-Lipschitz in its second argument, we have:

$$\begin{aligned}&\sup_{\mathbf{X},\mu}\big\|G(\mathbf{X},\langle\mathbf{g},\mu\rangle)-G(\mathbf{X},\langle\varphi_{\text{e}}^{(1:r)},\mu\rangle)\big\|_{\mathbb{R}^d} \\ \leq& L_G\sup_{\mu\in U}\big\|\langle\mathbf{g},\mu\rangle-\langle\varphi_{\text{e}}^{(1:r)},\mu\rangle\big\|_{\mathbb{R}^r} \\ =& L_G\left\|\int\big(\mathbf{g}(\mathbf{Y})-\varphi_{\text{e}}^{(1:r)}(\mathbf{Y})\big)\,\mu(d\mathbf{Y})\right\|_{\mathbb{R}^r} \\ \leq& L_G\int\big\|\mathbf{g}(\mathbf{Y})-\varphi_{\text{e}}^{(1:r)}(\mathbf{Y})\big\|_{\mathbb{R}^r}\,\mu(d\mathbf{Y}) \\ \leq& L_G\sup_{\mathbf{Y}\in[0,1]^d}\big\|\mathbf{g}(\mathbf{Y})-\varphi_{\text{e}}^{(1:r)}(\mathbf{Y})\big\|_{\mathbb{R}^r}.\end{aligned}$$

We will enforce:

$$\sup_{\mathbf{Y}\in[0,1]^d}\big\|\mathbf{g}(\mathbf{Y})-\varphi_{\text{e}}^{(1:r)}(\mathbf{Y})\big\|_{\mathbb{R}^r} \leq \epsilon_{\text{e}} := \frac{\epsilon}{2L_G},$$

so that

$$\sup_{\mathbf{X},\mu}\big\|G(\mathbf{X},\langle\mathbf{g},\mu\rangle)-G(\mathbf{X},\langle\varphi_{\text{e}}^{(1:r)},\mu\rangle)\big\|_{\mathbb{R}^d} \leq \epsilon/2. \tag{18}$$

By Theorem 1 in [87], for any $\epsilon_{\mathrm{e}} > 0$, there exists a deep ReLU network

$$\varphi_{\mathrm{e}} : \mathbb{R}^d \to \mathbb{R}^k, \quad k \geq r,$$

has the depth ast most $O(\log(\epsilon_{\mathrm{e}}^{-1})) = O(\log(\epsilon^{-1}))$ and width at most $O\big(r \cdot \epsilon_{\mathrm{e}}^{-d}\big) = O\big(r \cdot \epsilon^{-d}\big)$ and the total number of parameters at most $O(r \cdot \epsilon^{-d} \log(\epsilon^{-1}))$, such that where the constant depends on L_G, d. For the second term, consider the compact domain

$$\mathcal{D} := [0,1]^d \times [-R, R]^r \subset \mathbb{R}^{d+r}.$$

where $R := M_{\mathbf{g}} + \epsilon_{\mathrm{e}}$ with both $\langle \mathbf{g}, \mu \rangle$ and $\langle \varphi_{\mathrm{e}}^{(1:r)}, \mu \rangle$ lie in the box $[-R, R]^r$ for all $\mu \in U$. Again by Theorem 1 in [87], for any $\epsilon_{\mathrm{i}} > 0$ there exists a deep ReLU network:

$$\varphi_{\mathrm{i}} : \mathbb{R}^{d+r} \to \mathbb{R}^d$$

with depth L_{int} and width W_{int} at most:

$$L_{\mathrm{int}} = O(\log(\epsilon^{-1})), \qquad W_{\mathrm{int}} = O\big(d \cdot \epsilon^{-(d+r)}\big).$$

such that:

$$\sup_{(\mathbf{X},\mathbf{z}) \in \mathcal{D}} \big\| G(\mathbf{X}, \mathbf{z}) - \varphi_{\mathrm{i}}(\mathbf{X}, \mathbf{z}) \big\|_{\mathbb{R}^d} \leq \epsilon/2. \tag{19}$$

Combining the two estimates in Equation (18) and (19) gives

$$\sup_{X \in [0,1]^d,\ \mu \in U} \|\mathbf{b}^\star(\mathbf{X}, \mu) - \mathbf{b}_\theta(\mathbf{X}, \mu)\|_{\mathbb{R}^d} \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

This completes the proof. □