Mean-Field Games and Applications in Finance

New Challenges in Financial and Energy Markets — Math, Data & AI

Stefanos Theodorakopoulos

31/10/2025

Technische Universität Berlin

Table of contents

Introduction to stochastic differential N-Player Games

Definition: N-Player Game

An **N-player game**, see [1], consists of N interacting agents, each controlling their own state process and seeking to minimize an individual cost functional.

State dynamics:

$$dX_t^i = b(t, X_t^i, \mu_t^N, \alpha_t^i)\, dt + \sigma(t, X_t^i, \mu_t^N, \alpha_t^i)\, dW_t^i,$$

where:

- $X_t^i \in \mathbb{R}^d$: state of player i,
- $\alpha_t^i \in A$: control (action),
- W_t^i: independent Brownian motion,
- $\mu_t^N = \dfrac{1}{N} \sum_{j=1}^{N} \delta_{X_t^j}$: empirical distribution of states.

Objective Function

Each player i minimizes the expected cost:

$$J_i^N(\alpha^1, \ldots, \alpha^N) = \mathbb{E}\left[\int_0^T f(t, X_t^i, \mu_t^N, \alpha_t^i)\, dt + g(X_T^i, \mu_T^N)\right],$$

where:

- f: running cost,
- g: terminal cost.

Objective Function

Each player i minimizes the expected cost:

$$J_i^N(\alpha^1, \ldots, \alpha^N) = \mathbb{E}\left[\int_0^T f(t, X_t^i, \mu_t^N, \alpha_t^i)\, dt + g(X_T^i, \mu_T^N)\right],$$

where:

- f: running cost,
- g: terminal cost.

Goal
Each player chooses α^i to minimize J_i^N given the others' strategies.

Nash Equilibrium

A strategy profile

$$(\alpha^{1,*}, \ldots, \alpha^{N,*})$$

is a **Nash equilibrium** if, for all i and any alternative control α^i,

$$J_i^N(\alpha^{1,*}, \ldots, \alpha^{N,*}) \leq J_i^N(\alpha^{1,*}, \ldots, \alpha^i, \ldots, \alpha^{N,*}).$$

A strategy profile

$$(\alpha^{1,*}, \ldots, \alpha^{N,*})$$

is a **Nash equilibrium** if, for all i and any alternative control α^i,

$$J_i^N(\alpha^{1,*}, \ldots, \alpha^{N,*}) \leq J_i^N(\alpha^{1,*}, \ldots, \alpha^i, \ldots, \alpha^{N,*}).$$

Interpretation
No player can unilaterally change their control to achieve a lower expected cost.

Best Response function

Let $A = A_1 \times \cdots \times A_N$ be the set of strategy profiles, and for a profile $\alpha = (\alpha^1, \ldots, \alpha^N) \in A$, we write $\alpha^{-i} := (\alpha^1, \ldots, \alpha^{i-1}, \alpha^{i+1}, \ldots, \alpha^N)$ and $\alpha = (\alpha^i, \alpha^{-i})$.

Definition 1.1. The *best responses* of agent i to the actions α^{-i} of the other agents is the subset

$$\mathrm{B}_i(\alpha^{-i}) := \arg\min_{\gamma \in A_i} J_i\big(\gamma, \alpha^{-i}\big).$$

Moreover, *if we assume that there exists a unique minimum for every* B_i *each time*, then the *best response function* (for all agents simultaneously) is the map

$$\mathrm{B} : A \to A, \quad \mathrm{B}(\alpha) = \Big(\mathrm{B}_1(\alpha^{-1}), \cdots, \mathrm{B}_N(\alpha^{-N})\Big).$$

Best Response function

Hence, in this setting, a strategy profile $\alpha^\star \in A$ is a Nash equilibrium for the N-player game if and only if

$$\mathrm{B}(\alpha^\star) = \alpha^\star,$$

i.e. each $\alpha^{\star i}$ is a best response to

$$\alpha^{\star -i}.$$

In other words, $\alpha^\star$ is a fixed point for the best response function B.

Connection to Mean Field Games

As $N \to \infty$, the empirical measure

$$\mu_t^N = \frac{1}{N} \sum_{j=1}^{N} \delta_{X_t^j}$$

converges to a deterministic flow of measures m_t.

Connection to Mean Field Games

As $N \to \infty$, the empirical measure

$$\mu_t^N = \frac{1}{N} \sum_{j=1}^{N} \delta_{X_t^j}$$

converges to a deterministic flow of measures m_t.

Mean Field Limit
In the limit, each player interacts only with the **mean field** m_t, not with individual players.

- The limiting control problem defines a **Mean Field Game (MFG)**.
- MFGs represent the infinite-population limit of N-player games.

Mean Field Limit: Symmetry and Law of Large Numbers

Symmetry: All players have identical data and

$$\alpha_t^i = \phi(t, X_t^i), \quad \forall i \in \{1, ..., N\}, \; t \in [0, T]$$

$\Rightarrow$ all the players in the game are statistically identical.

Representative agent problem:

$$\begin{cases} dX_t = b(t, X_t, m_t, \alpha_t)\, dt + \sigma(t, X_t, m_t, \alpha_t)\, dW_t, \\ \text{Minimize } J(\alpha; m) = \mathbb{E}\Big[\int_0^T f(t, X_t, m_t, \alpha_t)\, dt + g(X_T, m_T)\Big]. \end{cases}$$

Each player optimizes given $m = (m_t)$; equilibrium requires self-consistency.

Mean Field Game Equilibrium

Mean-field best response:

$$\widehat{\alpha}[m] = \arg\min_{\alpha \in \mathcal{A}} J(\alpha; m),$$

where $J(\alpha; m)$ is the representative agent's cost given $m = (m_t)$.

Self-consistency (mean-field equilibrium):

$$m_t = (X_t^{\widehat{\alpha}[m]}), \quad \forall t \in [0, T].$$

Connection:

Best-response fixed point (Nash) $\implies$ Self-consistent fixed point (MFG).

Applications in Finance

Mean Field Game of Systemic Risk

Economic Context

- Population of N banks interacting through **interbank borrowing/lending**.
- Each bank manages its **liquidity reserve** X_t^i over $[0, T]$.
- No common noise: only idiosyncratic shocks W_t^i.

Dynamics of Bank i

$$dX_t^i = a(\bar{X}_t - X_t^i)\,dt + \alpha_t^i\,dt + \sigma\,dW_t^i, \quad \bar{X}_t = \frac{1}{N}\sum_{j=1}^{N} X_t^j.$$

- $a > 0$: rate of mean reversion to the interbank average $\bar{X}_t$.
- α_t^i: control = borrowing/lending rate (decision variable).
- σ: volatility of individual liquidity shocks.

Mean Field Game of Systemic Risk

Cost Functional

$$J^i(\alpha^i; \bar{X}_t) = \mathbb{E}\left[\int_0^T \left(\tfrac{1}{2}(\alpha_t^i)^2 + \tfrac{q}{2}(X_t^i - \bar{X}_t)^2\right) dt + \tfrac{c}{2}(X_T^i - \bar{X}_T)^2\right].$$

- $(\alpha_t^i)^2$: cost of active borrowing/lending.
- $(X_t^i - \bar{X}_t)^2$: deviation from system average.
- $(X_T^i - \bar{X}_T)^2$: terminal imbalance.

Mean Field Limit and Equilibrium System

Mean Field Limit ($N \to \infty$)

$$dX_t = a(\bar{X}_t - X_t)\,dt + \alpha_t\,dt + \sigma\,dW_t, \quad \bar{X}_t = \mathbb{E}[X_t].$$

Objective:

$$J(\alpha;\bar{X}) = \mathbb{E}\left[\int_0^T \left(\tfrac{1}{2}\alpha_t^2 + \tfrac{q}{2}(X_t - \bar{X}_t)^2\right)dt + \tfrac{c}{2}(X_T - \bar{X}_T)^2\right].$$

Consistency (Equilibrium Conditions)

$$\bar{X}_t = \mathbb{E}[X_t], \qquad \bar{Y}_t = \mathbb{E}[Y_t].$$

Interpretation: Each bank optimally adjusts liquidity relative to system average. Collectively, equilibrium captures the feedback between individual actions and systemic stability.

Interpretation of the Parameter q

Mathematical Role

- Appears in the running cost:

$$\frac{q}{2}(X_t - \bar{X}_t)^2$$

  penalizing deviations from the mean liquidity.
- In the adjoint equation:

$$dY_t = -\big((a+q)Y_t - a\bar{Y}_t + q(X_t - \bar{X}_t)\big)dt + Z_t dW_t.$$

- Controls the coupling strength between agents and affects stability and variance of X_t.

Economic Interpretation

- q = **alignment penalty**: strength of pressure to follow the system's average liquidity.
- High q: strong interbank coordination, lower dispersion, higher contagion risk.
- Low q: weaker coupling, more heterogeneity, lower systemic dependence.
- Models peer effects, market discipline, or regulatory incentives to remain close to the mean.

Interpretation of the Control α_t

Economic Interpretation

- α_t: **Active liquidity adjustment** or net **borrowing/lending rate**.
- $\alpha_t > 0$: bank borrows $\Rightarrow$ increases reserves.
- $\alpha_t < 0$: bank lends $\Rightarrow$ reduces reserves.
- Cost $\frac{1}{2}\alpha_t^2$: funding or transaction friction.
- Balances:
  (i) aligning with market average ($X_t \approx \bar{X}_t$),
  (ii) limiting costly liquidity adjustments.

Summary: α_t governs the bank's optimal liquidity response to systemic conditions — balancing stability and cost.

Model Setup
Banks manage log-reserves under idiosyncratic and common shocks:

$$dX_t^{i,N} = [a(\bar{X}_t - X_t^{i,N}) + \alpha_t^i]\,dt + \sigma\sqrt{1-\rho^2}\,dW_t^i + \sigma\rho\,dW_t^0, \quad \bar{X}_t = \frac{1}{N}\sum_{j=1}^{N} X_t^{j,N}.$$

Cost:

$$J^i = \mathbb{E}\left[\int_0^T \left(\tfrac{1}{2}(\alpha_t^i)^2 - q\,\alpha_t^i(X_t - \bar{X}_t) + \tfrac{c}{2}(X_t - \bar{X}_t)^2\right)dt + \tfrac{1}{2}(X_T - \bar{X}_T)^2\right].$$

Mean Field Limit
As $N \to \infty$:

$$dX_t = [a(\bar{X}_t - X_t) + \alpha_t]\,dt + \sigma\sqrt{1-\rho^2}\,dW_t + \sigma\rho\,dW_t^0,$$

with random mean $\bar{X}_t = \mathbb{E}[X_t | W^0]$ (conditional law).

Interpretation
Captures contagion and liquidity effects with systemic shocks.
Common noise $\Rightarrow$ stochastic mean field (random equilibrium measure).

Model Setup
A continuum of producers extracts a commodity. Market price depends on total production $\bar{\alpha}_t$.

$$dX_t^i = \alpha_t^i \, dt,$$

$$P_t^N = P_0 - \gamma \, \tfrac{1}{N} \sum_{j=1}^{N} \alpha_t^j,$$

$$J^i = \mathbb{E}\left[\int_0^T \left(C(\alpha_t^i) - \alpha_t^i P_t^N \right) dt + G(X_T^i, \bar{\alpha}_T^N) \right].$$

Mean Field Limit

$$dX_t = \alpha_t \, dt, \quad P_t = P_0 - \gamma \, \bar{\alpha}_t, \quad J(\alpha; \bar{\alpha}) = \mathbb{E}\left[\int_0^T \big(C(\alpha_t) - \alpha_t P_t \big) \, dt + G(X_T, \bar{\alpha}_T) \right],$$

with $\bar{\alpha}_t = \mathbb{E}[\alpha_t]$.

Interpretation

Firms' extraction rates jointly determine price. Mean control $\bar{\alpha}_t$ creates price externalities → **Extended MFG**. Applications: oil production, renewables, emission markets.

Energy Transition with Common Policy Noise

Model Setup

Firms control emissions α_t^i facing random policy shocks:

$$dX_t^i = \alpha_t^i \, dt + \sigma_1 \, dW_t^i + \sigma_0 \, dW_t^0, \quad X_0^i = x_0.$$

Common noise models regulatory or macroeconomic shocks.
Coupling through $\bar{X}_t$ represents aggregate emissions effects.

Cost functional:

$$J^i = \mathbb{E}\left[\int_0^T \big(c_1(\alpha_t^i) + c_2(X_t^i, \bar{X}_t)\big) dt + \Phi(X_T^i, \bar{X}_T)\right], \quad \bar{X}_t = \mathbb{E}[X_t \mid W^0].$$

Mean Field Equilibrium

Given $\bar{X}_t$, each firm's best response $\alpha^*(\bar{X})$ yields

$$\bar{X}_t = \mathbb{E}[X_t^{\alpha^*(\bar{X})} \mid W^0].$$

Price Impact under Common Market Shocks

Model Setup

Traders control execution rates α_t^i affecting the market price:

$$dX_t^i = \alpha_t^i\,dt + \sigma\,dW_t^i,$$
$$dS_t^N = \kappa\,\tfrac{1}{N}\sum_{j=1}^{N}\alpha_t^j\,dt + \sigma_0\,dW_t^0.$$

Market-wide noise W^0 drives price shocks. Each trader adapts to the common market factor.

Objective:

$$J^i = \mathbb{E}\left[\int_0^T e^{-rt}\Big(\tfrac{1}{2}(\alpha_t^i)^2 + \lambda\,\alpha_t^i S_t^N\Big)dt + \tfrac{q}{2}(X_T^i)^2\right].$$

Mean Field Limit

$$dS_t = \kappa\,\mathbb{E}[\alpha_t \mid W^0]\,dt + \sigma_0\,dW_t^0, \quad \bar{\alpha}_t = \mathbb{E}[\alpha_t \mid W^0].$$

The General Mean Field approach

Neural Networks, Hidden Layers, and Neurons

A **neural network** is a function

$$f_\Theta : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}},$$

built as a composition of layers:

$$f_\Theta(x) = W_L\sigma(W_{L-1}\sigma(\dots\sigma(W_1x + b_1)\dots) + b_{L-1}) + b_L.$$

It approximates complex nonlinear mappings between inputs and outputs.

A **hidden layer** transforms its input through multiple neurons:

$$h = \sigma(Wx + b),$$

where W is the weight matrix, b is the bias, and σ is an activation function. The outputs h are internal feature representations, not directly observed.

Neuron

A **neuron** is a single computational unit:

$$h_i = \sigma(w_i \cdot x + b_i),$$

which takes an input x, applies a linear transformation, adds a bias, and passes it through a nonlinearity.

Neurons form hidden layers, and layers compose the neural network.

Universal Approximation Theorem

Theorem (Universal Approximation)
Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a continuous, non-polynomial activation function (e.g. sigmoid, ReLU, tanh). Then, for any continuous function $f \in C(K)$, defined on a compact set $K \subset \mathbb{R}^d$, and for every $\varepsilon > 0$, there exists a neural network of the form

$$f_N(x) = \sum_{i=1}^{N} a_i \, \sigma(w_i \cdot x + b_i),$$

such that

$$\sup_{x \in K} |f(x) - f_N(x)| < \varepsilon.$$

Interpretation
A one-hidden-layer neural network with sufficiently many neurons can approximate any continuous function on a compact domain, to arbitrary accuracy.

Depth adds efficiency, but width alone ensures universality.

Shallow vs. Deep Neural Networks: Expressive Power and Efficiency

Key Idea
All sufficiently wide neural networks can approximate any continuous function (**universal approximation**), but adding **depth** dramatically improves **efficiency**.

Shallow Networks (1 hidden layer)

- Can approximate any $f \in C(K)$, but may require **exponentially many neurons**.
- Represent functions as wide, single-step mappings.
- Difficult to capture hierarchical or compositional structures.

Deep Networks (many layers)

- Achieve similar accuracy with **polynomially many neurons**.
- Build complexity through successive nonlinear compositions.
- Efficiently reuse features learned at earlier layers.

Model: One-Hidden-Layer Neural Network

We shall follow [5].

Network Structure

$$g_\theta^N(x) = \frac{1}{N}\sum_{i=1}^{N} c_i\,\sigma(w_i\cdot x), \quad \theta = (c_1,\dots,c_N,w_1,\dots,w_N).$$

$$\mathcal{L}_N(\theta) = \tfrac{1}{2}\,\mathbb{E}_{(X,Y)}\big[(Y - g_\theta^N(X))^2\big].$$

Model: One-Hidden-Layer Neural Network

We shall follow [5].

Network Structure

$$g_\theta^N(x) = \frac{1}{N}\sum_{i=1}^{N} c_i\,\sigma(w_i\cdot x), \quad \theta = (c_1, \ldots, c_N, w_1, \ldots, w_N).$$

$$\mathcal{L}_N(\theta) = \tfrac{1}{2}\,\mathbb{E}_{(X,Y)}\left[(Y - g_\theta^N(X))^2\right].$$

- $c_i \in \mathbb{R}$: output weights,
- $w_i \in \mathbb{R}^d$: hidden layer weights,
- σ: activation function (ReLU, sigmoid, etc.),
- The factor $1/N$ ensures bounded outputs as $N \to \infty$.

Training a Neural Network

Goal
Adjust the parameters θ (weights and biases) so that

$$f_\theta(x) \approx y$$

for given training data (x, y), where $y =$ output of x.

Optimization Problem
Minimize the expected loss:

$$\mathcal{L}(\theta) = \frac{1}{2}\,\mathbb{E}_{(x,y)}[(f_\theta(x) - y)^2].$$

Objective:

$$\theta^* = \arg\min_\theta \mathcal{L}(\theta).$$

Stochastic Gradient Descent (SGD)

Goal: Minimize Expected Loss
Given a parameter vector $\theta \in \mathbb{R}^d$, minimize

$$\mathcal{L}(\theta) = \mathbb{E}_\xi\left[\ell(\theta;\xi)\right],$$

where $\ell(\theta;\xi)$ is the loss on a random data sample ξ.

Discrete-Time SGD Iteration
At each iteration k:

$$\boxed{\theta_{k+1} = \theta_k - \alpha_k \nabla_\theta \ell(\theta_k;\xi_k)}$$

- α_k : learning rate.
- ξ_k : randomly drawn sample or mini-batch.
- $\nabla_\theta \ell(\theta_k;\xi_k)$: stochastic gradient estimate of $\nabla L(\theta_k)$.

In our case $\xi = (x, y)$.

Step 0: Initialization in Stochastic Gradient Descent

Definition
Before training, initialize all neuron parameters:

$$\theta_i^0 = (c_i^0, w_i^0) \sim \mu_0, \quad i = 1, \ldots, N.$$

Step 0: Initialization in Stochastic Gradient Descent

Definition
Before training, initialize all neuron parameters:

$$\theta_i^0 = (c_i^0, w_i^0) \sim \mu_0, \quad i = 1, \dots, N.$$

- μ_0: initial distribution of parameters (e.g. Gaussian, uniform).
- Parameters are usually drawn **i.i.d.**, ensuring

$$\nu_0^N = \frac{1}{N} \sum_i \delta_{\theta_i^0} \Rightarrow \mu_0.$$

- Sets the **initial condition** for the mean-field PDE:

$$\partial_t \rho_t + \nabla_\theta \cdot (\rho_t v_t) = 0, \quad \rho_{t=0} = \mu_0.$$

Step 0 defines the initial particle cloud in parameter space — the seed for the mean-field dynamics.

Stochastic Gradient Descent (SGD) Updates

Discrete SGD (for sample (x_k, y_k))

$$c_i^{k+1} = c_i^k + \alpha_N (y_k - g_{\theta^k}^N(x_k))\, \sigma(w_i^k \cdot x_k),$$
$$w_i^{k+1} = w_i^k + \alpha_N (y_k - g_{\theta^k}^N(x_k))\, c_i^k\, \sigma'(w_i^k \cdot x_k)\, x_k.$$

Stochastic Gradient Descent (SGD) Updates

Discrete SGD (for sample (x_k, y_k))

$$c_i^{k+1} = c_i^k + \alpha_N (y_k - g_{\theta^k}^N(x_k))\, \sigma(w_i^k \cdot x_k),$$
$$w_i^{k+1} = w_i^k + \alpha_N (y_k - g_{\theta^k}^N(x_k))\, c_i^k\, \sigma'(w_i^k \cdot x_k)\, x_k.$$

- α_N: learning rate (scaled with N for limit).
- Each neuron adjusts parameters based on prediction error.
- The evolution of $\{(c_i, w_i)\}$ forms a particle system.

Empirical Measure Representation

Empirical Measure of Neuron Parameters

$$\nu_k^N(dc, dw) = \frac{1}{N}\sum_{i=1}^{N} \delta_{(c_i^k, w_i^k)}(dc, dw), \quad g_{\theta^k}^N(x) = \int c\,\sigma(w \cdot x)\,\nu_k^N(dc, dw).$$

Empirical Measure Representation

Empirical Measure of Neuron Parameters

$$\nu_k^N(dc, dw) = \frac{1}{N}\sum_{i=1}^{N} \delta_{(c_i^k, w_i^k)}(dc, dw), \quad g_{\theta^k}^N(x) = \int c\,\sigma(w\cdot x)\,\nu_k^N(dc, dw).$$

- Network output becomes an average over the neuron distribution.
- Each update slightly changes the empirical measure ν_k^N.
- This is the **mean-field representation** of the network.

Time Scaling for Mean-Field Limit

Need for Scaling
Since $f_N(x) = \frac{1}{N}\sum_i c_i \sigma(w_i \cdot x)$, each gradient step is $\mathcal{O}(1/N)$.
Without rescaling, dynamics vanish as $N \to \infty$.

Time Scaling for Mean-Field Limit

Need for Scaling
Since $f_N(x) = \frac{1}{N}\sum_i c_i\sigma(w_i \cdot x)$, each gradient step is $\mathcal{O}(1/N)$.
Without rescaling, dynamics vanish as $N \to \infty$.

Rescaled Continuous Time
Define $t = k\alpha_N/N$ and interpolate the SGD:

$$\frac{d\theta_i(t)}{dt} = -\nabla_{\theta_i}\mathcal{L}_N(\theta(t)).$$

Time Scaling for Mean-Field Limit

Need for Scaling
Since $f_N(x) = \frac{1}{N}\sum_i c_i\sigma(w_i \cdot x)$, each gradient step is $\mathcal{O}(1/N)$.
Without rescaling, dynamics vanish as $N \to \infty$.

Rescaled Continuous Time
Define $t = k\alpha_N/N$ and interpolate the SGD:

$$\frac{d\theta_i(t)}{dt} = -\nabla_{\theta_i}\mathcal{L}_N(\theta(t)).$$

- Each neuron moves slowly ($O(1/N)$).
- The collective distribution evolves on an $O(1)$ time scale.

Mean-Field Limit: PDE for the Parameter Distribution

Limit as $N \to \infty$
The empirical measure ν_t^N converges to ρ_t, which satisfies:

$$\partial_t \rho_t + \nabla_\theta \cdot \big(\rho_t \, v_t(\theta)\big) = 0.$$

$$v_t(\theta) = -\nabla_\theta \mathcal{L}(\rho_t).$$

Mean-Field Limit: PDE for the Parameter Distribution

Limit as $N \to \infty$

The empirical measure ν_t^N converges to ρ_t, which satisfies:

$$\partial_t \rho_t + \nabla_\theta \cdot \big(\rho_t \, v_t(\theta)\big) = 0.$$

$$v_t(\theta) = -\nabla_\theta \mathcal{L}(\rho_t).$$

Equivalent Integral Form

$$f(x; \rho_t) = \int c\, \sigma(w \cdot x)\, \rho_t(dc, dw), \quad \mathcal{L}(\rho_t) = \tfrac{1}{2}\mathbb{E}\big[(Y - f(X; \rho_t))^2\big].$$

Training becomes a deterministic gradient flow of ρ_t in parameter space.

Main Results (Sirignano & Spiliopoulos, 2018)

- Law of Large Numbers:

$$\nu_t^N \Rightarrow \rho_t \quad \text{as } N \to \infty.$$

  The empirical measure converges to a deterministic limit.

Main Results (Sirignano & Spiliopoulos, 2018)

- Law of Large Numbers:

$$\nu_t^N \Rightarrow \rho_t \quad \text{as } N \to \infty.$$

  The empirical measure converges to a deterministic limit.

- Propagation of Chaos: Finite collections of neurons become independent with common law ρ_t.

Main Results (Sirignano & Spiliopoulos, 2018)

- **Law of Large Numbers:**

$$\nu_t^N \Rightarrow \rho_t \quad \text{as } N \to \infty.$$

  The empirical measure converges to a deterministic limit.
- **Propagation of Chaos:** Finite collections of neurons become independent with common law ρ_t.
- **Gradient Flow Structure:** ρ_t evolves according to a Wasserstein gradient flow of the limiting loss functional $\mathcal{L}(\rho_t)$.

Main Results (Sirignano & Spiliopoulos, 2018)

- **Law of Large Numbers:**

$$\nu_t^N \Rightarrow \rho_t \quad \text{as } N \to \infty.$$

  The empirical measure converges to a deterministic limit.
- **Propagation of Chaos:** Finite collections of neurons become independent with common law ρ_t.
- **Gradient Flow Structure:** ρ_t evolves according to a Wasserstein gradient flow of the limiting loss functional $\mathcal{L}(\rho_t)$.
- **Interpretation:** Training an infinitely wide network = evolving a probability distribution over parameters.

Summary: Particle to Mean-Field Transition

Finite Network (Particles)	Infinite Network (Mean Field)
N neurons with parameters (c_i, w_i)	Continuous density $\rho_t(c, w)$
Discrete SGD updates	PDE: $\partial_t \rho_t + \nabla \cdot (\rho_t v_t) = 0$
Randomness due to data / init.	Deterministic evolution
Output: $\frac{1}{N} \sum_i c_i \sigma(w_i \cdot x)$	Output: $\int c\sigma(w \cdot x)\rho_t$
Law of large numbers $\nu_t^N \Rightarrow \rho_t$	Gradient flow in parameter space

Training dynamics of large networks converge to deterministic mean-field equations.

From Gradient Descent to Mean-Field Dynamics

Starting Point: Training a Neural Network

- A network has many parameters (weights and biases).
- Each parameter is updated by gradient descent or SGD.
- When the number of parameters N is very large, their collective behaviour can be described statistically.

Key Idea

View each parameter as a particle moving under a force given by the gradient of the loss.

- The "population" of parameters behaves like an interacting particle system.
- Their empirical distribution captures the global learning dynamics.

From Many Particles to a Distribution

Particle Description

- Each parameter follows a deterministic (gradient) or noisy (SGD) trajectory.
- Collectively, they form an evolving cloud in parameter space.
- The shape of this cloud is summarized by its probability distribution ρ_t.

Mean-Field Limit

- As the number of parameters $N \to \infty$, random fluctuations average out.
- The distribution ρ_t evolves deterministically.
- This is the **mean-field approximation** of the learning process.

The Gradient Flow Picture

Continuous-Time Interpretation

- Gradient descent can be viewed as a continuous-time flow: parameters move downhill on the loss landscape.
- In the mean-field regime, the whole distribution ρ_t flows downhill on a *functional loss landscape*.

Intuition

- Instead of minimizing a scalar loss $L(\theta)$, we minimize a functional $\mathcal{L}(\rho)$.
- The direction of steepest descent (in the space of distributions) defines the **gradient flow**.

The Resulting Dynamics

Evolution of the Parameter Distribution

- The distribution ρ_t of parameters drifts toward regions that reduce the global loss.
- Randomness from SGD adds a small diffusion effect.
- The resulting dynamics resemble a **Fokker–Planck equation** for ρ_t.

Physical Analogy

- Parameters = particles in a potential field (the loss).
- Gradient descent = deterministic drift downhill.
- SGD noise = thermal motion (temperature $\propto$ learning rate / batch size).

Conceptual Summary

Microscopic (Finite N):

- Many particles / parameters.
- Each follows its own gradient update.
- Interactions through shared loss function.

Macroscopic (Mean-Field Limit):

- Collective behaviour described by ρ_t.
- Smooth deterministic evolution over time.
- Encoded by a mean-field or Wasserstein gradient flow.

From many interacting parameters to one evolving distribution.

Main Takeaways

- The training dynamics of large neural networks can be viewed as a continuum limit.
- The empirical parameter distribution evolves deterministically in time.
- This evolution is a **gradient flow in the space of probability measures**.
- Links deep learning to mean-field games, McKean–Vlasov equations, and optimal transport.

Mean-field methods bridge optimization, probability, and learning.

Forward-Backward Propagation of Chaos (FBPoC)

From N-Player Game to Mean Field Limit
Each player $i = 1, \ldots, N$ controls

$$\begin{cases} dX_t^{i,N} = b(X_t^{i,N}, \mu_t^N, \alpha_t^{i,N})\, dt + \sigma\, dW_t^i, \\ dY_t^{i,N} = -\partial_x H(X_t^{i,N}, \mu_t^N, Y_t^{i,N})\, dt + Z_t^{i,N}\, dW_t^i, \\ Y_T^{i,N} = \partial_x g(X_T^{i,N}, \mu_T^N), \end{cases} \qquad \mu_t^N = \frac{1}{N}\sum_{j=1}^{N} \delta_{X_t^{j,N}}.$$

- $X^{i,N}$: state (forward dynamics)
- $Y^{i,N}$: adjoint / costate (backward optimality condition)
- H: Hamiltonian of player's control problem

Forward-Backward Propagation of Chaos
As $N \to \infty$,

$$(X_t^{1,N}, Y_t^{1,N}), \dots, (X_t^{k,N}, Y_t^{k,N}) \xrightarrow{law} (X_t^1, Y_t^1), \dots, (X_t^k, Y_t^k),$$

where (X^i, Y^i) are i.i.d. copies solving the mean-field FBSDE:

$$\begin{cases} dX_t = b(X_t, \mu_t, \alpha_t)\, dt + \sigma\, dW_t, \\ dY_t = -\partial_x H(X_t, \mu_t, Y_t)\, dt + Z_t\, dW_t, \\ \mu_t = (X_t). \end{cases}$$

Interpretation: independence of both forward and backward variables in the limit.

Forward-Backward Propagation of Chaos

Meaning

- Extends classical *propagation of chaos* to coupled forward–backward systems.
- Ensures that the N-player Nash equilibrium converges to the MFG equilibrium.
- Mathematically delicate: information flows both forward (via X) and backward (via Y).

See also

Backward Propagation of Chaos mainly initiated by Lauriere and Tangpi in [2] and extended in various directions in [3] by Papapantoleon,Saplaouras and Theodorakopoulos.

Interacting Mean-Field BSDE System
For each N, consider a system of interacting BSDEs:

$$Y_t^{i,N} = \xi^{i,N} + \int_t^T f\big(s, Y_s^{i,N}, Z_s^{i,N}, (Y_s^{\cdot,N})\big)\, ds - \int_t^T Z_s^{i,N}\, dW_s^i, \quad i = 1, \dots, N.$$

Interaction comes via empirical law $(Y_s^{\cdot,N})$.

Backward Propagation of Chaos
As $N \to \infty$, one shows:

$$(Y^{1,N}, Y^{2,N}, \dots, Y^{k,N}) \to (Y^1, \dots, Y^k),$$

where each Y^i solves a **McKean–Vlasov BSDE**

$$Y_t = \xi + \int_t^T f\big(s, Y_s, Z_s, (Y_s)\big)\, ds - \int_t^T Z_s\, dW_s.$$

Stability of backward propagation of chaos was intro established by by Papapantoleon,Saplaouras and Theodorakopoulos in [4].

Stability of BPoC

$$D^k := \left(\{\overline{X}^{k,i}\}_{i\in\mathbb{N}}, T^k, \left\{\{\xi^{k,i,N}\}_{i\in N}\right\}_{N\in\mathbb{N}}, \{\xi^{k,i}\}_{i\in\mathbb{N}}, f^k\right)$$

The introduced stability: if a sequence of data sets (D^k) converges to limit data D^∞, then the interacting BSDE solutions converge to the McKean–Vlasov solution of D^∞. Small perturbations in terminal / drivers → small changes in limits.

Stability of Backward Propagation of Chaos

D^1	$S^{1,i,i}$	$S^{1,i,i+1}$	$S^{1,i,i+2}$	$\cdots$	$S^{1,i,N}$	$\xrightarrow{N\to\infty}$	$S^{1,i}$
D^2	$S^{2,i,i}$	$S^{2,i,i+1}$	$S^{2,i,i+2}$	$\cdots$	$S^{2,i,N}$	$\xrightarrow{N\to\infty}$	$S^{2,i}$
D^3	$S^{3,i,i}$	$S^{3,i,i+1}$	$S^{3,i,i+2}$	$\cdots$	$S^{3,i,N}$	$\xrightarrow{N\to\infty}$	$S^{3,i}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
D^k	$S^{k,i,i}$	$S^{k,i,i+1}$	$S^{k,i,i+2}$	$\cdots$	$S^{k,i,N}$	$\xrightarrow{N\to\infty}$	$S^{k,i}$
$\downarrow$						$\rightsquigarrow$	$\downarrow$
D^∞	$S^{\infty,i,i}$	$S^{\infty,i,i+1}$	$S^{\infty,i,i+2}$	$\cdots$	$S^{\infty,i,N}$	$\xrightarrow{N\to\infty}$	$S^{\infty,i}$

Table 1: The doubly-indexed scheme for the stability of backward propagation of chaos.

References i

R. Carmona, F. Delarue, et al.
Probabilistic theory of mean field games with applications I-II, volume 3.
Springer, 2018.

M. Laurière and L. Tangpi.
Backward propagation of chaos.
Electronic Journal of Probability, 27:1–30, 2022.

A. Papapantoleon, A. Saplaouras, and S. Theodorakopoulos.
Existence, uniqueness and propagation of chaos for general mckean-vlasov and mean-field bsdes.
arXiv preprint arXiv:2408.13758, 2024.

A. Papapantoleon, A. Saplaouras, and S. Theodorakopoulos.
Stability of backward propagation of chaos.
arXiv preprint arXiv:2506.03562, 2025.

J. Sirignano and K. Spiliopoulos.
Mean field analysis of neural networks: A law of large numbers.
SIAM Journal on Applied Mathematics, 80(2):725–752, 2020.