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** \( f : \mathcal{X} \rightarrow \mathcal{Y} \) 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** \(\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}\) to measure potential misfits.

**Neural network (Illustration)**

\[\begin{align*}
\text{Input } x^0 \in \mathbb{R}^3 \\
\text{Hidden layers} \\
\text{Mismatch } \ell(x(3), y) \\
\text{Output } x(3) \in \mathbb{R}^2
\end{align*}\]
Introduction – *Supervised learning and neural networks*

**Heuristic definition (Supervised learning)**

Family of schemes used to **learn a mapping** $f : \mathcal{X} \rightarrow \mathcal{Y}$ 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** $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ to measure potential misfits.

**Neural network (Illustration)**

![Neural network diagram]

Input $x^0 \in \mathbb{R}^3$

Output $x(3) \in \mathbb{R}^2$

Mismatch $\ell(x(3), y)$

Hidden layers
Introduction – *Supervised learning and neural networks*

Heuristic definition (Supervised learning)

Family of schemes used to **learn a mapping** $f : \mathcal{X} \rightarrow \mathcal{Y}$ 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** $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ to measure potential misfits.

Neural network (Illustration)

![Neural network diagram](image-url)
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)**

![Neural network diagram](Image)
Introduction – *Supervised learning and neural networks*

Heuristic definition (Supervised learning)

Family of schemes used to **learn a mapping** $f : \mathcal{X} \rightarrow \mathcal{Y}$ 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** $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ to measure potential misfits.

Neural network (Illustration)
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

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

\[
\begin{cases}
\min_{(W_k, b_k)} \mathbb{E}_{\mu^0} \left[ \ell(x(n), y) \right], \\
\text{s.t. } x(k + 1) = \rho(W_k x(k) + b_k) \text{ for } k \in \{0, \ldots, n - 1\}.
\end{cases}
\]
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 $\leadsto$ expected risk minimisation

**Statement (Training as a stochastic optimisation problem)**

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

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


**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 $\leadsto$ 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

$$\min_{(W_k, b_k)} \mathbb{E}_{\mu^0} \left[ \ell(x(n), y) \right],$$

s.t. $x(k + 1) = \rho(W_k x(k) + b_k)$ for $k \in \{0, \ldots, n-1\}$. 
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]

\[
x(k) \rightarrow x \rightarrow \rho(W_kx + b_k) \rightarrow x(k + 1)
\]

*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]

\[ x(k) \xrightarrow{k-th \ hidden \ layer} x \mapsto \rho(W_k x + b_k) \xrightarrow{k}\ x(k+1) \]

**Remarks (Concerning residual blocks)**

- **Con:** rectangular networks only \( \leadsto \) 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]

\[
x(k) \rightarrow x \mapsto \rho(W_k x + b_k) \rightarrow x(k + 1)
\]

**k-th hidden layer**

Remarks (Concerning residual blocks)

- **Con:** rectangular networks only \(\sim\) 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]

\[
x(k) \xrightarrow{x \mapsto \rho(W_k x + b_k)} x(k+1)
\]

*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]

\[
\begin{align*}
x(k) & \quad \rightarrow \quad \rho(W_k x + b_k) \quad \rightarrow \quad x(k + 1) \\
\end{align*}
\]

**k-th hidden layer** \hspace{2cm} **Residual block**

Remarks (Concerning residual blocks)

- **Con:** rectangular networks only \(\leadsto\) 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]

\[
x(k) \xrightarrow{k \text{-th hidden layer}} x \mapsto \rho(W_kx + b_k) \xrightarrow{\text{Residual block}} x(k+1)
\]

Remarks (Concerning residual blocks)

- **Con**: rectangular networks only \(\sim\) 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 $\leadsto$ 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]

\[ x(k) \rightarrow x \mapsto \rho(W_k x + b_k) \rightarrow x(k + 1) \]

\(k\)-th hidden layer  Residual block

Remarks (Concerning residual blocks)

- **Con:** rectangular networks only \(\sim\) 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

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

\[ \rightarrow \text{Recast questions on deep networks as control problems!} \]

Control of NeurODEs (Some literature overview)

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

\[ \rightarrow \text{Reformulation as a *mean-field* optimal control problem.} \]
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

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

→ 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**

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

\[ \rightarrow \quad \text{Recast questions on deep networks as control problems!} \]

**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. \([AS'20&21, TG'20, S'21]\)).

\[ \rightarrow \quad \text{Reformulation as a mean-field optimal control problem.} \]
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 \( \rightsquigarrow \text{stochastic optimal control} \) problem (see e.g. [E’17, EH’17, JSS’21]).

- **Expressivity** of deep networks \( \rightsquigarrow \text{controllability properties} \) 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**

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

\[ \rightarrow \text{Recast questions on deep networks as control problems!} \]

**Control of NeurODEs (Some literature overview)**

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

\[ \rightarrow \text{Reformulation as a mean-field optimal control problem.} \]
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

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

\[ \rightarrow \] 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]).

\[ \rightarrow \] Reformulation as a **mean-field optimal control** problem.
The continuous-time version of the training problem writes

\[
\min_{\theta(\cdot)} \left[ \mathbb{E}_{\mu^0} \left[ \ell(X(T), Y(T)) \right] + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 dt \right],
\]

\[\begin{aligned}
\dot{X}(t) &= F(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\
(X(0), Y(0)) &\sim \mu^0,
\end{aligned}\]

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

\[
\partial_t \mu(t) + \text{div}_x(F(t, \theta(t)) \mu(t)) = 0,
\]

and

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

Idea: Learning as linear optimal control on measures!
The **continuous-time** version of the training problem writes

\[
\min_{\theta(\cdot)} \left[ \mathbb{E}_{\mu^0} \left[ \ell(X(T), Y(T)) \right] + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 dt \right],
\]

s.t. \[
\begin{align*}
\dot{X}(t) &= \mathcal{F}(t, \theta(t), X(t)), \\
\dot{Y}(t) &= 0, \\
(X(0), Y(0)) &\sim \mu^0,
\end{align*}
\]

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**

\[
\partial_t \mu(t) + \text{div}_x (\mathcal{F}(t, \theta(t)) \mu(t)) = 0,
\]

and

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

**Idea:** Learning as **linear** optimal control on **measures**!
NeurODEs – *From stochastic to mean-field control*

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

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

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

\[
\partial_t \mu(t) + \text{div}_x (\mathcal{F}(t, \theta(t)) \mu(t)) = 0,
\]

and

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

**Idea:** Learning as linear optimal control on **measures**!
NeurODEs – From stochastic to mean-field control

The continuous-time version of the training problem writes

\[
\begin{aligned}
\min_{\theta(\cdot)} & \quad \mathbb{E}_{\mu^0} \left[ \ell(X(T), Y(T)) \right] + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 dt,
\end{aligned}
\]

\[
\begin{aligned}
\text{s.t.} & \quad \dot{X}(t) = \mathcal{F}(t, \theta(t), X(t)), \quad \dot{Y}(t) = 0, \\
& \quad (X(0), Y(0)) \sim \mu^0,
\end{aligned}
\]

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

\[
\partial_t \mu(t) + \text{div}_x (\mathcal{F}(t, \theta(t)) \mu(t)) = 0,
\]

and

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

Idea: Learning as linear optimal control on measures!
NeurODEs – *From stochastic to mean-field control*

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

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

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

\[
\partial_t \mu(t) + \text{div}_x (\mathcal{F}(t, \theta(t)) \mu(t)) = 0,
\]

and

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

**Idea:** Learning as **linear** optimal control on **measures**!
NeurODEs – From stochastic to mean-field control

The continuous-time version of the training problem writes

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

where \( \theta(\cdot) \) are \textit{controls} and \( \lambda > 0 \) is \textit{regularisation} parameter.

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

\[
\partial_t \mu(t) + \text{div}_x (\mathcal{F}(t, \theta(t)) \mu(t)) = 0,
\]

and

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

**Idea:** Learning as \textit{linear} optimal control on \textit{measures}!
NeurODEs – *Mean-field control formulation of learning*

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

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

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

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)**

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

→ 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)

\[
\min_{\theta(\cdot)} \left[ \int_{\mathcal{X}^2} \ell(x, y) d\mu(T)(x, y) + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 dt \right],
\]

\[
\text{s.t. } \left\{ \begin{array}{l}
\partial_t \mu(t) + \text{div}_x (\mathcal{F}(t, \theta(t)) \mu(t)) = 0, \\
\mu(0) = \mu^0 \in \mathcal{P}(\mathcal{X}^2). 
\end{array} \right.
\]

→ 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)**

\[
\min_{\theta(\cdot)} \left[ \int_{X^2} \ell(x, y) d\mu(T)(x, y) + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 dt \right],
\]

s.t. \[
\begin{cases}
\partial_t \mu(t) + \text{div}_x(F(t, \theta(t))\mu(t)) = 0, \\
\mu(0) = \mu^0 \in \mathcal{P}(X^2).
\end{cases}
\]

→ *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].
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*

\[
\begin{aligned}
\partial_t \mu^*(t) + \mathrm{div}_x(\mathcal{F}(t, \theta^*(t)) \mu^*(t)) &= 0, \quad \mu^*(0) = \mu^0, \\
\partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t, \theta^*(t)) \rangle &= 0, \quad \psi^*(T) = \ell, \\
\theta^*(t) &= -\frac{1}{\lambda} \int_{\mathcal{X}^2} \psi^*(t) \mathcal{D}_\theta \mathcal{F}(t, \theta^*(t))^T \psi^*(t) d\mu^*(t),
\end{aligned}
\]

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** $\leftrightarrow$ **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

$$
\begin{align*}
\partial_t \mu^*(t) + \text{div}_x(\mathcal{F}(t, \theta^*(t))\mu^*(t)) &= 0, \quad \mu^*(0) = \mu^0, \\
\partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), \mathcal{F}(t, \theta^*(t)) \rangle &= 0, \quad \psi^*(T) = \ell, \\
\theta^*(t) &= -\frac{1}{\lambda} \int_{\mathcal{X}^2} D_\theta \mathcal{F}(t, \theta^*(t))^\top \psi^*(t) d\mu^*(t),
\end{align*}
$$

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 $\mapsto$ 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

\[
\begin{align*}
\partial_t \mu^*(t) + \text{div}_x(F(t, \theta^*(t))\mu^*(t)) &= 0, \quad \mu^*(0) = \mu^0, \\
\partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), F(t, \theta^*(t)) \rangle &= 0, \quad \psi^*(T) = \ell, \\
\theta^*(t) &= -\frac{1}{\lambda} \int_{\mathcal{X}^2} D_\theta F(t, \theta^*(t))^\top \psi^*(t) d\mu^*(t),
\end{align*}
\]

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 ensures numerical convergence.
- Efficient methods available to solve each equation.
- Allows to derive quantitative generalisation errors.
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

$$
\begin{align*}
\partial_t \mu^*(t) + \text{div}_x(F(t, \theta^*(t))\mu^*(t)) &= 0, \quad \mu^*(0) = \mu^0, \\
\partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), F(t, \theta^*(t)) \rangle &= 0, \quad \psi^*(T) = \ell, \\
\theta^*(t) &= -\frac{1}{\lambda} \int_{\mathcal{X}^2} D_\theta F(t, \theta^*(t))^\top \psi^*(t) d\mu^*(t),
\end{align*}
$$

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 $\Rightarrow$ 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

$$
\begin{align*}
\partial_t \mu^*(t) + \text{div}_x(F(t, \theta^*(t)))\mu^*(t) &= 0, \quad \mu^*(0) = \mu^0, \\
\partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), F(t, \theta^*(t)) \rangle &= 0, \quad \psi^*(T) = \ell,
\end{align*}
$$

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

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 $\mapsto$ 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

$$\begin{cases}
\partial_t \mu^*(t) + \text{div}_x(F(t, \theta^*(t))\mu^*(t)) = 0, & \mu^*(0) = \mu^0, \\
\partial_t \psi^*(t) + \langle \nabla_x \psi^*(t), F(t, \theta^*(t)) \rangle = 0, & \psi^*(T) = \ell, \\
\theta^*(t) = -\frac{1}{\lambda} \int_{\chi^2} D_\theta F(t, \theta^*(t))^\top \psi^*(t) d\mu^*(t),
\end{cases}$$

where $\psi^* \in C^0([0, T] \times \chi^2, \chi^2)$ 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

\[
\begin{aligned}
\partial_t \mu^*(t) + \text{div}_x (F(t, \theta^*(t))\mu^*(t)) &= 0, \quad \mu^*(0) = \mu^0, \\
\partial_t \psi^*(t) + \left\langle \nabla_x \psi^*(t), F(t, \theta^*(t)) \right\rangle &= 0, \quad \psi^*(T) = \ell, \\
\theta^*(t) &= -\frac{1}{\lambda} \int_{\mathcal{X}^2} \text{D}_\theta F(t, \theta^*(t))^\top \psi^*(t) d\mu^*(t),
\end{aligned}
\]

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 $\iff$ 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

\[ \mathcal{L}(\mu, \psi, \theta) := \int_{\mathcal{X}^2} \ell \, d\mu(T) + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 \, dt \]

\[ + \int_{\mathcal{X}^2} \psi(0)d\mu^0 - \int_{\mathcal{X}^2} \psi(T)d\mu(T) \]

\[ + \int_0^T \int_{\mathcal{X}^2} \left( \partial_t \psi(t) + \langle \nabla_x \psi(t), F(t, \theta(t)) \rangle \right) d\mu(t) dt. \]

2. Abstract **KKT rule** in Banach spaces \( \Rightarrow \) there exists \( \psi^* \) s.t.

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

\( \Rightarrow \) Constraint qualification "requires" continuous controls.

3. Well-posedness by **Schauder’s fixed-point** theorem \( \Rightarrow \) QED!
Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

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

   \[
   L(\mu, \psi, \theta) := \int_{\mathcal{X}} \ell \, d\mu(T) + \frac{\lambda}{2} \int_{0}^{T} |\theta(t)|^2 dt \\
   + \int_{\mathcal{X}} \psi(0) d\mu^0 - \int_{\mathcal{X}} \psi(T) d\mu(T) \\
   + \int_{0}^{T} \int_{\mathcal{X}} \left( \partial_t \psi(t) + \langle \nabla_x \psi(t), F(t, \theta(t)) \rangle \right) d\mu(t) dt.
   \]

2. Abstract **KKT rule** in Banach spaces \( \leadsto \) there exists \( \psi^* \) s.t.

   \[
   \frac{\delta L}{\delta \mu}(\mu^*, \psi^*, \theta^*) = 0 \quad \text{and} \quad \frac{\delta L}{\delta \theta}(\mu^*, \psi^*, \theta^*) = 0.
   \]

   \( \leadsto \) Constraint qualification "requires" continuous controls.

3. Well-posedness by **Schauder’s fixed-point theorem** \( \leadsto \) QED!
Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

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

\[
\mathcal{L}(\mu, \psi, \theta) := \int_{\mathcal{X}} \ell \, d\mu(T) + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 \, dt \\
+ \int_{\mathcal{X}} \psi(0) \, d\mu^0 - \int_{\mathcal{X}} \psi(T) \, d\mu(T) \\
+ \int_0^T \int_{\mathcal{X}} \left( \partial_t \psi(t) + \langle \nabla_x \psi(t), \mathcal{F}(t, \theta(t)) \rangle \right) \, d\mu(t) \, dt.
\]

2. Abstract **KKT rule** in Banach spaces \( \rightsquigarrow \) there exists \( \psi^* \) s.t.

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

\( \rightsquigarrow \) Constraint qualification "requires" continuous controls.

3. Well-posedness by **Schauder’s fixed-point theorem** \( \rightsquigarrow \text{QED!} \)
Proof of the optimality conditions – *Lagrangian approach*

Proof of the optimality conditions (Lagrangian heuristic)

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

\[ \mathcal{L}(\mu, \psi, \theta) := \int_{\mathcal{X}^2} \ell \, d\mu(T) + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 dt \]
\[ + \int_{\mathcal{X}^2} \psi(0)d\mu^0 - \int_{\mathcal{X}^2} \psi(T)d\mu(T) \]
\[ + \int_0^T \int_{\mathcal{X}^2} \left( \partial_t \psi(t) + \langle \nabla_x \psi(t), \mathcal{F}(t, \theta(t)) \rangle \right) d\mu(t) dt. \]

2. Abstract **KKT rule** in **Banach spaces** \( \rightsquigarrow \) there exists \( \psi^* \) s.t.

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

\( \leftrightarrow \) **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

\[
\mathcal{L}(\mu, \psi, \theta) := \int_{\mathcal{X}}^2 \ell \, d\mu(T) + \frac{\lambda}{2} \int_0^T |\theta(t)|^2 \, dt \\
+ \int_{\mathcal{X}}^2 \psi(0) d\mu^0 - \int_{\mathcal{X}}^2 \psi(T) d\mu(T) \\
+ \int_0^T \int_{\mathcal{X}}^2 \left( \partial_t \psi(t) + \langle \nabla_x \psi(t), F(t, \theta(t)) \rangle \right) d\mu(t) dt.
\]

2. Abstract **KKT rule** in Banach spaces \( \rightsquigarrow \) there exists \( \psi^* \) s.t.

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

\( \rightsquigarrow \) **Constraint qualification** “requires” **continuous** controls.

3. Well-posedness by **Schauder’s fixed-point theorem** \( \rightsquigarrow \) **QED!**
Proof of the optimality conditions – \textit{Lagrangian approach}

Proof of the optimality conditions (Lagrangian heuristic)

1. Define the \textbf{Lagrangian} of the problem by

\[
\mathcal{L}(\mu, \psi, \theta) := \int_{\mathcal{X}^2} \ell \, d\mu(T) + \frac{\lambda}{2} \int_{0}^{T} |\theta(t)|^2 dt \\
+ \int_{\mathcal{X}^2} \psi(0)d\mu^0 - \int_{\mathcal{X}^2} \psi(T)d\mu(T) \\
+ \int_{0}^{T} \int_{\mathcal{X}^2} \left( \partial_t \psi(t) + \langle \nabla_x \psi(t), F(t, \theta(t)) \rangle \right) d\mu(t) dt.
\]

2. Abstract \textbf{KKT rule} in Banach spaces \implies there exists \( \psi^* \) s.t.

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

\implies \textbf{Constraint qualification} “requires” \textbf{continuous} controls.

3. Well-posedness by \textbf{Schauder’s fixed-point} theorem \implies \textbf{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.

\[
\begin{cases}
\partial_t \sigma^*(t) = -D_x F(t, \theta^*(t))^\top \sigma^*(t), & \sigma^*(T) = -\nabla_x \ell, \\
\theta^*(t) \in \arg\max_{\theta \in \mathbb{R}^m} \left[ \int_{\mathcal{X}^2} \langle \sigma^*(t), F(t, \theta) \rangle d\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right].
\end{cases}
\]

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

\[
\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} D_\theta F(t, \theta^*(t))^\top \sigma^*(t) d\mu^*(t).
\]

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

**Question (Link between both approaches)**

We have **Lagrangian $\subset$ Hamiltonian $\leadsto$ 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.
   \[
   \begin{cases}
   \partial_t \sigma^*(t) = -D_x \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t), \\
   \sigma^*(T) = -\nabla_x \ell,
   \end{cases}
   \]
   \[
   \theta^*(t) \in \arg\max_{\theta \in \mathbb{R}^m} \left[ \int_{\chi^2} \langle \sigma^*(t), \mathcal{F}(t, \theta) \rangle d\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right].
   \]

2. Because \(\lambda > 0\) is large \(\Rightarrow\) unique maximiser \(\theta^*(t)\) satisfying
   \[
   \theta^*(t) = \frac{1}{\lambda} \int_{\chi^2} D_\theta \mathcal{F}(t, \theta^*(t))^\top \sigma^*(t) d\mu^*(t).
   \]

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

**Question (Link between both approaches)**

We have **Lagrangian \(\subset\) Hamiltonian** \(\sim\) **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.

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

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

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

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

Question (Link between both approaches)
We have Lagrangian $\subset$ Hamiltonian $\implies$ 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.

\[
\begin{align*}
\partial_t \sigma^*(t) &= -D_x F(t, \theta^*(t))^\top \sigma^*(t), \\
\sigma^*(T) &= -\nabla_x \ell, \\
\end{align*}
\]

\[
\theta^*(t) \in \arg\max_{\theta \in \mathbb{R}^m} \left[ \int_{\mathcal{X}^2} \langle \sigma^*(t), F(t, \theta) \rangle d\mu^*(t) - \frac{\lambda}{2} |\theta|^2 \right].
\]

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

\[
\theta^*(t) = \frac{1}{\lambda} \int_{\mathcal{X}^2} D_\theta F(t, \theta^*(t))^\top \sigma^*(t) d\mu^*(t).
\]

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

**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.
   \[
   \begin{cases}
   \partial_t \sigma^*(t) = -D_x F(t, \theta^*(t))^\top \sigma^*(t), & \sigma^*(T) = -\nabla_x \ell, \\
   \theta^*(t) \in \arg\max_{\theta \in \mathbb{R}^m} \left[ \int_{\chi^2} \langle \sigma^*(t), F(t, \theta) \rangle d\mu^*(t) - \frac{\lambda}{2} |\theta|_2^2 \right].
   \end{cases}
   \]

2. Because \( \lambda > 0 \) is large \( \leadsto \) unique maximiser \( \theta^*(t) \) satisfying
   \[
   \theta^*(t) = \frac{1}{\lambda} \int_{\chi^2} D_\theta F(t, \theta^*(t))^\top \sigma^*(t) d\mu^*(t).
   \]

3. Cauchy-Lip uniqueness \( \Rightarrow \sigma^*(t) = -\nabla_x \psi^*(t) \leadsto \text{QED}! \)

Question (Link between both approaches)
We have \textbf{Lagrangian} \( \subset \) \textbf{Hamiltonian} \( \leadsto \) \text{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.

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

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

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

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

**Question (Link between both approaches)**

We have **Lagrangian $\subset$ Hamiltonian $\leadsto$ Equivalence?**
Outline of the talk

Quick primer on neural networks

NeurODE models and mean-field control

Optimality conditions: Lagrangian and Hamiltonian approaches

Numerical illustrations
Numerical illustrations – *Algorithmic schemes*

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

**Algorithm (General framework)**

Fix initial layers $\theta^0$ and for $k = 1 \ldots K_{\text{max}}$

1. Solve simultaneously the forward-backward equations

   \[
   \begin{aligned}
   \partial_t \mu_k(t) + \text{div}_x(F(t, \theta_k(t))\mu_k(t)) &= 0, & \mu_k(0) &= \mu^0, \\
   \partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), F(t, \theta_k(t)) \rangle &= 0, & \psi_k(T) &= \ell.
   \end{aligned}
   \]

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

2. Update the layers by solving

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

   $\rightarrow$ **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 $\theta^0$ and for $k = 1 \ldots K_{\text{max}}$

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

\[
\begin{align*}
\partial_t \mu_k(t) + \text{div}_x(F(t, \theta_k(t))\mu_k(t)) &= 0, \quad \mu_k(0) = \mu^0, \\
\partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), F(t, \theta_k(t)) \rangle &= 0, \quad \psi_k(T) = \ell.
\end{align*}
\]

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

2. Update the **layers** by solving

\[
\theta_{k+1}(t) + \frac{1}{\lambda} \int_{\chi^2} D_\theta F(t, \theta_{k+1}(t))^\top \nabla_x \psi_k(t) d\mu_k(t) = 0.
\]

$\leftrightarrow$ **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 $\theta^0$ and for $k = 1 \ldots K_{\text{max}}$

1. Solve simultaneously the forward-backward equations

\[
\begin{align*}
\partial_t \mu_k(t) + \text{div}_x(F(t, \theta_k(t))\mu_k(t)) &= 0, \quad \mu_k(0) = \mu^0, \\
\partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), F(t, \theta_k(t)) \rangle &= 0, \quad \psi_k(T) = \ell.
\end{align*}
\]

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

2. Update the layers by solving

\[
\theta_{k+1}(t) + \frac{1}{\lambda} \int_{\chi^2} D_{\theta} F(t, \theta_{k+1}(t))^\top \nabla_x \psi_k(t) d\mu_k(t) = 0.
\]

$\rightharpoonup$ **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 \( \theta^0 \) and for \( k = 1 \ldots K_{\text{max}} \)

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

\[
\begin{aligned}
\partial_t \mu_k(t) + \text{div}_x(\mathcal{F}(t, \theta_k(t))\mu_k(t)) &= 0, \quad \mu_k(0) = \mu^0, \\
\partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), \mathcal{F}(t, \theta_k(t)) \rangle &= 0, \quad \psi_k(T) = \ell.
\end{aligned}
\]

\(\hookrightarrow\) **Particle approximation** or **semi-Lagrangian scheme**.

2. Update the **layers** by solving

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

\(\hookrightarrow\) **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 $\theta^0$ and for $k = 1 \ldots K_{\text{max}}$

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

$$
\begin{align*}
\partial_t \mu_k(t) + \text{div}_x(\mathcal{F}(t, \theta_k(t))\mu_k(t)) &= 0, \quad \mu_k(0) = \mu^0, \\
\partial_t \psi_k(t) + \langle \nabla_x \psi_k(t), \mathcal{F}(t, \theta_k(t)) \rangle &= 0, \quad \psi_k(T) = l.
\end{align*}
$$

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

2. Update the **layers** by solving

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

$\hookrightarrow$ **Particle approximation** of the integral and **Newton**.
Numerical illustrations – *Toy example*

Example (Binary classification of points on the real line)

Let $\mu_0^x = \mathcal{N}(0, 1)$, $X_i^0 \sim \mu^0$ for $i \in \{1, \ldots, N\}$, and find $\theta^*(\cdot)$ s.t.

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

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 $\mu_0 = \mathcal{N}(0, 1)$, $X^0_i \sim \mu_0$ for $i \in \{1, \ldots, N\}$, and find $\theta^*(\cdot)$ s.t.

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

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

Example (Binary classification of points on the real line)

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

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

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

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

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 $\mu_0^0 = \mathcal{N}(0, 1)$, $X_i^0 \sim \mu^0$ for $i \in \{1, \ldots, N\}$, and find $\theta^*(\cdot)$ s.t.

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

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 $\leadsto$ **mathematically rich**
2. Learning problem $\leadsto$ **linear** optimal control on **measures**.

Thank you for your attention!