A Mean-Field Optimal Control Approach to the Training of NeurODEs

Benoît Bonnet

(*in collaboration with C. Cipriani, M. Fornasier and H. Huang*)

BrainPOP Seminar

January 10, 2022

Outline of the talk

Quick primer on neural networks

NeurODE models and mean-field control

Optimality conditions: Lagrangian and Hamiltonian approaches

Numerical illustrations

Outline of the talk

Quick primer on neural networks

NeurODE models and mean-field control

Optimality conditions: Lagrangian and Hamiltonian approaches

Numerical illustrations

Introduction – *Supervised learning and neural networks*

Heuristic definition (Supervised learning)

Family of schemes used to **learn a mapping** $f : \mathcal{X} \to \mathcal{Y}$ by using

- a series of **inputs** $(x_1, \ldots, x_N) \in \mathcal{X}^N$,
- matching **outputs** $(y_1, \ldots, y_N) \in \mathcal{Y}^N$,
- a **loss function** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to measure potential misfits.

Neural network (Illustration)

Introduction – *Supervised learning and neural networks*

Heuristic definition (Supervised learning)

Family of schemes used to **learn a mapping** $f : \mathcal{X} \to \mathcal{Y}$ by using

- a series of **inputs** $(x_1, \ldots, x_N) \in \mathcal{X}^N$,
- matching **outputs** $(y_1, \ldots, y_N) \in \mathcal{Y}^N$,
- a **loss function** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to measure potential misfits.

Neural network (Illustration)

Introduction – *Supervised learning and neural networks*

Heuristic definition (Supervised learning)

Family of schemes used to **learn a mapping** $f : \mathcal{X} \to \mathcal{Y}$ by using

- a series of **inputs** $(x_1, \ldots, x_N) \in \mathcal{X}^N$,
- matching **outputs** $(y_1, \ldots, y_N) \in \mathcal{Y}^N$,
- a **loss function** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to measure potential misfits.

Neural network (Illustration)

Introduction – *Supervised learning and neural networks*

Heuristic definition (Supervised learning)

Family of schemes used to **learn a mapping** $f : \mathcal{X} \to \mathcal{Y}$ by using

- a series of **inputs** $(x_1, \ldots, x_N) \in \mathcal{X}^N$,
- matching **outputs** $(y_1, \ldots, y_N) \in \mathcal{Y}^N$,
- a **loss function** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to measure potential misfits.

Neural network (Illustration)

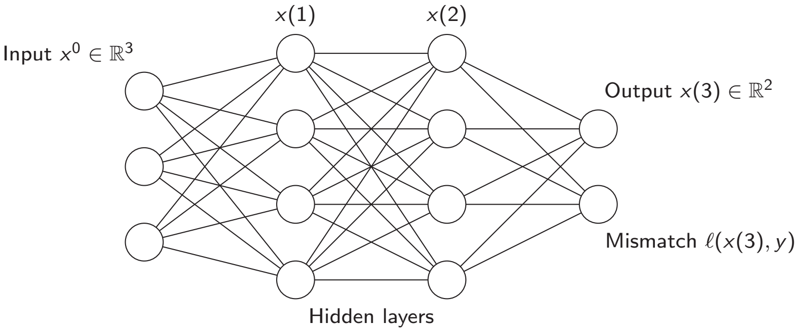
Introduction – *Supervised learning and neural networks*

Heuristic definition (Supervised learning)

Family of schemes used to **learn a mapping** $f : \mathcal{X} \to \mathcal{Y}$ by using

- ◇ a series of **inputs** $(x_1, \ldots, x_N) \in \mathcal{X}^N$,
- ◇ matching **outputs** $(y_1, \ldots, y_N) \in \mathcal{Y}^N$,
- ◇ a **loss function** $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ to measure potential misfits.

Neural network (Illustration)



Introduction – *Mathematical model for neural networks*

The update of $x(\cdot)$ from layer k to $k+1$ writes

$$x(k+1) = \rho(W_k x(k) + b_k),$$

where $k \in \{0, \dots, n-1\}$, and

- $W_k \in \mathbb{R}^{d_k \times d_{k+1}}$ are **weight matrices**,
- $b_k \in \mathbb{R}^{d_{k+1}}$ are called the **biases**,
- $\rho : \mathbb{R} \to \mathbb{R}$ is a componentwise **activation function**.

Idea: Network training $\rightsquigarrow$ **expected risk** minimisation

Statement (Training as a stochastic optimisation problem)

Assuming that (x_i, y_i) are **sampled** from $\mu^0 \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, solve

$$\begin{cases} \min\limits_{(W_k, b_k)} \mathbb{E}_{\mu^0}\Big[\ell(x(n), y)\Big], \\ \text{s.t.} \quad x(k+1) = \rho(W_k x(k) + b_k) \quad \text{for } k \in \{0, \dots, n-1\}. \end{cases}$$

Introduction – *Mathematical model for neural networks*

The update of $x(\cdot)$ from layer k to $k+1$ writes

$$x(k+1) = \rho(W_k x(k) + b_k),$$

where $k \in \{0, \dots, n-1\}$, and

- $W_k \in \mathbb{R}^{d_k \times d_{k+1}}$ are **weight matrices**,
- $b_k \in \mathbb{R}^{d_{k+1}}$ are called the **biases**,
- $\rho : \mathbb{R} \to \mathbb{R}$ is a componentwise **activation function**.

Idea: Network training $\rightsquigarrow$ **expected risk** minimisation

Statement (Training as a stochastic optimisation problem)

Assuming that (x_i, y_i) are **sampled** from $\mu^0 \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, solve

$$\begin{cases} \min\limits_{(W_k, b_k)} \mathbb{E}_{\mu^0}\Big[\ell(x(n), y)\Big], \\ \text{s.t. } \; x(k+1) = \rho(W_k x(k) + b_k) \;\; \text{for } k \in \{0, \dots, n-1\}. \end{cases}$$

Introduction – *Mathematical model for neural networks*

The update of $x(\cdot)$ from layer k to $k+1$ writes

$$x(k+1) = \rho(W_k x(k) + b_k),$$

where $k \in \{0, \dots, n-1\}$, and

- $W_k \in \mathbb{R}^{d_k \times d_{k+1}}$ are **weight matrices**,
- $b_k \in \mathbb{R}^{d_{k+1}}$ are called the **biases**,
- $\rho : \mathbb{R} \to \mathbb{R}$ is a componentwise **activation function**.

Idea: Network training $\rightsquigarrow$ **expected risk** minimisation

Statement (Training as a stochastic optimisation problem)
Assuming that (x_i, y_i) are **sampled** from $\mu^0 \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, solve

$$\begin{cases} \min\limits_{(W_k, b_k)} \mathbb{E}_{\mu^0}\Big[\ell(x(n), y)\Big], \\ \text{s.t.} \quad x(k+1) = \rho(W_k x(k) + b_k) \quad \text{for } k \in \{0, \dots, n-1\}. \end{cases}$$

Introduction – *Mathematical model for neural networks*

The update of $x(\cdot)$ from layer k to $k+1$ writes

$$x(k+1) = \rho(W_k x(k) + b_k),$$

where $k \in \{0, \ldots, n-1\}$, and

- $W_k \in \mathbb{R}^{d_k \times d_{k+1}}$ are **weight matrices**,
- $b_k \in \mathbb{R}^{d_{k+1}}$ are called the **biases**,
- $\rho : \mathbb{R} \to \mathbb{R}$ is a componentwise **activation function**.

Idea: Network training $\rightsquigarrow$ **expected risk** minimisation

Statement (Training as a stochastic optimisation problem)

Assuming that (x_i, y_i) are **sampled** from $\mu^0 \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, solve

$$\begin{cases} \min\limits_{(W_k, b_k)} \mathbb{E}_{\mu^0}\Big[\ell(x(n), y)\Big], \\ \text{s.t.} \quad x(k+1) = \rho(W_k x(k) + b_k) \quad \text{for } k \in \{0, \ldots, n-1\}. \end{cases}$$

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]

k-th hidden layer

Remarks (Concerning residual blocks)

- **Con:** rectangular networks only ⇝ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to mathematical analysis.

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]

k-th hidden layer

Remarks (Concerning residual blocks)

- **Con:** rectangular networks only ⇝ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to mathematical analysis.

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]

k-th hidden layer

Remarks (Concerning residual blocks)

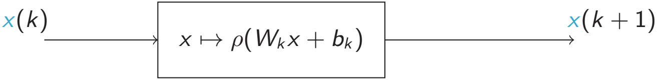
- **Con:** rectangular networks only $\rightsquigarrow$ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to mathematical analysis!

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]



k-th hidden layer

Remarks (Concerning residual blocks)

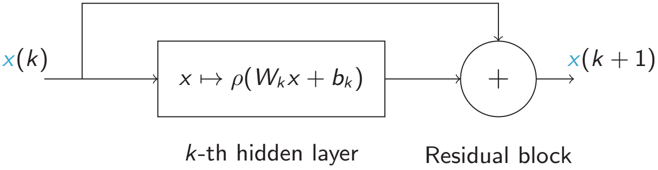
- **Con:** rectangular networks only ⇝ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to mathematical analysis.

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]



Remarks (Concerning residual blocks)

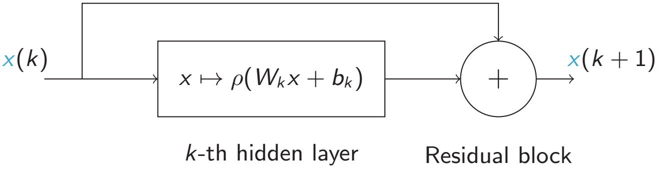
- **Con:** rectangular networks only ⇝ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to mathematical analysis.

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]



Remarks (Concerning residual blocks)

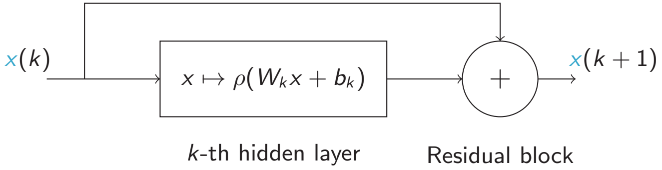
- **Con:** rectangular networks only $\rightsquigarrow$ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to **mathematical analysis**!

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]



Remarks (Concerning residual blocks)

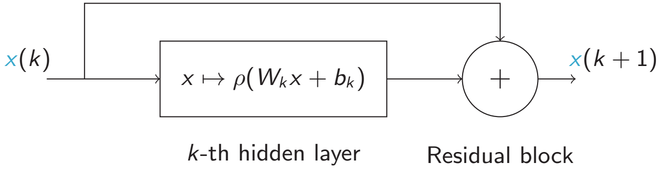
- **Con:** rectangular networks only $\rightsquigarrow$ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to **mathematical analysis**!

Introduction – *The concept of residual block*

Main limitations (Stability and explainability)

1. Their accuracy may **decrease** as the depth **increases**.
2. Few **theoretical certificates** explain why they work so well.

Idea: Regularise the network by inserting **residual blocks** [HZ'16]



Remarks (Concerning residual blocks)

- **Con:** rectangular networks only $\rightsquigarrow$ need to add **constraints**
- **Pros:** 1) Improved **stability** for **deep** networks.
  2) Opens the door to **mathematical analysis**!

Outline of the talk

Quick primer on neural networks

NeurODE models and mean-field control

Optimality conditions: Lagrangian and Hamiltonian approaches

Numerical illustrations

NeurODEs – *Continuous approximation of deep networks*

Observation: For networks with many layers, the update

$$x(k+1) = x(k) + \rho(W_k x(k) + b_k),$$

can be seen as the **Euler approximation** of the **NeurODE**

$$\dot{x}(t) = \rho(W(t)x(t) + b(t)).$$

↪ Recast questions on **deep networks** as **control problems**!

Control of NeurODEs (Some literature overview)

- **Learning** procedure ⇝ **stochastic optimal control** problem (see e.g. [E'17, EH'17, JSS'21]).
- **Expressivity** of deep networks ⇝ **controllability properties** of NeurODEs (see e.g. [AS'20&21, TG'20, S'21]).

↪ Reformulation as a **mean-field optimal control** problem.

NeurODEs – *Continuous approximation of deep networks*

Observation: For networks with many layers, the update

$$x(k+1) = x(k) + \rho(W_k x(k) + b_k),$$

can be seen as the **Euler approximation** of the **NeurODE**

$$\dot{x}(t) = \rho(W(t)x(t) + b(t)).$$

↪ Recast questions on **deep networks** as **control problems**!

Control of NeurODEs (Some literature overview)

- **Learning** procedure ⇝ **stochastic optimal control** problem (see e.g. [E'17, EH'17, JSS'21]).
- **Expressivity** of deep networks ⇝ **controllability properties** of NeurODEs (see e.g. [AS'20&21, TG'20, S'21]).

↪ Reformulation as a **mean-field optimal control** problem.

NeurODEs – *Continuous approximation of deep networks*

Observation: For networks with many layers, the update

$$x(k+1) = x(k) + \rho(W_k x(k) + b_k),$$

can be seen as the **Euler approximation** of the **NeurODE**

$$\dot{x}(t) = \rho(W(t)x(t) + b(t)).$$

$\hookrightarrow$ Recast questions on **deep networks** as **control problems**!

Control of NeurODEs (Some literature overview)

- **Learning** procedure $\rightsquigarrow$ **stochastic optimal control** problem (see e.g. [E'17, EH'17, JSS'21]).
- **Expressivity** of deep networks $\rightsquigarrow$ **controllability properties** of NeurODEs (see e.g. [AS'20&21, TG'20, S'21]).

$\hookrightarrow$ Reformulation as a **mean-field optimal control** problem.

NeurODEs – *Continuous approximation of deep networks*

Observation: For networks with many layers, the update

$$x(k+1) = x(k) + \rho(W_k x(k) + b_k),$$

can be seen as the **Euler approximation** of the **NeurODE**

$$\dot{x}(t) = \rho(W(t)x(t) + b(t)).$$

$\hookrightarrow$ Recast questions on **deep networks** as **control problems**!

Control of NeurODEs (Some literature overview)

- **Learning** procedure $\rightsquigarrow$ **stochastic optimal control** problem (see e.g. [E'17, EH'17, JSS'21]).
- **Expressivity** of deep networks $\rightsquigarrow$ **controllability properties** of NeurODEs (see e.g. [AS'20&21, TG'20, S'21]).

$\hookrightarrow$ Reformulation as a **mean-field optimal control** problem.

NeurODEs – *Continuous approximation of deep networks*

Observation: For networks with many layers, the update

$$x(k+1) = x(k) + \rho(W_k x(k) + b_k),$$

can be seen as the **Euler approximation** of the **NeurODE**

$$\dot{x}(t) = \rho(W(t)x(t) + b(t)).$$

$\hookrightarrow$ Recast questions on **deep networks** as **control problems**!

Control of NeurODEs (Some literature overview)

- $\diamond$ **Learning** procedure $\rightsquigarrow$ **stochastic optimal control** problem (see e.g. [E'17, EH'17, JSS'21]).
- $\diamond$ **Expressivity** of deep networks $\rightsquigarrow$ **controllability properties** of NeurODEs (see e.g. [AS'20&21, TG'20, S'21]).

$\hookrightarrow$ Reformulation as a **mean-field optimal control** problem.

NeurODEs – *Continuous approximation of deep networks*

Observation: For networks with many layers, the update

$$x(k+1) = x(k) + \rho(W_k x(k) + b_k),$$

can be seen as the **Euler approximation** of the **NeurODE**

$$\dot{x}(t) = \rho(W(t)x(t) + b(t)).$$

↪ Recast questions on **deep networks** as **control problems**!

Control of NeurODEs (Some literature overview)

- ◇ **Learning** procedure ⇝ **stochastic optimal control** problem (see e.g. [E'17, EH'17, JSS'21]).
- ◇ **Expressivity** of deep networks ⇝ **controllability properties** of NeurODEs (see e.g. [AS'20&21, TG'20, S'21]).

↪ Reformulation as a **mean-field optimal control** problem.

NeurODEs – *Continuous approximation of deep networks*

Observation: For networks with many layers, the update

$$x(k+1) = x(k) + \rho(W_k x(k) + b_k),$$

can be seen as the **Euler approximation** of the **NeurODE**

$$\dot{x}(t) = \rho(W(t)x(t) + b(t)).$$

$\hookrightarrow$ Recast questions on **deep networks** as **control problems**!

Control of NeurODEs (Some literature overview)

- ⋄ **Learning** procedure ⇝ **stochastic optimal control** problem (see e.g. [E'17, EH'17, JSS'21]).
- ⋄ **Expressivity** of deep networks ⇝ **controllability properties** of NeurODEs (see e.g. [AS'20&21, TG'20, S'21]).

$\hookrightarrow$ Reformulation as a **mean-field optimal control** problem.

NeurODEs – *From stochastic to mean-field control*

The **continuous-time** version of the training problem writes

$$\begin{cases} \min\limits_{\theta(\cdot)} \left[\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \dot{X}(t) = \mathcal{F}(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\ (X(0), Y(0)) \sim \mu^0, \end{cases} \end{cases}$$

where $\theta(\cdot)$ are **controls** and $\lambda > 0$ is **regularisation** parameter.

Facts: the law $\mu(t) := \mathcal{L}(X(t), Y(t))$ solves the **transport PDE**

$$\partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta(t))\mu(t)\big) = 0,$$

and

$$\mathbb{E}_{\mu^0}\Big[\ell(X(T), Y(T))\Big] = \int_{\mathcal{X}^2} \ell(x, y)\mathrm{d}\mu(T)(x, y).$$

Idea: **Learning** as **linear** optimal control on **measures**!

NeurODEs – *From stochastic to mean-field control*

The **continuous-time** version of the training problem writes

$$\begin{cases} \min_{\theta(\cdot)} \left[\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \dot{X}(t) = \mathcal{F}(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\ (X(0), Y(0)) \sim \mu^0, \end{cases} \end{cases}$$

where $\theta(\cdot)$ are **controls** and $\lambda > 0$ is **regularisation** parameter.

Facts: the law $\mu(t) := \mathcal{L}(X(t), Y(t))$ solves the **transport PDE**

$$\partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta(t))\mu(t)\big) = 0,$$

and

$$\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] = \int_{\mathcal{X}^2} \ell(x, y)\mathrm{d}\mu(T)(x, y).$$

Idea: **Learning** as **linear** optimal control on **measures**!

NeurODEs – *From stochastic to mean-field control*

The **continuous-time** version of the training problem writes

$$\begin{cases} \min_{\theta(\cdot)} \left[\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t\right], \\ \text{s.t.} \begin{cases} \dot{X}(t) = \mathcal{F}(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\ (X(0), Y(0)) \sim \mu^0, \end{cases} \end{cases}$$

where $\theta(\cdot)$ are **controls** and $\lambda > 0$ is **regularisation** parameter.

Facts: the law $\mu(t) := \mathcal{L}(X(t), Y(t))$ solves the **transport PDE**

$$\partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta(t))\mu(t)\big) = 0,$$

and

$$\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] = \int_{\mathcal{X}^2} \ell(x, y)\mathrm{d}\mu(T)(x, y).$$

Idea: **Learning** as **linear** optimal control on **measures**!

NeurODEs – *From stochastic to mean-field control*

The **continuous-time** version of the training problem writes

$$\begin{cases} \min_{\theta(\cdot)} \left[\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \dot{X}(t) = \mathcal{F}(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\ (X(0), Y(0)) \sim \mu^0, \end{cases} \end{cases}$$

where $\theta(\cdot)$ are **controls** and $\lambda > 0$ is **regularisation** parameter.

Facts: the law $\mu(t) := \mathcal{L}(X(t), Y(t))$ solves the **transport PDE**

$$\partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta(t))\mu(t)\big) = 0,$$

and

$$\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] = \int_{\mathcal{X}^2} \ell(x, y)\mathrm{d}\mu(T)(x, y).$$

Idea: **Learning** as **linear** optimal control on **measures**!

NeurODEs – *From stochastic to mean-field control*

The **continuous-time** version of the training problem writes

$$\begin{cases} \min\limits_{\theta(\cdot)} \left[\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \dot{X}(t) = \mathcal{F}(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\ (X(0), Y(0)) \sim \mu^0, \end{cases} \end{cases}$$

where $\theta(\cdot)$ are **controls** and $\lambda > 0$ is **regularisation** parameter.

Facts: the law $\mu(t) := \mathcal{L}(X(t), Y(t))$ solves the **transport PDE**

$$\partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta(t))\mu(t)\big) = 0,$$

and

$$\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] = \int_{\mathcal{X}^2} \ell(x, y)\mathrm{d}\mu(T)(x, y).$$

Idea: **Learning** as **linear** optimal control on **measures**!

NeurODEs – *From stochastic to mean-field control*

The **continuous-time** version of the training problem writes

$$\begin{cases} \min\limits_{\theta(\cdot)} \left[\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \dot{X}(t) = \mathcal{F}(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\ (X(0), Y(0)) \sim \mu^0, \end{cases} \end{cases}$$

where $\theta(\cdot)$ are **controls** and $\lambda > 0$ is **regularisation** parameter.

Facts: the law $\mu(t) := \mathcal{L}(X(t), Y(t))$ solves the **transport PDE**

$$\partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta(t))\mu(t)\big) = 0,$$

and

$$\mathbb{E}_{\mu^0}\left[\ell(X(T), Y(T))\right] = \int_{\mathcal{X}^2} \ell(x, y)\mathrm{d}\mu(T)(x, y).$$

Idea: **Learning** as **linear** optimal control on **measures**!

NeurODEs – *Mean-field control formulation of learning*

Definition (Training as a mean-field optimal control problem)

$$\begin{cases} \displaystyle\min_{\theta(\cdot)} \left[\int_{\mathcal{X}^2} \ell(x,y)\mathrm{d}\mu(T)(x,y) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta(t))\mu(t)\big) = 0, \\ \mu(0) = \mu^0 \in \mathcal{P}(\mathcal{X}^2). \end{cases} \end{cases}$$

$\hookrightarrow$ Wealth of **mathematical tools** to study these problems!

Mean-field control (Short literature overview)

- Existence, well-posedness and **regularity** results (see e.g. [BF'20, BR'21, CLOS'22, FPR'14, FS'14, FLOS'19, P'16]).
- **Optimality conditions**
  1) **DP** [AL'19, AL'20, BaF'21, BF'22, CMNP'18, CMP'20, JMQ'21]
  2) **Pontryagin** [B'19, BR'19, BF'21, BFRS'17, P'16, PS'21]
  3) **Lagrangian** [BCFH'22, BPTT'20, BPTT'21].

NeurODEs – *Mean-field control formulation of learning*

Definition (Training as a mean-field optimal control problem)

$$\begin{cases} \min\limits_{\theta(\cdot)} \left[\int_{\mathcal{X}^2} \ell(x,y)\mathrm{d}\mu(T)(x,y) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta(t))\mu(t)\big) = 0, \\ \mu(0) = \mu^0 \in \mathcal{P}(\mathcal{X}^2). \end{cases} \end{cases}$$

$\hookrightarrow$ Wealth of **mathematical tools** to study these problems!

Mean-field control (Short literature overview)

- Existence, well-posedness and **regularity** results (see e.g. [BF'20, BR'21, CLOS'22, FPR'14, FS'14, FLOS'19, P'16]).
- **Optimality conditions**
  1) **DP** [AL'19, AL'20, BaF'21, BF'22, CMNP'18, CMP'20, JMQ'21]
  2) **Pontryagin** [B'19, BR'19, BF'21, BFRS'17, P'16, PS'21]
  3) **Lagrangian** [BCFH'22, BPTT'20, BPTT'21].

NeurODEs – *Mean-field control formulation of learning*

Definition (Training as a mean-field optimal control problem)

$$\begin{cases} \min\limits_{\theta(\cdot)} \left[\displaystyle\int_{\mathcal{X}^2} \ell(x,y)\mathrm{d}\mu(T)(x,y) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta(t))\mu(t)\big) = 0, \\ \mu(0) = \mu^0 \in \mathcal{P}(\mathcal{X}^2). \end{cases} \end{cases}$$

↪ Wealth of **mathematical tools** to study these problems!

Mean-field control (Short literature overview)

- **Existence**, well-posedness and **regularity** results (see e.g. [BF'20, BR'21, CLOS'22, FPR'14, FS'14, FLOS'19, P'16]).
- **Optimality conditions**
  1) **DP** [AL'19, AL'20, BaF'21, BF'22, CMNP'18, CMP'20, JMQ'21]
  2) **Pontryagin** [B'19, BR'19, BF'21, BFRS'17, P'16, PS'21]
  3) **Lagrangian** [BCFH'22, BPTT'20, BPTT'21].

NeurODEs – *Mean-field control formulation of learning*

Definition (Training as a mean-field optimal control problem)

$$\begin{cases} \min_{\theta(\cdot)} \left[\int_{\mathcal{X}^2} \ell(x,y)\mathrm{d}\mu(T)(x,y) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta(t))\mu(t)\big) = 0, \\ \mu(0) = \mu^0 \in \mathcal{P}(\mathcal{X}^2). \end{cases} \end{cases}$$

$\hookrightarrow$ Wealth of **mathematical tools** to study these problems!

Mean-field control (Short literature overview)

- **Existence**, well-posedness and **regularity** results (see e.g. [BF'20, BR'21, CLOS'22, FPR'14, FS'14, FLOS'19, P'16]).
- **Optimality conditions**
  1) **DP** [AL'19, AL'20, BaF'21, BF'22, CMNP'18, CMP'20, JMQ'21]
  2) **Pontryagin** [B'19, BR'19, BF'21, BFRS'17, P'16, PS'21]
  3) **Lagrangian** [BCFH'22, BPTT'20, BPTT'21].

NeurODEs – *Mean-field control formulation of learning*

Definition (Training as a mean-field optimal control problem)

$$\begin{cases} \min\limits_{\theta(\cdot)} \left[\displaystyle\int_{\mathcal{X}^2} \ell(x,y)\mathrm{d}\mu(T)(x,y) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \right], \\ \text{s.t.} \begin{cases} \partial_t \mu(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta(t))\mu(t)\big) = 0, \\ \mu(0) = \mu^0 \in \mathcal{P}(\mathcal{X}^2). \end{cases} \end{cases}$$

$\hookrightarrow$ Wealth of **mathematical tools** to study these problems!

Mean-field control (Short literature overview)

- ◇ **Existence**, well-posedness and **regularity** results (see e.g. [BF'20, BR'21, CLOS'22, FPR'14, FS'14, FLOS'19, P'16]).
- ◇ **Optimality conditions**
  1) **DP** [AL'19, AL'20, BaF'21, BF'22, CMNP'18, CMP'20, JMQ'21]
  2) **Pontryagin** [B'19, BR'19, BF'21, BFRS'17, P'16, PS'21]
  3) **Lagrangian** [BCFH'22, BPTT'20, BPTT'21].

Outline of the talk

Quick primer on neural networks

NeurODE models and mean-field control

Optimality conditions: Lagrangian and Hamiltonian approaches

Numerical illustrations

Optimality Conditions – *General statement*

Theorem (Characterisation of optimal solutions)[BCFH'22]

When $\lambda > 0$ is large, there exist **optimal pairs** $(\mu^*(\cdot), \theta^*(\cdot))$, and they **exactly coincide** with the solutions of the **optimality system**

$$\begin{cases} \partial_t \mu^*(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta^*(t))\mu^*(t)\big) = 0, & \mu^*(0) = \mu^0, \\ \partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t,\theta^*(t)) \rangle = 0, & \psi^*(T) = \ell, \\ \theta^*(t) = -\dfrac{1}{\lambda} \displaystyle\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t,\theta^*(t))^\top \psi^*(t) \mathrm{d}\mu^*(t), & \end{cases}$$

where $\psi^* \in C^0([0,T] \times \mathcal{X}^2, \mathcal{X}^2)$ is a **Lagrange multiplier**.

Remarks (On the optimality system)

- NSC by **fixed-point** $\rightsquigarrow$ ensures **numerical convergence**.
- **Efficient methods** available to solve **each equation**.
- Allows to derive **quantitative generalisation errors**.

Optimality Conditions – *General statement*

Theorem (Characterisation of optimal solutions)[BCFH'22]

When $\lambda > 0$ is large, there exist **optimal pairs** $(\mu^*(\cdot), \theta^*(\cdot))$, and they **exactly coincide** with the solutions of the **optimality system**

$$\begin{cases} \partial_t \mu^*(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta^*(t))\mu^*(t)\big) = 0, & \mu^*(0) = \mu^0, \\ \partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t,\theta^*(t))\rangle = 0, & \psi^*(T) = \ell, \\ \theta^*(t) = -\dfrac{1}{\lambda}\displaystyle\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t,\theta^*(t))^\top \psi^*(t) \mathrm{d}\mu^*(t), \end{cases}$$

where $\psi^* \in C^0([0,T] \times \mathcal{X}^2, \mathcal{X}^2)$ is a **Lagrange multiplier**.

Remarks (On the optimality system)

- NSC by **fixed-point** $\rightsquigarrow$ ensures **numerical convergence**.
- **Efficient methods** available to solve **each equation**.
- Allows to derive **quantitative generalisation errors**.

Optimality Conditions – *General statement*

Theorem (Characterisation of optimal solutions)[BCFH'22]

When $\lambda > 0$ is large, there exist **optimal pairs** $(\mu^*(\cdot), \theta^*(\cdot))$, and they **exactly coincide** with the solutions of the **optimality system**

$$\begin{cases} \partial_t \mu^*(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta^*(t))\mu^*(t)\big) = 0, & \mu^*(0) = \mu^0, \\ \partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t,\theta^*(t))\rangle = 0, & \psi^*(T) = \ell, \\ \theta^*(t) = -\dfrac{1}{\lambda}\displaystyle\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t,\theta^*(t))^\top \psi^*(t) \mathrm{d}\mu^*(t), & \end{cases}$$

where $\psi^* \in C^0([0,T] \times \mathcal{X}^2, \mathcal{X}^2)$ is a **Lagrange multiplier**.

Remarks (On the optimality system)

- NSC by **fixed-point** $\rightsquigarrow$ ensures **numerical convergence**.
- **Efficient methods** available to solve **each equation**.
- Allows to derive **quantitative generalisation errors**.

Optimality Conditions – *General statement*

Theorem (Characterisation of optimal solutions)[BCFH'22]

When $\lambda > 0$ is large, there exist **optimal pairs** $(\mu^*(\cdot), \theta^*(\cdot))$, and they **exactly coincide** with the solutions of the **optimality system**

$$\begin{cases} \partial_t \mu^*(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta^*(t))\mu^*(t)\big) = 0, \quad \mu^*(0) = \mu^0, \\ \partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t, \theta^*(t))\rangle = 0, \quad \psi^*(T) = \ell, \\ \theta^*(t) = -\dfrac{1}{\lambda}\displaystyle\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \psi^*(t) \mathrm{d}\mu^*(t), \end{cases}$$

where $\psi^* \in C^0([0, T] \times \mathcal{X}^2, \mathcal{X}^2)$ is a **Lagrange multiplier**.

Remarks (On the optimality system)

- NSC by **fixed-point** $\rightsquigarrow$ ensures **numerical convergence**.
- **Efficient methods** available to solve **each equation**.
- Allows to derive **quantitative generalisation errors**.

Optimality Conditions – *General statement*

Theorem (Characterisation of optimal solutions)[BCFH'22]

When $\lambda > 0$ is large, there exist **optimal pairs** $(\mu^*(\cdot), \theta^*(\cdot))$, and they **exactly coincide** with the solutions of the **optimality system**

$$\begin{cases} \partial_t \mu^*(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta^*(t))\mu^*(t)\big) = 0, & \mu^*(0) = \mu^0, \\ \partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t, \theta^*(t))\rangle = 0, & \psi^*(T) = \ell, \\ \theta^*(t) = -\dfrac{1}{\lambda} \displaystyle\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \psi^*(t) \mathrm{d}\mu^*(t), \end{cases}$$

where $\psi^* \in C^0([0,T] \times \mathcal{X}^2, \mathcal{X}^2)$ is a **Lagrange multiplier**.

Remarks (On the optimality system)

- NSC by **fixed-point** $\rightsquigarrow$ ensures **numerical convergence**.
- **Efficient methods** available to solve **each equation**.
- Allows to derive **quantitative generalisation errors**.

Optimality Conditions – *General statement*

Theorem (Characterisation of optimal solutions)[BCFH'22]

When $\lambda > 0$ is large, there exist **optimal pairs** $(\mu^*(\cdot), \theta^*(\cdot))$, and they **exactly coincide** with the solutions of the **optimality system**

$$\begin{cases} \partial_t \mu^*(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta^*(t))\mu^*(t)\big) = 0, & \mu^*(0) = \mu^0, \\ \partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t,\theta^*(t))\rangle = 0, & \psi^*(T) = \ell, \\ \theta^*(t) = -\dfrac{1}{\lambda}\displaystyle\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t,\theta^*(t))^\top \psi^*(t) \mathrm{d}\mu^*(t), \end{cases}$$

where $\psi^* \in C^0([0,T] \times \mathcal{X}^2, \mathcal{X}^2)$ is a **Lagrange multiplier**.

Remarks (On the optimality system)

- NSC by **fixed-point** $\rightsquigarrow$ ensures **numerical convergence**.
- **Efficient methods** available to solve **each equation**.
- Allows to derive **quantitative generalisation errors**.

Optimality Conditions – *General statement*

Theorem (Characterisation of optimal solutions)[BCFH'22]

When $\lambda > 0$ is large, there exist **optimal pairs** $(\mu^*(\cdot), \theta^*(\cdot))$, and they **exactly coincide** with the solutions of the **optimality system**

$$\begin{cases} \partial_t \mu^*(t) + \mathrm{div}_x\big(\mathcal{F}(t,\theta^*(t))\mu^*(t)\big) = 0, \quad \mu^*(0) = \mu^0, \\ \partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t,\theta^*(t))\rangle = 0, \quad \psi^*(T) = \ell, \\ \theta^*(t) = -\dfrac{1}{\lambda} \displaystyle\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t,\theta^*(t))^\top \psi^*(t) \mathrm{d}\mu^*(t), \end{cases}$$

where $\psi^* \in C^0([0,T] \times \mathcal{X}^2, \mathcal{X}^2)$ is a **Lagrange multiplier**.

Remarks (On the optimality system)

- NSC by **fixed-point** $\rightsquigarrow$ ensures **numerical convergence**.
- **Efficient methods** available to solve **each equation**.
- Allows to derive **quantitative generalisation errors**.

Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

1. Define the **Lagrangian** of the problem by

$$\mathcal{L}(\mu,\psi,\theta) := \int_{\mathcal{X}^2} \ell \, \mathrm{d}\mu(T) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t$$
$$+ \int_{\mathcal{X}^2} \psi(0)\mathrm{d}\mu^0 - \int_{\mathcal{X}^2} \psi(T)\mathrm{d}\mu(T)$$
$$+ \int_0^T \int_{\mathcal{X}^2} \Big(\partial_t \psi(t) + \big\langle \nabla_x \psi(t), \mathcal{F}(t,\theta(t))\big\rangle\Big)\mathrm{d}\mu(t)\mathrm{d}t.$$

2. Abstract **KKT rule** in **Banach spaces** $\rightsquigarrow$ there exists ψ^* s.t.

$$\frac{\delta \mathcal{L}}{\delta \mu}(\mu^*,\psi^*,\theta^*) = 0 \qquad \text{and} \qquad \frac{\delta \mathcal{L}}{\delta \theta}(\mu^*,\psi^*,\theta^*) = 0.$$

$\hookrightarrow$ **Constraint qualification** "requires" **continuous** controls.

3. Well-posedness by **Schauder's fixed-point** theorem $\rightsquigarrow$ **QED**!

Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

1. Define the **Lagrangian** of the problem by

$$\begin{aligned}\mathcal{L}(\mu,\psi,\theta) := \int_{\mathcal{X}^2} \ell \,\mathrm{d}\mu(T) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \\ + \int_{\mathcal{X}^2} \psi(0)\mathrm{d}\mu^0 - \int_{\mathcal{X}^2} \psi(T)\mathrm{d}\mu(T) \\ + \int_0^T \int_{\mathcal{X}^2} \Big(\partial_t \psi(t) + \big\langle \nabla_x \psi(t), \mathcal{F}(t,\theta(t))\big\rangle\Big)\mathrm{d}\mu(t)\mathrm{d}t.\end{aligned}$$

2. Abstract **KKT rule** in **Banach spaces** $\rightsquigarrow$ there exists ψ^* s.t.

$$\frac{\delta\mathcal{L}}{\delta\mu}(\mu^*,\psi^*,\theta^*) = 0 \qquad \text{and} \qquad \frac{\delta\mathcal{L}}{\delta\theta}(\mu^*,\psi^*,\theta^*) = 0.$$

$\hookrightarrow$ **Constraint qualification** "requires" **continuous** controls.

3. Well-posedness by **Schauder's fixed-point** theorem $\rightsquigarrow$ **QED**!

Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

1. Define the **Lagrangian** of the problem by

$$
\begin{aligned}
\mathcal{L}(\mu,\psi,\theta) := \int_{\mathcal{X}^2} \ell \,\mathrm{d}\mu(T) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \\
+ \int_{\mathcal{X}^2} \psi(0)\mathrm{d}\mu^0 - \int_{\mathcal{X}^2} \psi(T)\mathrm{d}\mu(T) \\
+ \int_0^T \int_{\mathcal{X}^2} \Big(\partial_t\psi(t) + \big\langle \nabla_x\psi(t), \mathcal{F}(t,\theta(t))\big\rangle\Big)\mathrm{d}\mu(t)\mathrm{d}t.
\end{aligned}
$$

2. Abstract **KKT rule** in **Banach spaces** $\rightsquigarrow$ there exists ψ^* s.t.

$$
\frac{\delta\mathcal{L}}{\delta\mu}(\mu^*,\psi^*,\theta^*) = 0 \qquad \text{and} \qquad \frac{\delta\mathcal{L}}{\delta\theta}(\mu^*,\psi^*,\theta^*) = 0.
$$

$\hookrightarrow$ **Constraint qualification** "requires" **continuous** controls.

3. Well-posedness by **Schauder's fixed-point** theorem $\rightsquigarrow$ **QED**!

Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

1. Define the **Lagrangian** of the problem by

$$\begin{aligned}\mathcal{L}(\mu,\psi,\theta) := &\int_{\mathcal{X}^2} \ell \, \mathrm{d}\mu(T) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t \\ &+ \int_{\mathcal{X}^2} \psi(0)\mathrm{d}\mu^0 - \int_{\mathcal{X}^2} \psi(T)\mathrm{d}\mu(T) \\ &+ \int_0^T \int_{\mathcal{X}^2} \Big(\partial_t \psi(t) + \big\langle \nabla_x \psi(t), \mathcal{F}(t,\theta(t))\big\rangle\Big)\mathrm{d}\mu(t)\mathrm{d}t.\end{aligned}$$

2. Abstract **KKT rule** in **Banach spaces** $\rightsquigarrow$ there exists ψ^* s.t.

$$\frac{\delta \mathcal{L}}{\delta \mu}(\mu^*,\psi^*,\theta^*) = 0 \qquad \text{and} \qquad \frac{\delta \mathcal{L}}{\delta \theta}(\mu^*,\psi^*,\theta^*) = 0.$$

$\hookrightarrow$ **Constraint qualification** “requires” **continuous** controls.

3. Well-posedness by **Schauder's fixed-point** theorem $\rightsquigarrow$ **QED**!

Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

1. Define the **Lagrangian** of the problem by

$$\mathcal{L}(\mu,\psi,\theta) := \int_{\mathcal{X}^2} \ell \, \mathrm{d}\mu(T) + \frac{\lambda}{2}\int_0^T |\theta(t)|^2 \mathrm{d}t + \int_{\mathcal{X}^2} \psi(0)\mathrm{d}\mu^0 - \int_{\mathcal{X}^2} \psi(T)\mathrm{d}\mu(T) + \int_0^T \int_{\mathcal{X}^2} \Big(\partial_t \psi(t) + \big\langle \nabla_x \psi(t), \mathcal{F}(t,\theta(t))\big\rangle\Big)\mathrm{d}\mu(t)\mathrm{d}t.$$

2. Abstract **KKT rule** in **Banach spaces** $\rightsquigarrow$ there exists ψ^* s.t.

$$\frac{\delta\mathcal{L}}{\delta\mu}(\mu^*,\psi^*,\theta^*) = 0 \qquad \text{and} \qquad \frac{\delta\mathcal{L}}{\delta\theta}(\mu^*,\psi^*,\theta^*) = 0.$$

$\hookrightarrow$ **Constraint qualification** “requires” **continuous** controls.

3. Well-posedness by **Schauder's fixed-point** theorem $\rightsquigarrow$ **QED**!

Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

1. Define the **Lagrangian** of the problem by

$$\mathcal{L}(\mu, \psi, \theta) := \int_{\mathcal{X}^2} \ell \, \mathrm{d}\mu(T) + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 \mathrm{d}t + \int_{\mathcal{X}^2} \psi(0) \mathrm{d}\mu^0 - \int_{\mathcal{X}^2} \psi(T) \mathrm{d}\mu(T) + \int_0^T \int_{\mathcal{X}^2} \Big(\partial_t \psi(t) + \big\langle \nabla_x \psi(t), \mathcal{F}(t, \theta(t)) \big\rangle \Big) \mathrm{d}\mu(t) \mathrm{d}t.$$

2. Abstract **KKT rule** in **Banach spaces** $\rightsquigarrow$ there exists ψ^* s.t.

$$\frac{\delta \mathcal{L}}{\delta \mu}(\mu^*, \psi^*, \theta^*) = 0 \qquad \text{and} \qquad \frac{\delta \mathcal{L}}{\delta \theta}(\mu^*, \psi^*, \theta^*) = 0.$$

$\hookrightarrow$ **Constraint qualification** "requires" **continuous** controls.

3. Well-posedness by **Schauder's fixed-point** theorem $\rightsquigarrow$ **QED**!

Proof of the optimality conditions– *Hamiltonian approach*

Proof of the optimality conditions (Hamiltonian heuristic)

1. By the **PMP** of [B'19,BF'21,BR'19], there exists $\sigma^*(\cdot)$ s.t.

$$\begin{cases} \partial_t \sigma^*(t) = -\mathrm{D}_x \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t), \quad \sigma^*(T) = -\nabla_x \ell, \\ \theta^*(t) \in \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \left[\int_{\mathcal{X}^2} \langle \sigma^*(t), \mathcal{F}(t, \theta) \rangle \mathrm{d}\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right]. \end{cases}$$

2. Because $\lambda > 0$ is large $\rightsquigarrow$ **unique** **maximiser** $\theta^*(t)$ satisfying

$$\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t) \, \mathrm{d}\mu^*(t).$$

3. **Cauchy-Lip** uniqueness $\Rightarrow \sigma^*(t) = -\nabla_x \psi^*(t) \rightsquigarrow$ **QED**!

Question (Link between both approaches)

We have **Lagrangian** $\subset$ **Hamiltonian** $\rightsquigarrow$ **Equivalence** ?

Proof of the optimality conditions– *Hamiltonian approach*

Proof of the optimality conditions (Hamiltonian heuristic)

1. By the **PMP** of [B'19,BF'21,BR'19], there exists $\sigma^*(\cdot)$ s.t.

$$\begin{cases} \partial_t \sigma^*(t) = -\mathrm{D}_x \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t), \quad \sigma^*(T) = -\nabla_x \ell, \\ \theta^*(t) \in \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \Big[\int_{\mathcal{X}^2} \langle \sigma^*(t), \mathcal{F}(t,\theta) \rangle \mathrm{d}\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \Big]. \end{cases}$$

2. Because $\lambda > 0$ is large $\rightsquigarrow$ **unique maximiser** $\theta^*(t)$ satisfying

$$\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t) \, \mathrm{d}\mu^*(t).$$

3. **Cauchy-Lip** uniqueness $\Rightarrow \sigma^*(t) = -\nabla_x \psi^*(t) \rightsquigarrow$ **QED**!

Question (Link between both approaches)

We have **Lagrangian** $\subset$ **Hamiltonian** $\rightsquigarrow$ **Equivalence** ?

Proof of the optimality conditions– *Hamiltonian approach*

Proof of the optimality conditions (Hamiltonian heuristic)

1. By the **PMP** of [B'19,BF'21,BR'19], there exists $\sigma^*(\cdot)$ s.t.

$$\begin{cases} \partial_t \sigma^*(t) = -\mathrm{D}_x \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t), \quad \sigma^*(T) = -\nabla_x \ell, \\ \theta^*(t) \in \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \left[\int_{\mathcal{X}^2} \langle \sigma^*(t), \mathcal{F}(t, \theta) \rangle \mathrm{d}\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right]. \end{cases}$$

2. Because $\lambda > 0$ is large $\rightsquigarrow$ **unique maximiser** $\theta^*(t)$ satisfying

$$\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t) \, \mathrm{d}\mu^*(t).$$

3. **Cauchy-Lip** uniqueness $\Rightarrow \sigma^*(t) = -\nabla_x \psi^*(t) \rightsquigarrow$ **QED**!

Question (Link between both approaches)

We have **Lagrangian** $\subset$ **Hamiltonian** $\rightsquigarrow$ **Equivalence** ?

Proof of the optimality conditions– *Hamiltonian approach*

Proof of the optimality conditions (Hamiltonian heuristic)

1. By the **PMP** of [B'19,BF'21,BR'19], there exists $\sigma^*(\cdot)$ s.t.

$$\begin{cases} \partial_t \sigma^*(t) = -\mathrm{D}_x \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t), \quad \sigma^*(T) = -\nabla_x \ell, \\ \theta^*(t) \in \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \left[\int_{\mathcal{X}^2} \langle \sigma^*(t), \mathcal{F}(t,\theta) \rangle \mathrm{d}\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right]. \end{cases}$$

2. Because $\lambda > 0$ is large $\rightsquigarrow$ **unique maximiser** $\theta^*(t)$ satisfying

$$\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t) \, \mathrm{d}\mu^*(t).$$

3. **Cauchy-Lip** uniqueness $\Rightarrow \sigma^*(t) = -\nabla_x \psi^*(t) \rightsquigarrow$ **QED**!

Question (Link between both approaches)

We have **Lagrangian** $\subset$ **Hamiltonian** $\rightsquigarrow$ **Equivalence** ?

Proof of the optimality conditions– *Hamiltonian approach*

Proof of the optimality conditions (Hamiltonian heuristic)

1. By the **PMP** of [B'19,BF'21,BR'19], there exists $\sigma^*(\cdot)$ s.t.

$$\begin{cases} \partial_t \sigma^*(t) = -\mathrm{D}_x \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t), \quad \sigma^*(T) = -\nabla_x \ell, \\ \theta^*(t) \in \underset{\theta \in \mathbb{R}^m}{\mathrm{argmax}} \left[\int_{\mathcal{X}^2} \langle \sigma^*(t), \mathcal{F}(t,\theta) \rangle \mathrm{d}\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right]. \end{cases}$$

2. Because $\lambda > 0$ is large $\rightsquigarrow$ **unique maximiser** $\theta^*(t)$ satisfying

$$\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t) \, \mathrm{d}\mu^*(t).$$

3. **Cauchy-Lip** uniqueness $\Rightarrow \sigma^*(t) = -\nabla_x \psi^*(t) \rightsquigarrow$ **QED**!

Question (Link between both approaches)

We have **Lagrangian** $\subset$ **Hamiltonian** $\rightsquigarrow$ **Equivalence** ?

Proof of the optimality conditions– *Hamiltonian approach*

Proof of the optimality conditions (Hamiltonian heuristic)

1. By the **PMP** of [B'19,BF'21,BR'19], there exists $\sigma^*(\cdot)$ s.t.

$$\begin{cases} \partial_t \sigma^*(t) = -\mathrm{D}_x \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t), \quad \sigma^*(T) = -\nabla_x \ell, \\ \theta^*(t) \in \underset{\theta \in \mathbb{R}^m}{\operatorname{argmax}} \left[\int_{\mathcal{X}^2} \langle \sigma^*(t), \mathcal{F}(t, \theta) \rangle \mathrm{d}\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right]. \end{cases}$$

2. Because $\lambda > 0$ is large $\rightsquigarrow$ **unique maximiser** $\theta^*(t)$ satisfying

$$\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t) \, \mathrm{d}\mu^*(t).$$

3. **Cauchy-Lip** uniqueness $\Rightarrow \sigma^*(t) = -\nabla_x \psi^*(t) \rightsquigarrow$ **QED**!

Question (Link between both approaches)

We have **Lagrangian** $\subset$ **Hamiltonian** $\rightsquigarrow$ **Equivalence** ?

Outline of the talk

Numerical illustrations – *Algorithmic schemes*

Idea: Solve the optimality system with a **shooting method**

Algorithm (General framework)

Fix initial layers θ^0 and for $k = 1 \dots K_{\max}$

1. Solve **simultaneously** the forward-backward equations

$$\begin{cases} \partial_t \mu_k(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta_k(t))\mu_k(t)\big) = 0, & \mu_k(0) = \mu^0, \\ \partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), \mathcal{F}(t, \theta_k(t))\rangle = 0, & \psi_k(T) = \ell. \end{cases}$$

$\hookrightarrow$ **Particle approximation** or **semi-Lagrangian scheme**.

2. Update the **layers** by solving

$$\theta_{k+1}(t) + \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta_{k+1}(t))^\top \nabla_x \psi_k(t) \, \mathrm{d}\mu_k(t) = 0.$$

$\hookrightarrow$ **Particle approximation** of the integral and **Newton**.

Numerical illustrations – *Algorithmic schemes*

Idea: Solve the optimality system with a **shooting method**

Algorithm (General framework)

Fix initial layers θ^0 and for $k = 1 \dots K_{\max}$

1. Solve **simultaneously** the forward-backward equations

$$\begin{cases} \partial_t \mu_k(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta_k(t))\mu_k(t)\big) = 0, & \mu_k(0) = \mu^0, \\ \partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), \mathcal{F}(t, \theta_k(t))\rangle = 0, & \psi_k(T) = \ell. \end{cases}$$

$\hookrightarrow$ **Particle approximation** or **semi-Lagrangian scheme**.

2. Update the **layers** by solving

$$\theta_{k+1}(t) + \frac{1}{\lambda}\int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta_{k+1}(t))^\top \nabla_x \psi_k(t)\, \mathrm{d}\mu_k(t) = 0.$$

$\hookrightarrow$ **Particle approximation** of the integral and **Newton**.

Numerical illustrations – *Algorithmic schemes*

Idea: Solve the optimality system with a **shooting method**

Algorithm (General framework)

Fix initial layers θ^0 and for $k = 1 \dots K_{\max}$

1. Solve **simultaneously** the forward-backward equations

$$\begin{cases} \partial_t \mu_k(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta_k(t))\mu_k(t)\big) = 0, & \mu_k(0) = \mu^0, \\ \partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), \mathcal{F}(t, \theta_k(t))\rangle = 0, & \psi_k(T) = \ell. \end{cases}$$

$\hookrightarrow$ **Particle approximation** or **semi-Lagrangian scheme**.

2. Update the **layers** by solving

$$\theta_{k+1}(t) + \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta_{k+1}(t))^\top \nabla_x \psi_k(t) \, \mathrm{d}\mu_k(t) = 0.$$

$\hookrightarrow$ **Particle approximation** of the integral and **Newton**.

Numerical illustrations – *Algorithmic schemes*

Idea: Solve the optimality system with a **shooting method**

Algorithm (General framework)

Fix initial layers θ^0 and for $k = 1 \dots K_{\max}$

1. Solve **simultaneously** the forward-backward equations

$$\begin{cases} \partial_t \mu_k(t) + \operatorname{div}_x\big(\mathcal{F}(t, \theta_k(t))\mu_k(t)\big) = 0, & \mu_k(0) = \mu^0, \\ \partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), \mathcal{F}(t, \theta_k(t))\rangle = 0, & \psi_k(T) = \ell. \end{cases}$$

$\hookrightarrow$ **Particle approximation** or **semi-Lagrangian scheme**.

2. Update the **layers** by solving

$$\theta_{k+1}(t) + \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta_{k+1}(t))^\top \nabla_x \psi_k(t) \, \mathrm{d}\mu_k(t) = 0.$$

$\hookrightarrow$ **Particle approximation** of the integral and **Newton**.

Numerical illustrations – *Algorithmic schemes*

Idea: Solve the optimality system with a **shooting method**

Algorithm (General framework)

Fix initial layers θ^0 and for $k = 1 \dots K_{\max}$

1. Solve **simultaneously** the forward-backward equations

$$\begin{cases} \partial_t \mu_k(t) + \mathrm{div}_x\big(\mathcal{F}(t, \theta_k(t))\mu_k(t)\big) = 0, & \mu_k(0) = \mu^0, \\ \partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), \mathcal{F}(t, \theta_k(t))\rangle = 0, & \psi_k(T) = \ell. \end{cases}$$

$\hookrightarrow$ **Particle approximation** or **semi-Lagrangian scheme**.

2. Update the **layers** by solving

$$\theta_{k+1}(t) + \frac{1}{\lambda} \int_{\mathcal{X}^2} \mathrm{D}_\theta \mathcal{F}(t, \theta_{k+1}(t))^\top \nabla_x \psi_k(t) \, \mathrm{d}\mu_k(t) = 0.$$

$\hookrightarrow$ **Particle approximation** of the integral and **Newton**.

Numerical illustrations – *Toy example*

Example (Binary classification of points on the real line)

Let $\mu_x^0 = \mathcal{N}(0,1)$, $X_i^0 \sim \mu^0$ for $i \in \{1,\ldots,N\}$, and find $\theta^*(\cdot)$ s.t.

$$\begin{cases} X_i(T) = -1 & \text{if } X_i^0 < 0, \\ X_i(T) = \ \ 1 & \text{if } X_i^0 > 0. \end{cases}$$

$\hookrightarrow$ **Choose** $\ell(x,y) := |x-y|^2$ and **expect** $\mu_x(T) \sim \frac{1}{2}(\delta_{-1} + \delta_1)$.

Particle trajectories after learning the classifier with $\lambda > 0$ large enough.

Numerical illustrations – *Toy example*

Example (Binary classification of points on the real line)

Let $\mu_x^0 = \mathcal{N}(0,1)$, $X_i^0 \sim \mu^0$ for $i \in \{1,\ldots,N\}$, and find $\theta^*(\cdot)$ s.t.

$$\begin{cases} X_i(T) = -1 & \text{if } X_i^0 < 0, \\ X_i(T) = \ \ 1 & \text{if } X_i^0 > 0. \end{cases}$$

$\hookrightarrow$ **Choose** $\ell(x,y) := |x-y|^2$ and **expect** $\mu_x(T) \sim \frac{1}{2}(\delta_{-1} + \delta_1)$.

Particle trajectories after learning the classifier with $\lambda > 0$ large enough.

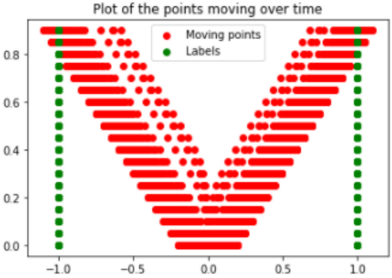
Numerical illustrations – *Toy example*

Example (Binary classification of points on the real line)

Let $\mu_x^0 = \mathcal{N}(0,1)$, $X_i^0 \sim \mu^0$ for $i \in \{1,\dots,N\}$, and find $\theta^*(\cdot)$ s.t.

$$\begin{cases} X_i(T) = -1 & \text{if } X_i^0 < 0, \\ X_i(T) = \ \ 1 & \text{if } X_i^0 > 0. \end{cases}$$

$\hookrightarrow$ **Choose** $\ell(x,y) := |x-y|^2$ and **expect** $\mu_x(T) \sim \frac{1}{2}(\delta_{-1} + \delta_1)$.



Particle trajectories after learning the classifier with $\lambda > 0$ large enough.

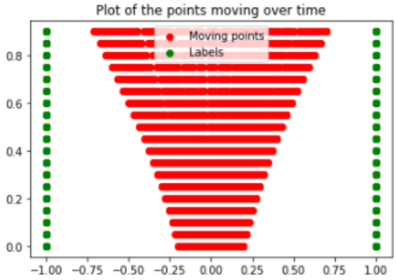
Numerical illustrations – *Toy example*

Example (Binary classification of points on the real line)

Let $\mu_x^0 = \mathcal{N}(0,1)$, $X_i^0 \sim \mu^0$ for $i \in \{1,\dots,N\}$, and find $\theta^*(\cdot)$ s.t.

$$\begin{cases} X_i(T) = -1 & \text{if } X_i^0 < 0, \\ X_i(T) = \ \ 1 & \text{if } X_i^0 > 0. \end{cases}$$

$\hookrightarrow$ **Choose** $\ell(x,y) := |x-y|^2$ and **expect** $\mu_x(T) \sim \frac{1}{2}(\delta_{-1} + \delta_1)$.



Particle trajectories after learning the classifier with $\lambda > 0$ too large.

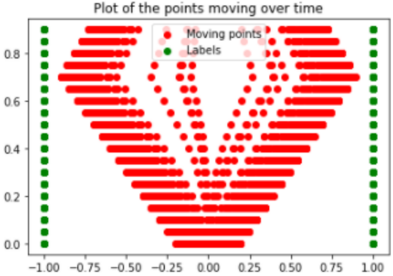
Numerical illustrations – *Toy example*

Example (Binary classification of points on the real line)

Let $\mu_x^0 = \mathcal{N}(0,1)$, $X_i^0 \sim \mu^0$ for $i \in \{1, \dots, N\}$, and find $\theta^*(\cdot)$ s.t.

$$\begin{cases} X_i(T) = -1 & \text{if } X_i^0 < 0, \\ X_i(T) = \ \ 1 & \text{if } X_i^0 > 0. \end{cases}$$

$\hookrightarrow$ **Choose** $\ell(x,y) := |x-y|^2$ and **expect** $\mu_x(T) \sim \frac{1}{2}(\delta_{-1} + \delta_1)$.



Particle trajectories after learning the classifier with $\lambda > 0$ too small.

Conclusion – *That's all folks!*

Wrap-up (Summary of results)

1. **ODE** approach to deep networks ⇝ **mathematically rich**
2. **Learning** problem ⇝ **linear** optimal control on **measures**.

Thank you for your attention !

Conclusion – *That's all folks!*

Wrap-up (Summary of results)

1. **ODE** approach to deep networks $\rightsquigarrow$ **mathematically rich**
2. **Learning** problem $\rightsquigarrow$ **linear** optimal control on **measures**.

Thank you for your attention !

Conclusion – *That's all folks!*

Wrap-up (Summary of results)

1. **ODE** approach to deep networks ⇝ **mathematically rich**
2. **Learning** problem ⇝ **linear** optimal control on **measures**.

Thank you for your attention !