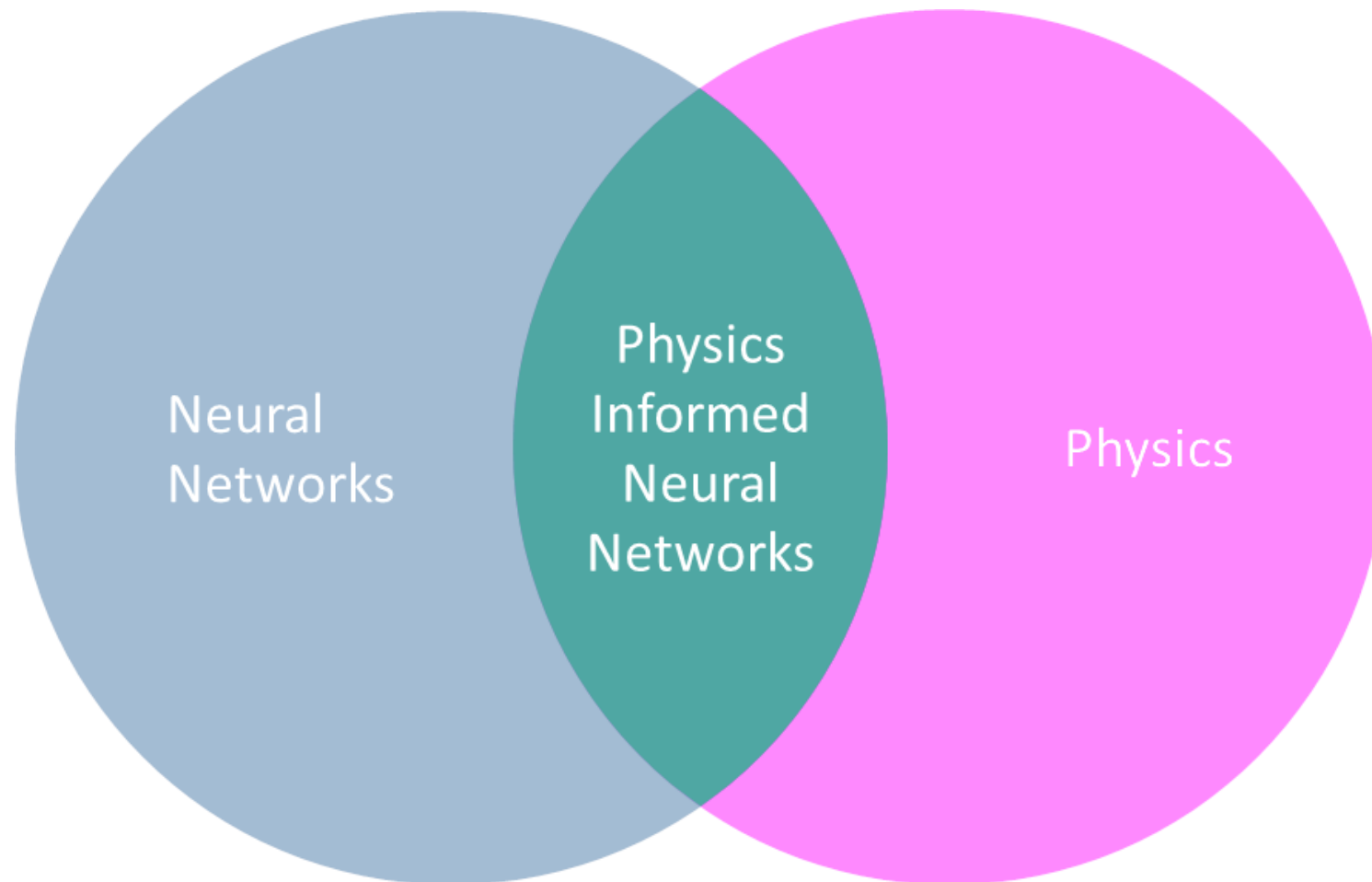


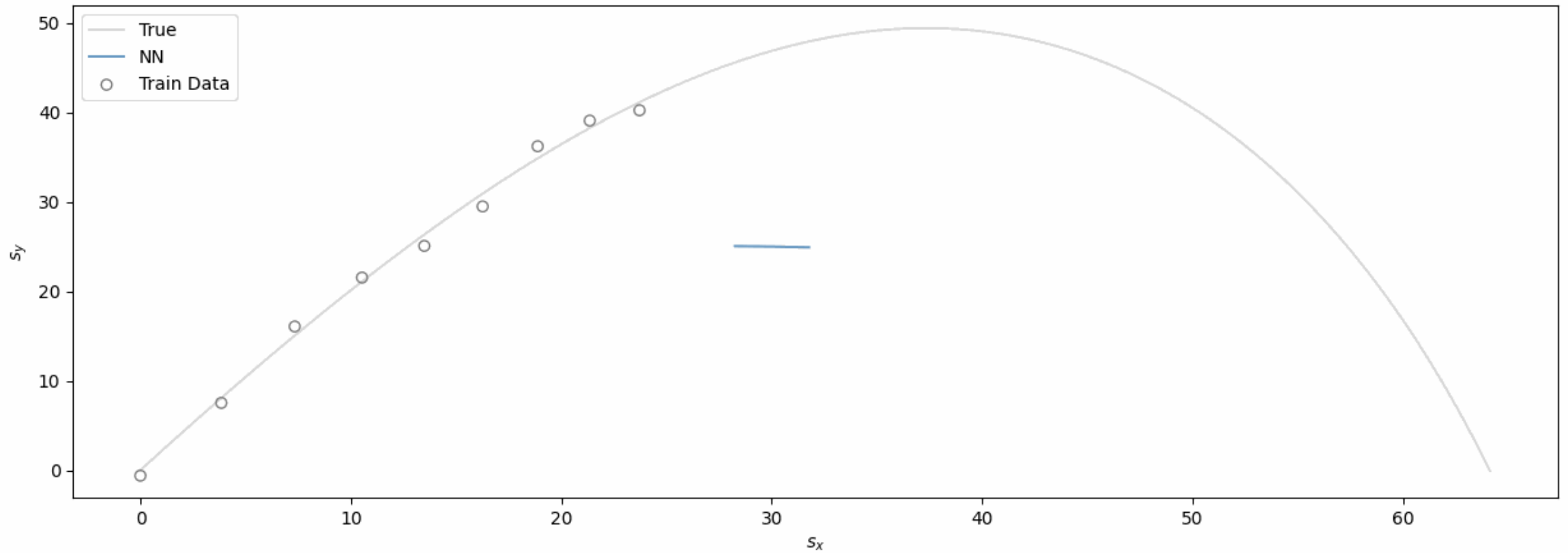
Physics-Informed Neural Networks



Physics-Informed Neural Networks

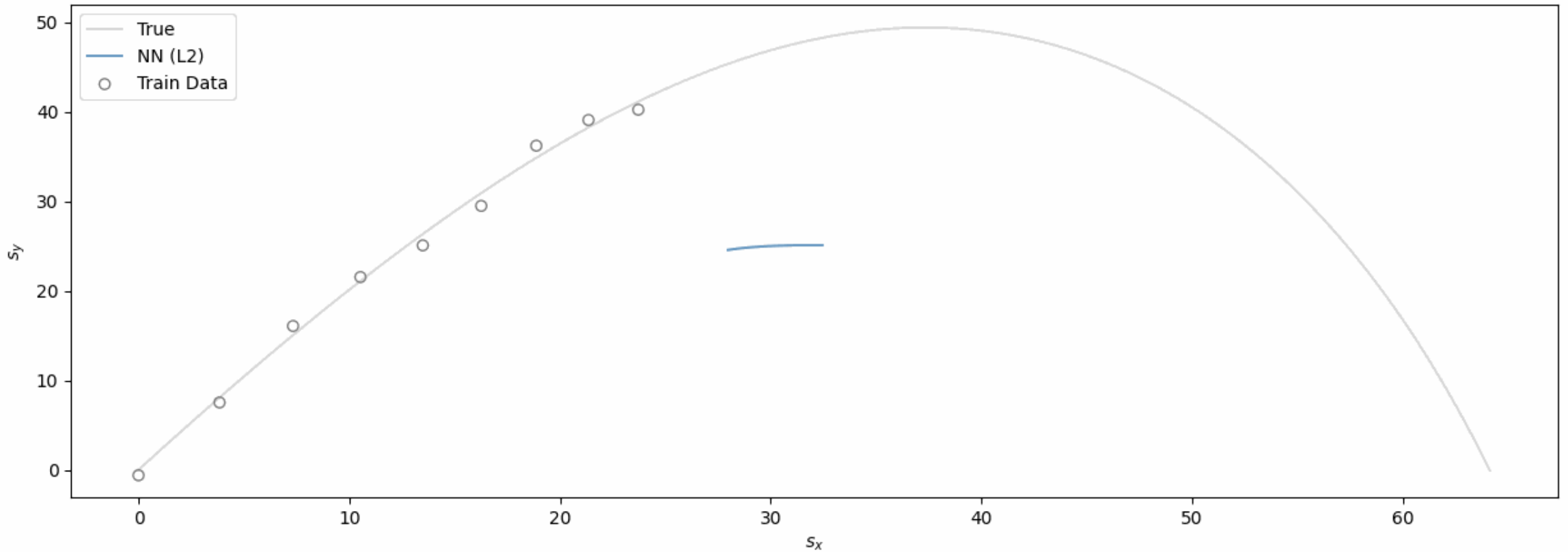


Neural Network



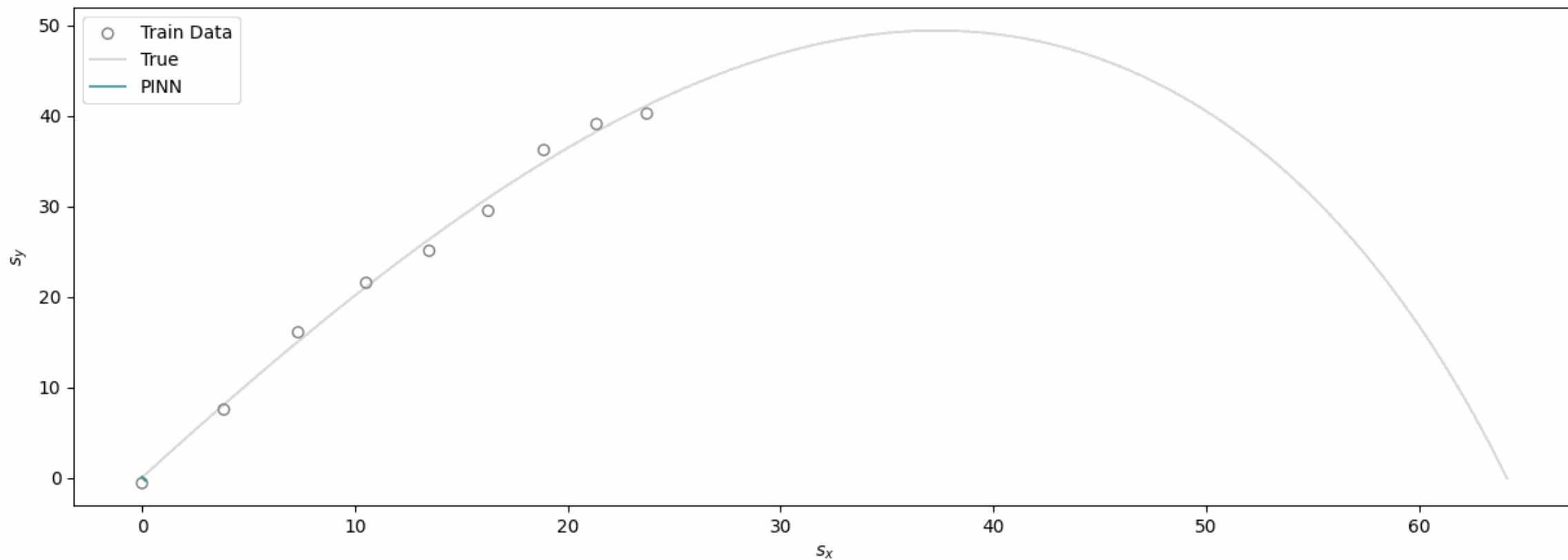
over-fit

Neural Network with Regularization

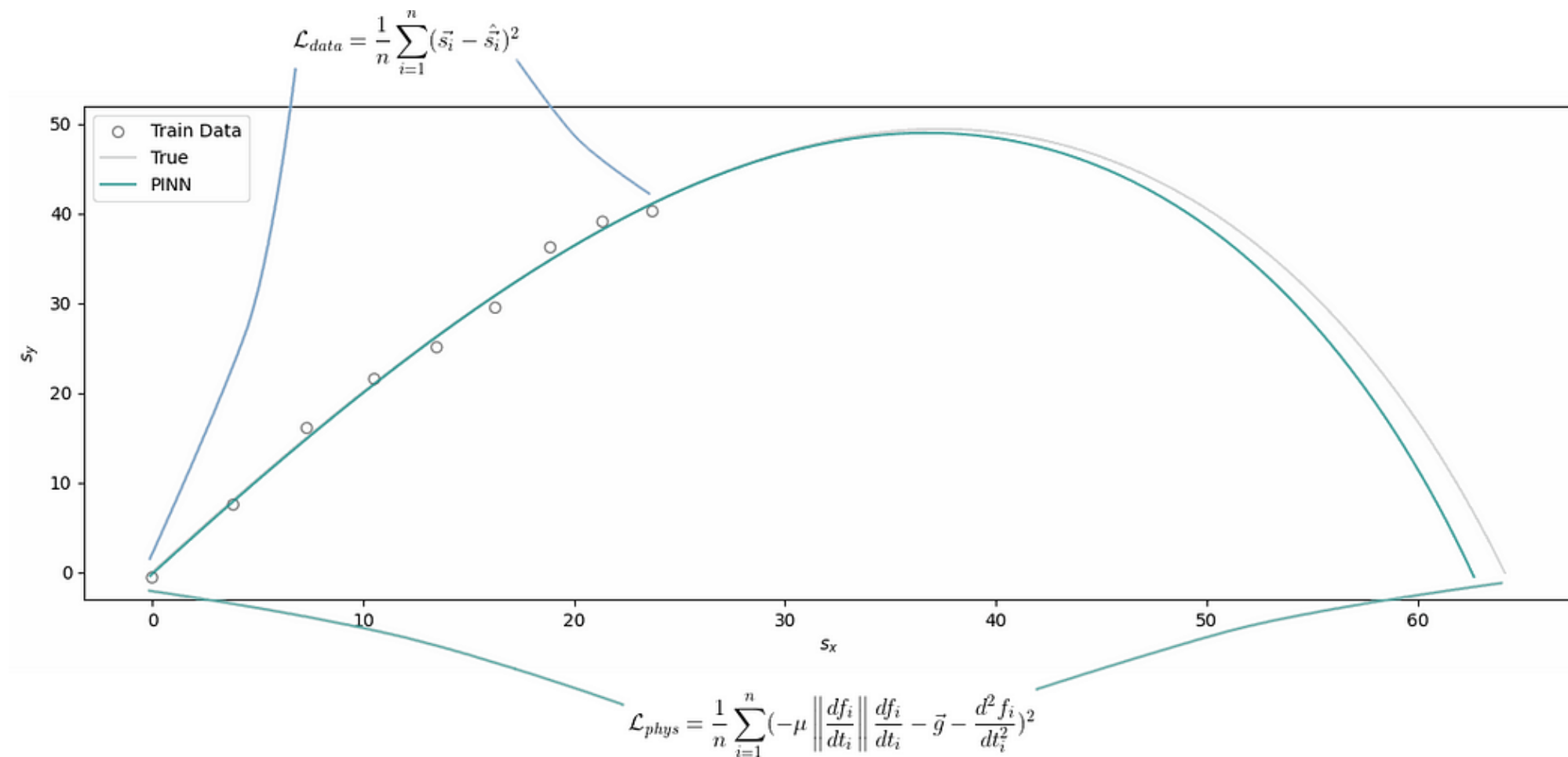


not able to extrapolate across the domain

Physics-Informed Neural Networks



Physics-Informed Neural Networks



物理知識導向神經網絡 (PINNs)

PINNs 將偏微分方程 (Partial Differential Equations, PDEs) 的數學結構和邊界、初始條件嵌入神經網絡的損失函數中，適用於數值模擬、參數辨識以及反問題求解。

數學模型

考慮以下描述物理系統的偏微分方程 (PDE) 與其對應的邊界及初始條件：

$$\begin{aligned}\mathcal{N}[u(x, t)] &= f(x, t), & x \in \Omega, t \in [0, T], \\ u(x, t) &= g(x, t), & x \in \partial\Omega, t \in [0, T], \quad (\text{邊界條件}) \\ u(x, 0) &= h(x), & x \in \Omega. \quad (\text{初始條件})\end{aligned}$$

- \mathcal{N} : 描述系統的微分算子（可能包括時間、空間的一階或高階導數）。
- $f(x, t)$: 系統的外部源項。
- $g(x, t)$: 定義的邊界條件。
- $h(x)$: 初始條件。

神經網絡模型

神經網絡 $u_\theta(x, t)$ 的輸出是 u 的近似解，透過損失函數的優化滿足 PDE 和其邊界/初始條件。

損失函數

PINNs 的損失函數 $\mathcal{L}(\theta)$ 包含三部分：

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{physics}} + \mathcal{L}_{\text{boundary}} + \mathcal{L}_{\text{initial}},$$

1. 物理損失 $\mathcal{L}_{\text{physics}}$ ：

針對 PDE 的約束：

$$\mathcal{L}_{\text{physics}} = \frac{1}{N_r} \sum_{j=1}^{N_r} |\mathcal{N}[u_\theta(x_j, t_j)] - f(x_j, t_j)|^2,$$

其中 $\{x_j, t_j\}_{j=1}^{N_r}$ 是 PDE 的離散點集，透過自動微分計算微分項。

2. 邊界條件損失 $\mathcal{L}_{\text{boundary}}$:

針對邊界條件 $u(x, t) = g(x, t)$:

$$\mathcal{L}_{\text{boundary}} = \frac{1}{N_b} \sum_{k=1}^{N_b} |u_{\theta}(x_k, t_k) - g(x_k, t_k)|^2,$$

其中 $\{x_k, t_k\}_{k=1}^{N_b}$ 是邊界點集。

3. 初始條件損失 $\mathcal{L}_{\text{initial}}$:

針對初始條件 $u(x, 0) = h(x)$:

$$\mathcal{L}_{\text{initial}} = \frac{1}{N_i} \sum_{m=1}^{N_i} |u_{\theta}(x_m, 0) - h(x_m)|^2,$$

其中 $\{x_m\}_{m=1}^{N_i}$ 是初始條件的離散點集。

最終優化目標

透過調整神經網絡參數 θ ，最小化損失函數：

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

這使得網絡解 $u_{\theta}(x, t)$ 滿足物理方程、邊界條件及初始條件。

特點

1. 強化物理一致性：避免僅依賴數據驅動的過擬合。
2. 自動微分技術：高效計算導數，無需明確離散化。
3. 適應性強：適用於高維問題或複雜幾何。

應用案例：熱傳導問題

考慮一維熱傳導方程 (Heat Equation)：

$$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0, \quad x \in [0, 1], \quad t \in [0, T],$$

其中：

- $u(x, t)$ 是溫度分佈。
- α 是熱擴散係數。

邊界條件與初始條件

1. 邊界條件：

$$u(0, t) = 0, \quad u(1, t) = 0, \quad t \in [0, T].$$

2. 初始條件：

$$u(x, 0) = \sin(\pi x), \quad x \in [0, 1].$$

PINNs 的實現步驟

1. 設計一個神經網絡 $u_{\theta}(x, t)$ 來近似 $u(x, t)$ 。
2. 定義損失函數，包括：
 - PDE 損失：滿足熱傳導方程。
 - 邊界條件損失：滿足 $u(0, t) = 0$ 和 $u(1, t) = 0$ 。
 - 初始條件損失：滿足 $u(x, 0) = \sin(\pi x)$ 。
3. 使用自動微分技術計算偏導數，並優化神經網絡參數 θ 。

1. 熱傳導問題的物理域與離散點

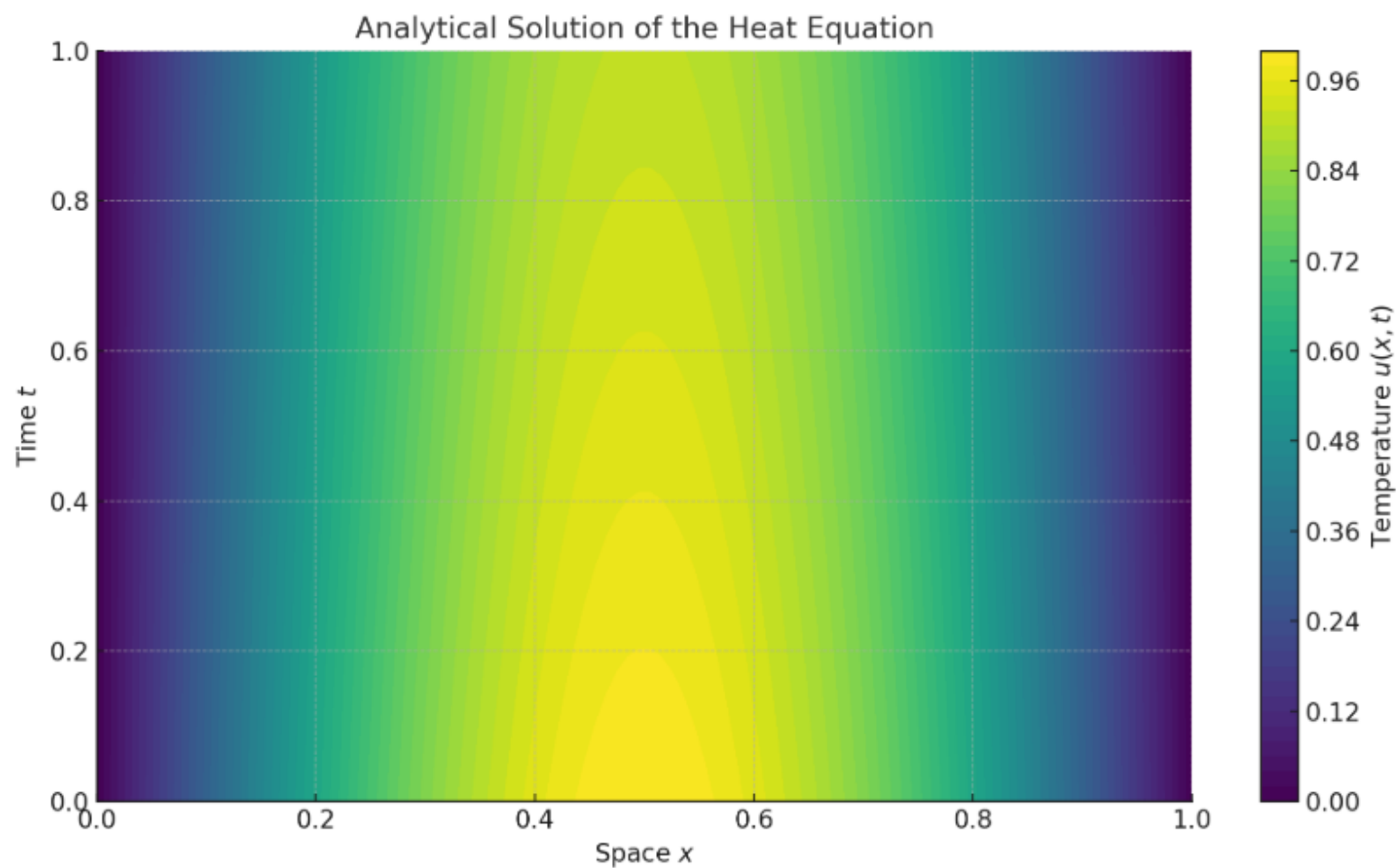
- 在 $x \in [0, 1]$ 和 $t \in [0, T]$ 上均勻取樣點，包括：
 - PDE 節點 (x_j, t_j) 。
 - 邊界節點 $(0, t_k)$ 和 $(1, t_k)$ 。
 - 初始條件節點 $(x_m, 0)$ 。

2. 損失函數收斂圖

- 展示損失函數隨訓練迭代次數的下降過程。

3. 神經網絡的預測結果

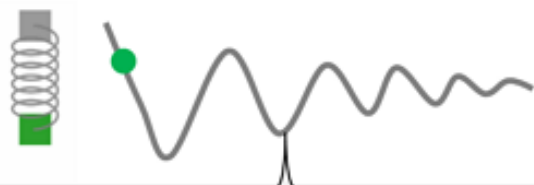
- 在空間和時間域中展示 $u_\theta(x, t)$ 與解析解 $u(x, t) = e^{-\pi^2 \alpha t} \sin(\pi x)$ 的對比圖。



上圖展示了一維熱傳導方程的解析解 $u(x, t)$ 在空間 $x \in [0, 1]$ 和時間 $t \in [0, T]$ 中的分佈情況。溫度隨時間逐漸衰減，並符合初始條件與邊界條件。

Physics-Informed Neural Networks

Real world :
簡諧運動



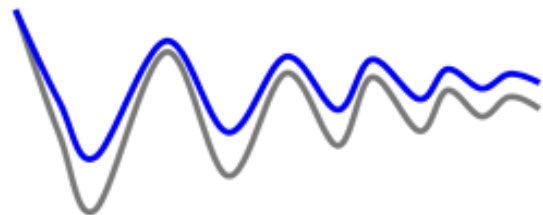
理論模型(偏微分方程式)

目的：描述 t 時間點的位置，即 $y(t)$

數據模型

$$m \frac{d^2 y(t)}{dt^2} + \mu \frac{dy(t)}{dt} + ky(t) = 0$$

where m is the mass of the oscillator, μ is the coefficient of friction and k is the spring constant.

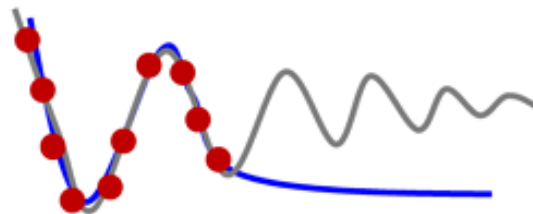


理論模型的限制：

- 高度依賴領域知識
- 複雜系統不容易完整描述
- 不易即時反應現況，預測有偏差

$$y(t) = f(b + \sum_{i=1}^n t_i w_i)$$

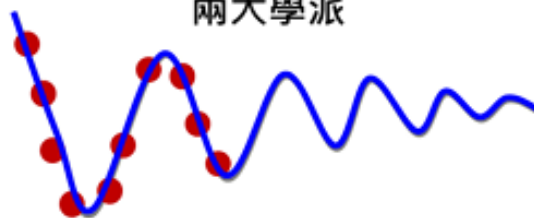
where w is the weight, b is bias.



數據模型的限制：

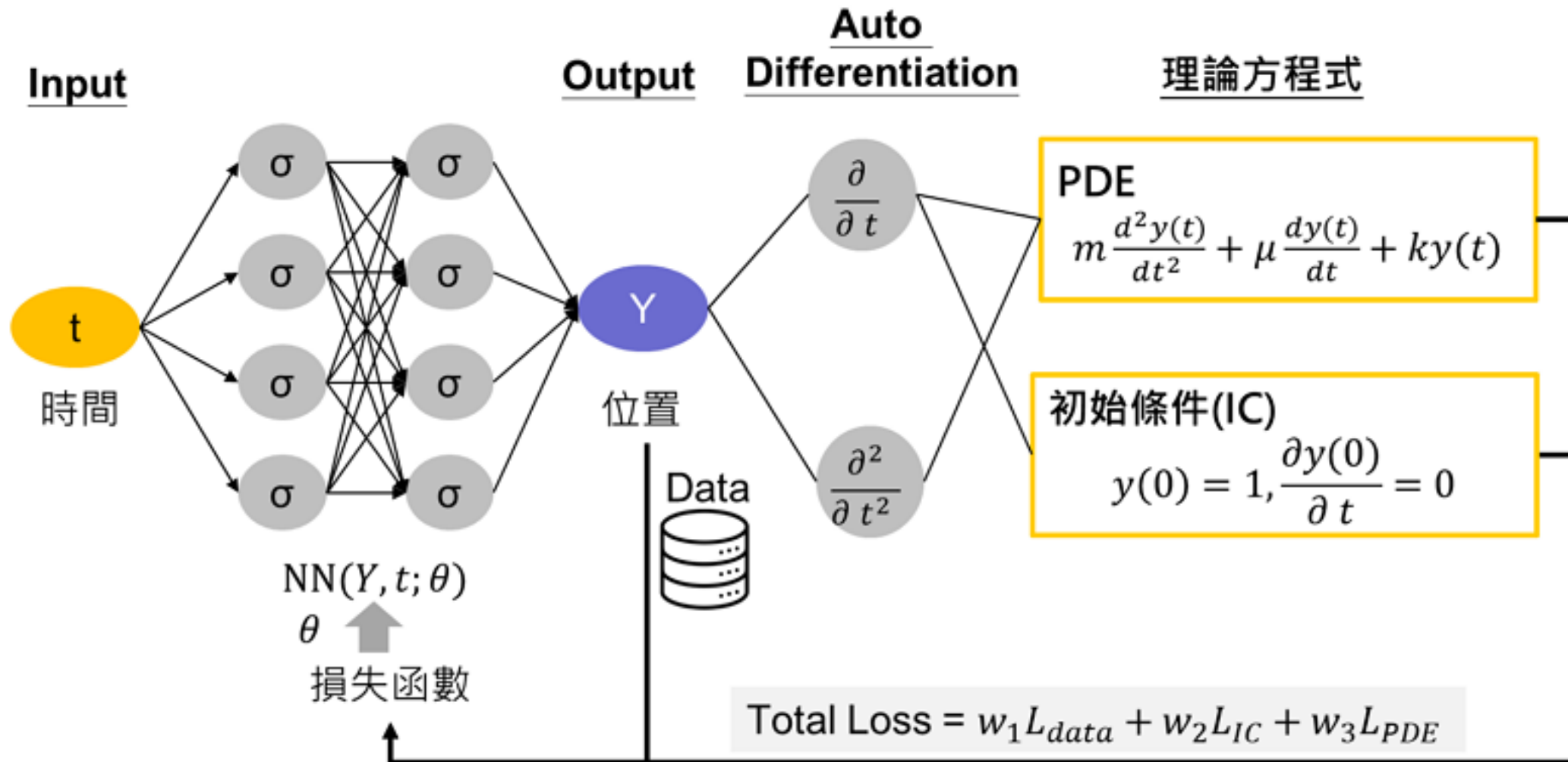
- 需收集大量資料
- 沒有可解釋性
- 不容易處理外插

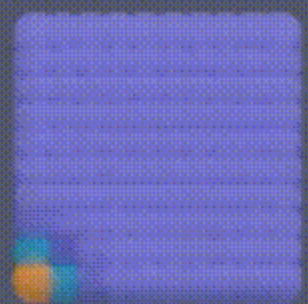
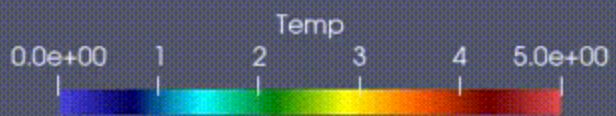
PINN可串聯學理基礎與AI
兩大學派



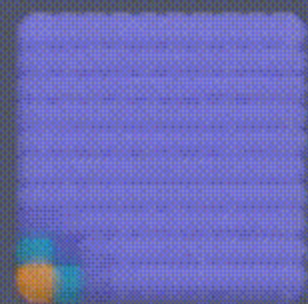
理論知識+資料學習
=有限資料下的精準建模

Physics-Informed Neural Networks





Physics-informed
Neural Network



$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$



模型類型	優勢	劣勢
數據模型	<p>相較於理論模型：</p> <ul style="list-style-type: none"> 可由資料，透過數據模型自動學習輸入與輸出的關聯性 	<p>相較於理論模型：</p> <ul style="list-style-type: none"> 仰賴大量資料，且預測範圍局限於訓練資料集 模型解釋性較差
理論模型	<p>相較於數據模型：</p> <ul style="list-style-type: none"> 理論方程式具有高度解釋性 求解理論方程式無須資料或僅需要極少量的資料 	<p>相較於數據模型：</p> <ul style="list-style-type: none"> 僅用理論方程式較難即時反應真實場域狀況 複雜理論方程式的數值方法求解效率低（耗時） 若理論方程式中帶有未知係數（方程式求解之反向問題），會提高數值方法求解難度
PINN 模型	<p>相較於數據模型：</p> <ul style="list-style-type: none"> 小量資料建模，且提升無資料參數空間之預測準確度（外插預測） 模型具有解釋性 <p>相較於理論模型：</p> <ul style="list-style-type: none"> 可結合場域資料，即時反應製程現況 提升預測精確度 透過高效率運算，求解複雜理論方程式 可同時估計模型參數與方程式係數 	<p>相較於數據模型：</p> <ul style="list-style-type: none"> 整合實際場域資料與物理方程式，因此模型訓練複雜度提升 <p>相較於理論模型：</p> <ul style="list-style-type: none"> 必須蒐集少量真實場域資料 僅能處理特定物理方程式（目前 PINN 主要是處理微分方程式為主）

$$\nabla^2 T = \nabla \cdot (\nabla T)$$

散度 梯度
(守恒)(變化)

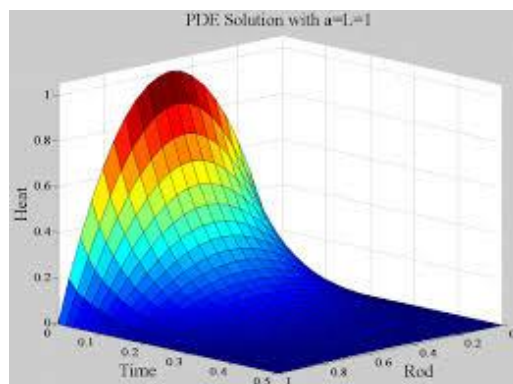


Partial Differential Equation

Laplace
Equation

$$\Delta^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$$

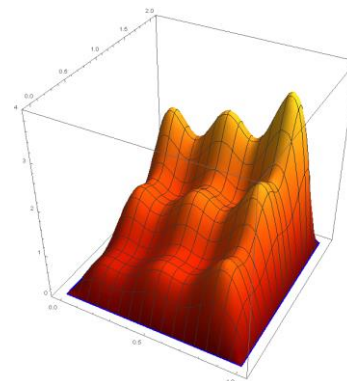
空間平衡狀態：平滑



Heat Conduction
Equation

$$\frac{\partial T}{\partial t} = c \frac{\partial^2 T}{\partial x^2}$$

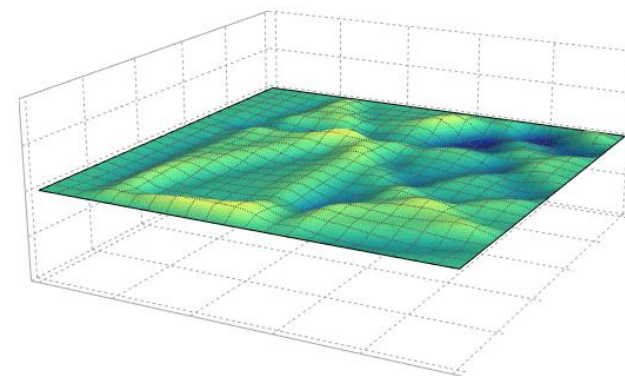
時間動態變化



Wave Equation of
Vibrating Membrane

$$\frac{\partial^2 u}{\partial t^2} = c \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

時間二階導數：質量效應(加速度)
空間二階導數：彈性效應(恢復力)



Q：為什麼空間導數是二階(拉普拉斯算子)？

散度是矢量場的一種運算，衡量某點處矢量場的**「發散程度」**，即該點是否是源或匯：

$$\nabla \cdot \mathbf{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$

- $\mathbf{F} = (F_x, F_y, F_z)$ 是一個矢量場。

意義：

- 如果 $\nabla \cdot \mathbf{F} > 0$ ：表明該點是源頭，表示有量向外流出。
- 如果 $\nabla \cdot \mathbf{F} < 0$ ：表明該點是匯，表示有量向內流入。
- 如果 $\nabla \cdot \mathbf{F} = 0$ ：該點附近無淨流出或流入。

守恒定律描述某物理量（如質量、動量、能量）的總量隨時間的變化情況。以控制體積（Control Volume）為基礎，守恒定律可以表示為：

$$\text{總變化率} = \text{流入率} - \text{流出率} + \text{內部生成}$$

局部形式：

利用散度定理（Gauss's Divergence Theorem），控制體積內的淨流率可以轉化為散度形式：

$$\int_V \frac{\partial u}{\partial t} dV = - \int_V \nabla \cdot \mathbf{F} dV + \int_V S dV$$

- u ：體積內的物理量密度（如濃度、內能密度）。
- \mathbf{F} ：物理量的通量（如熱流密度、質量流密度）。
- S ：內部生成項。

當這個關係對任意小體積都成立時，可以導出偏微分形式：

$$\frac{\partial u}{\partial t} = -\nabla \cdot \mathbf{F} + S$$

傅立葉定律描述熱流密度 \mathbf{q} 與溫度梯度的關係：

$$\mathbf{q} = -k\nabla T$$

負號：高溫流向低溫

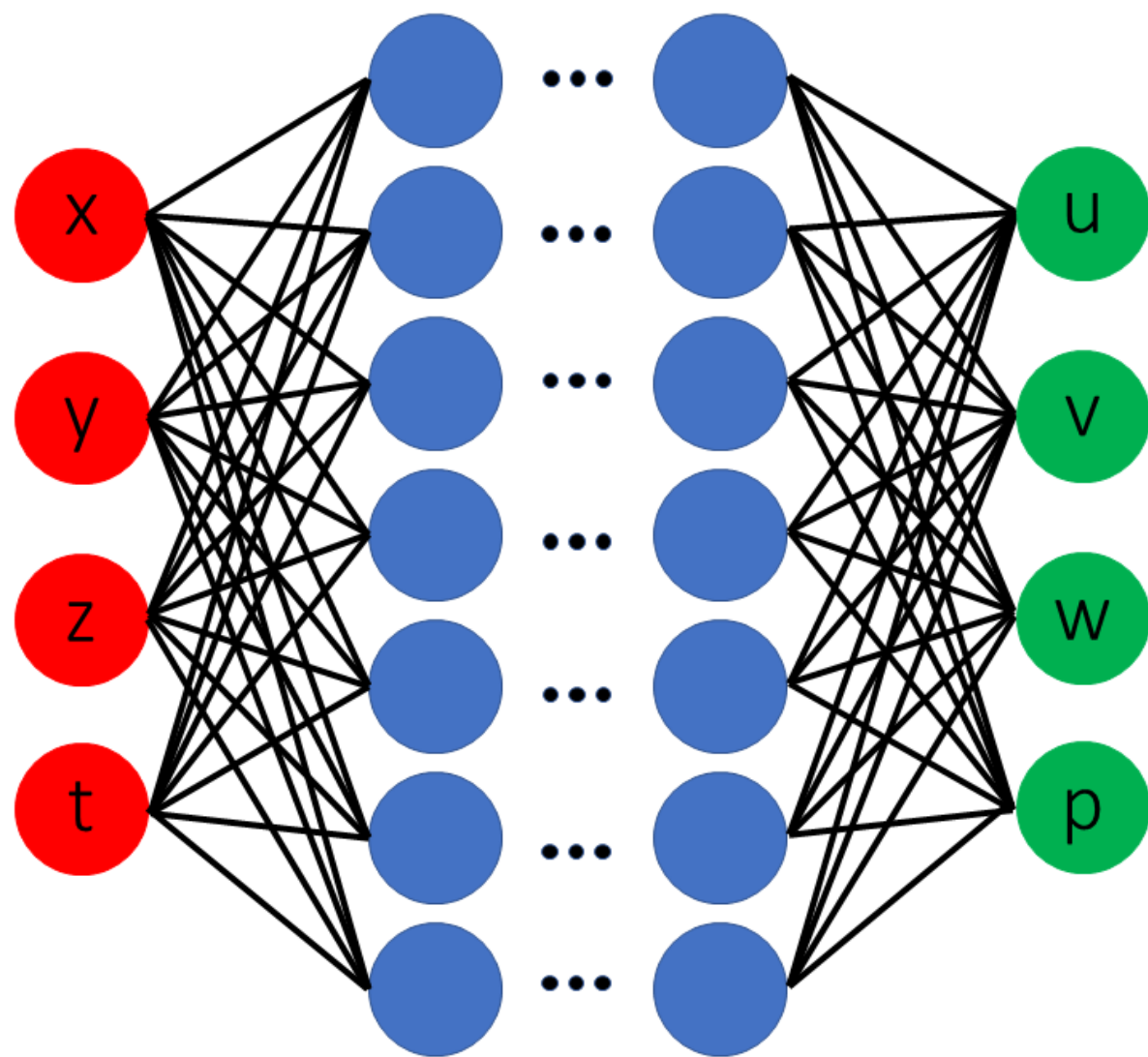
將其代入熱量守恆方程：

$$\frac{\partial u}{\partial t} = -\nabla \cdot \mathbf{q}$$

因為內能密度 u 與溫度 T 有關： $u = \rho c_p T$ ，最終得熱傳導方程：

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T$$

- 散度 $\nabla \cdot \mathbf{q}$ 表示局部熱流的淨流出或流入。
- 拉普拉斯算子 $\nabla^2 T$ 是溫度梯度的梯度，進一步反映熱量的擴散趨勢。



Naviers-Stokes loss

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

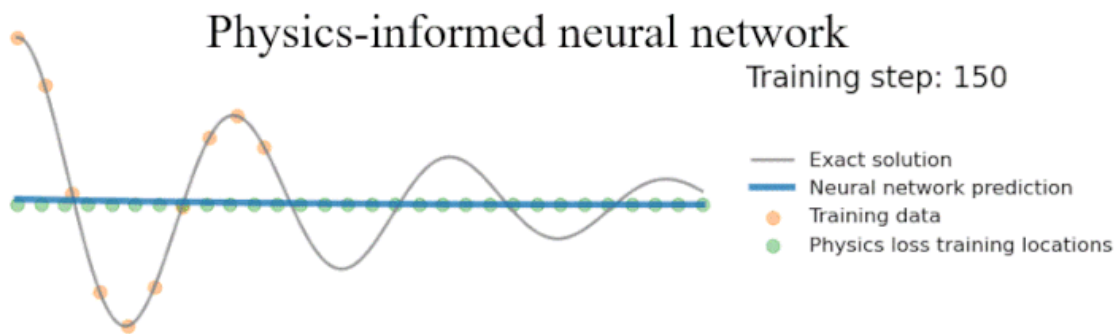
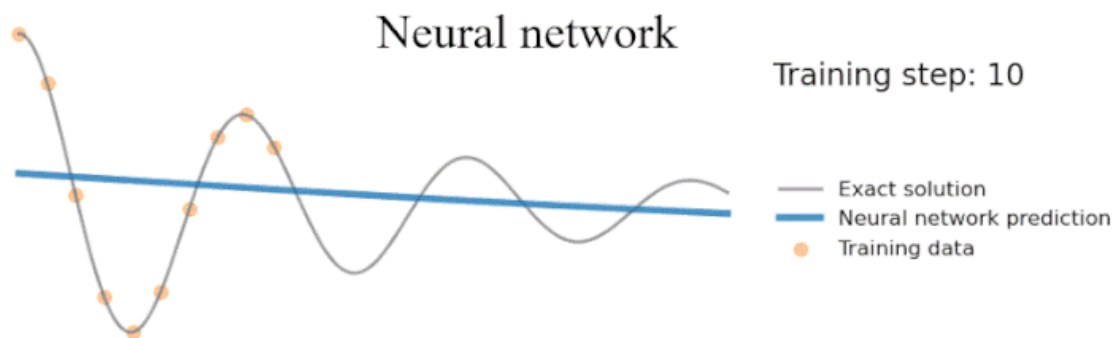
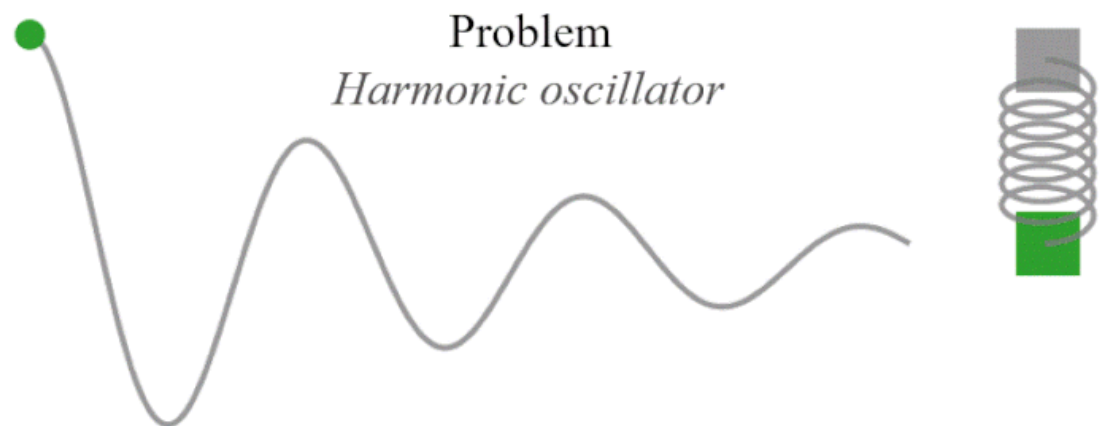
$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} + \frac{\partial P}{\partial x} - \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) = 0$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + \frac{\partial P}{\partial y} - \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right) = 0$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} + \frac{\partial P}{\partial z} - \frac{1}{Re} \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) = 0$$

Experimental data loss

$$\|\vec{V} - \vec{V}_{exp}\|^2 = 0$$



MeshGraphNet 簡介

MeshGraphNet 是 DeepMind 提出的基於圖神經網路 (Graph Neural Network, GNN) 的物理模擬模型。該模型使用圖結構來表示模擬域，其中：

- 節點 (Nodes) 表示物理網格中的離散點 (如網格點或粒子) 。
- 邊 (Edges) 表示節點間的相互關係 (如鄰接點之間的連結) 。
- 特徵向量包括位置、速度、壓力、應力等物理量。

MeshGraphNet 通過 GNN 模擬這些物理特徵的動態演化，基於訓練數據學習物理規律。

MeshGraphNet 的核心數學表達

1. 圖結構定義

MeshGraphNet 的輸入是一個圖 $G = (V, E, \mathbf{X})$ ，其中：

- $V = \{v_1, v_2, \dots, v_N\}$ ：節點集合。
- $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$ ：邊集合。
- \mathbf{X} ：節點及邊的特徵矩陣。

2. 消息傳遞 (Message Passing)

MeshGraphNet 使用 GNN 的消息傳遞機制來更新節點和邊的狀態：

$$\mathbf{h}_i^{(l+1)} = \phi_v \left(\mathbf{h}_i^{(l)}, \sum_{j \in \mathcal{N}(i)} \phi_e \left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij} \right) \right)$$

- $\mathbf{h}_i^{(l)}$ ：第 l 層節點 v_i 的特徵。
- ϕ_v 和 ϕ_e ：分別是節點和邊的更新函數（由神經網路參數化）。
- \mathbf{e}_{ij} ：邊的特徵。
- $\mathcal{N}(i)$ ：節點 v_i 的鄰居集合。

3. 全域更新 (Global Update)

MeshGraphNet 支援整體特徵的全域更新：

$$\mathbf{g}^{(l+1)} = \phi_g \left(\mathbf{g}^{(l)}, \sum_i \phi_v^{\text{global}}(\mathbf{h}_i^{(l)}) \right)$$

- $\mathbf{g}^{(l)}$ ：全域特徵向量。

4. 時間步進 (Temporal Evolution)

模型預測節點狀態的時間演化：

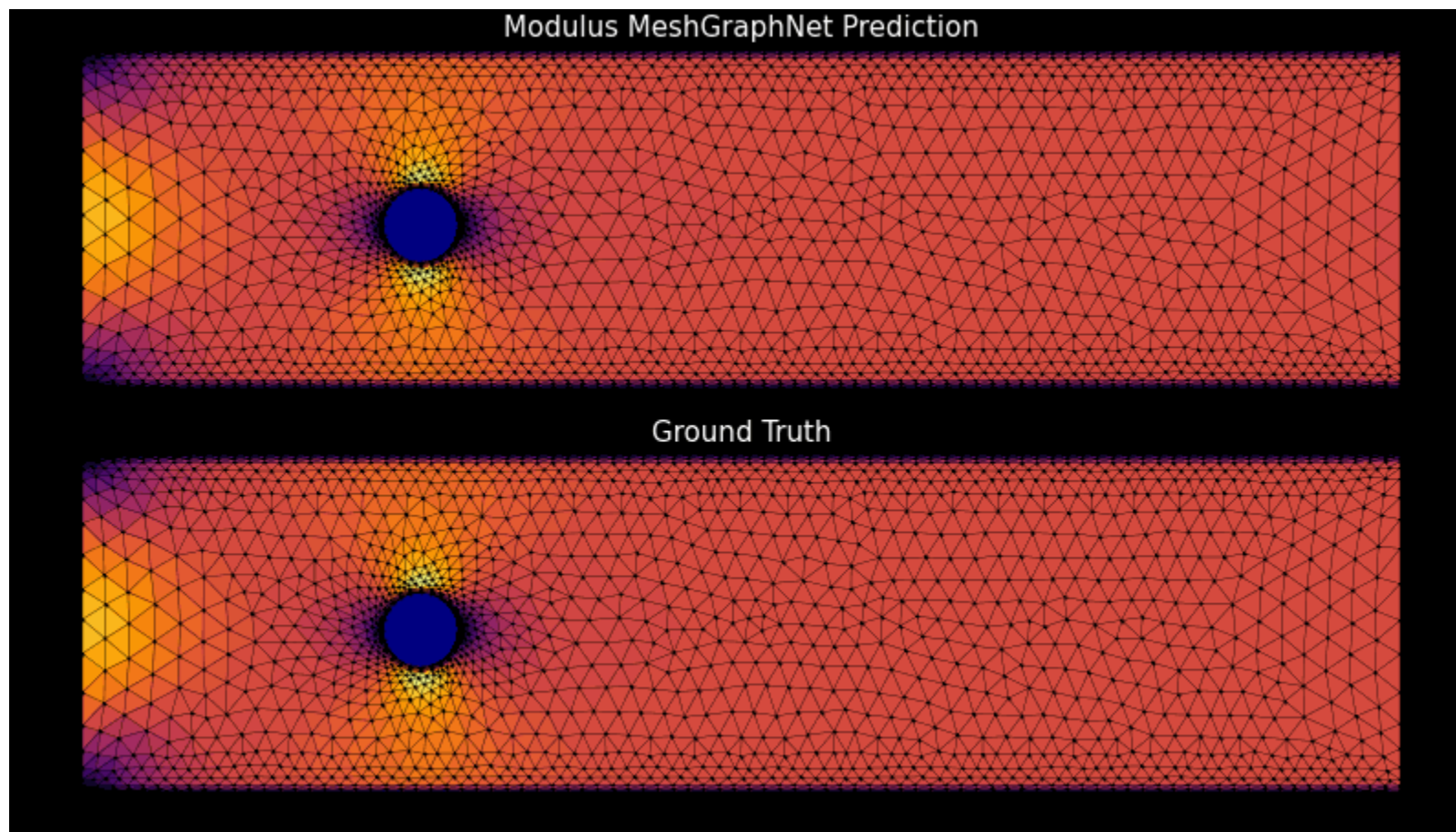
$$\mathbf{X}_{t+1} = \mathbf{X}_t + \Delta t \cdot f_{\text{MeshGraphNet}}(\mathbf{X}_t, G)$$

- \mathbf{X}_t ：當前時間步的節點特徵。
- Δt ：時間步長。
- $f_{\text{MeshGraphNet}}$ ：GNN 模型的輸出。

應用領域

1. 流體力學：模擬不可壓縮流體（如水、空氣）的運動。
2. 結構力學：模擬彈性體或結構的應力應變行為。
3. 多尺度建模：結合宏觀和微觀模擬，實現高效的多尺度模擬。

Graph Neural Network Support



1.1 PINN 與 RNN 的整合

PINN-RNN 將 PINN 用於學習物理場中的空間分佈特性，RNN 用於建模時間相關的動態行為。其數學表達如下：

1. 空間輸入與物理場輸出：

$$u(x, y, z, t) = \text{NN}(x, y, z, h_t)$$

- u ：物理量（如溫度、速度、壓力等）。
- (x, y, z) ：空間座標。
- h_t ：RNN 在時間 t 的隱藏狀態。

2. RNN 的時間建模：

$$h_t = f_{\text{RNN}}(h_{t-1}, t; \theta_{\text{RNN}})$$

- h_t ：時間 t 的隱藏狀態。
- θ_{RNN} ：RNN 的參數。
- f_{RNN} ：遞迴更新函數（例如 LSTM 或 GRU）。

3. PINN 的控制方程損失：PINN 通過物理方程的殘差最小化進行學習：

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \|\mathcal{N}(u(x_i, y_i, z_i, t_i))\|^2$$

- $\mathcal{N}(u)$ ：物理控制方程的殘差。
- N_f ：訓練點的數量。

4. 數據損失與邊界條件損失：

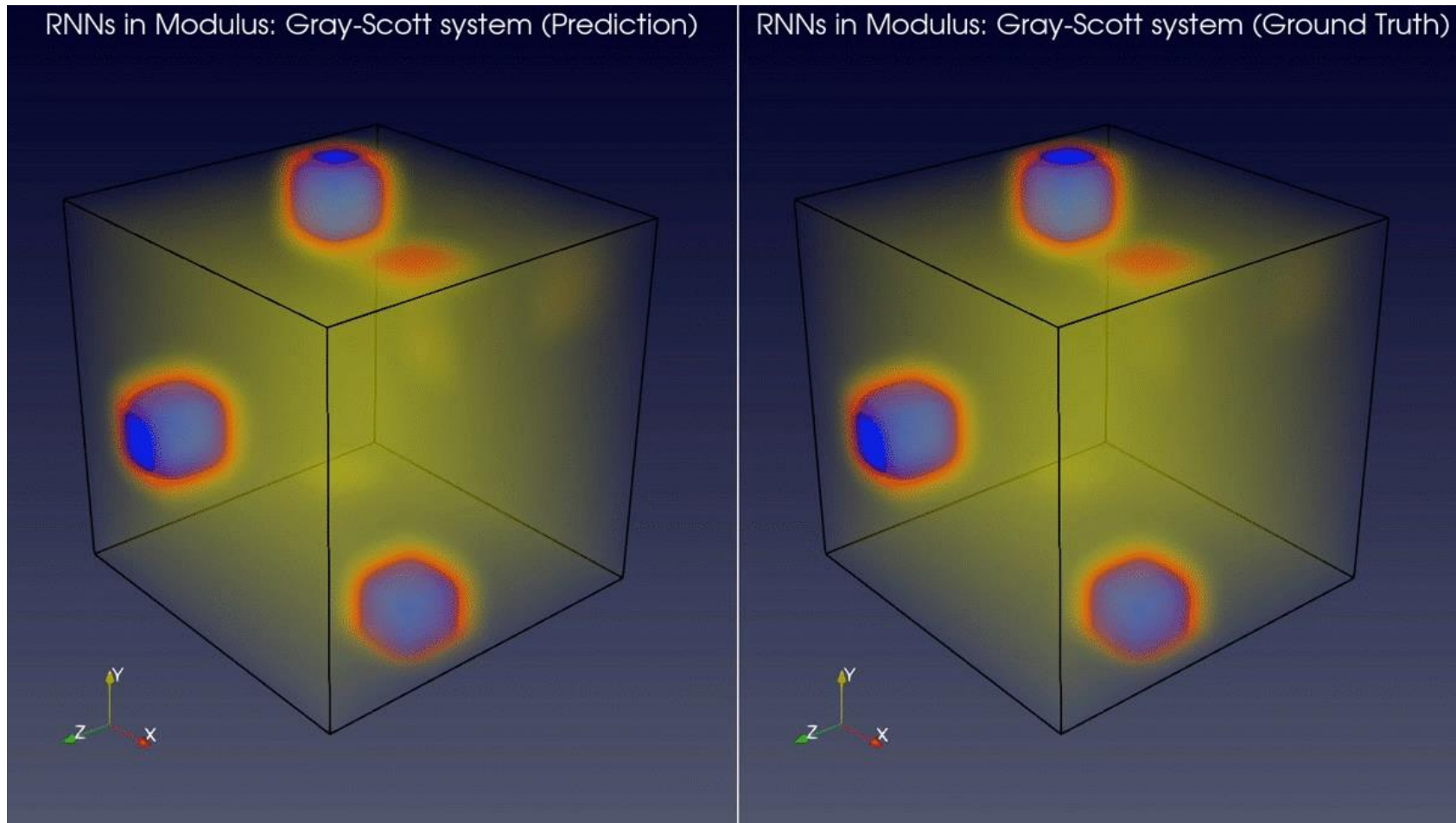
$$\mathcal{L}_{\text{data}} = \frac{1}{N_d} \sum_{i=1}^{N_d} \|u(x_i, y_i, z_i, t_i) - u_{\text{data},i}\|^2$$

$$\mathcal{L}_{\text{BC/IC}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \|u(x_i, y_i, z_i, t_i) - u_{\text{BC/IC},i}\|^2$$

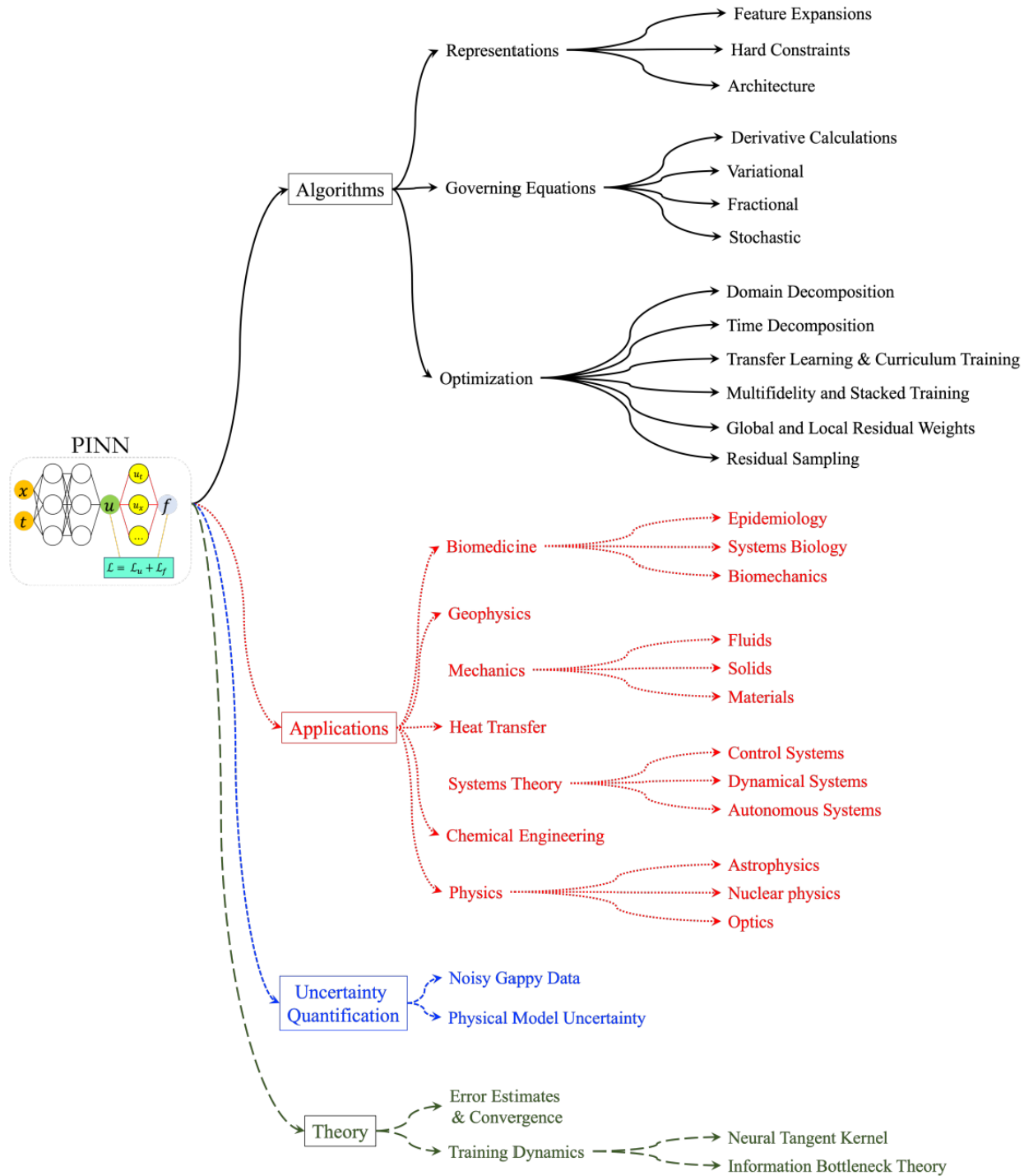
5. 總損失函數：

$$\mathcal{L} = \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{RNN}} + \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{BC/IC}}$$

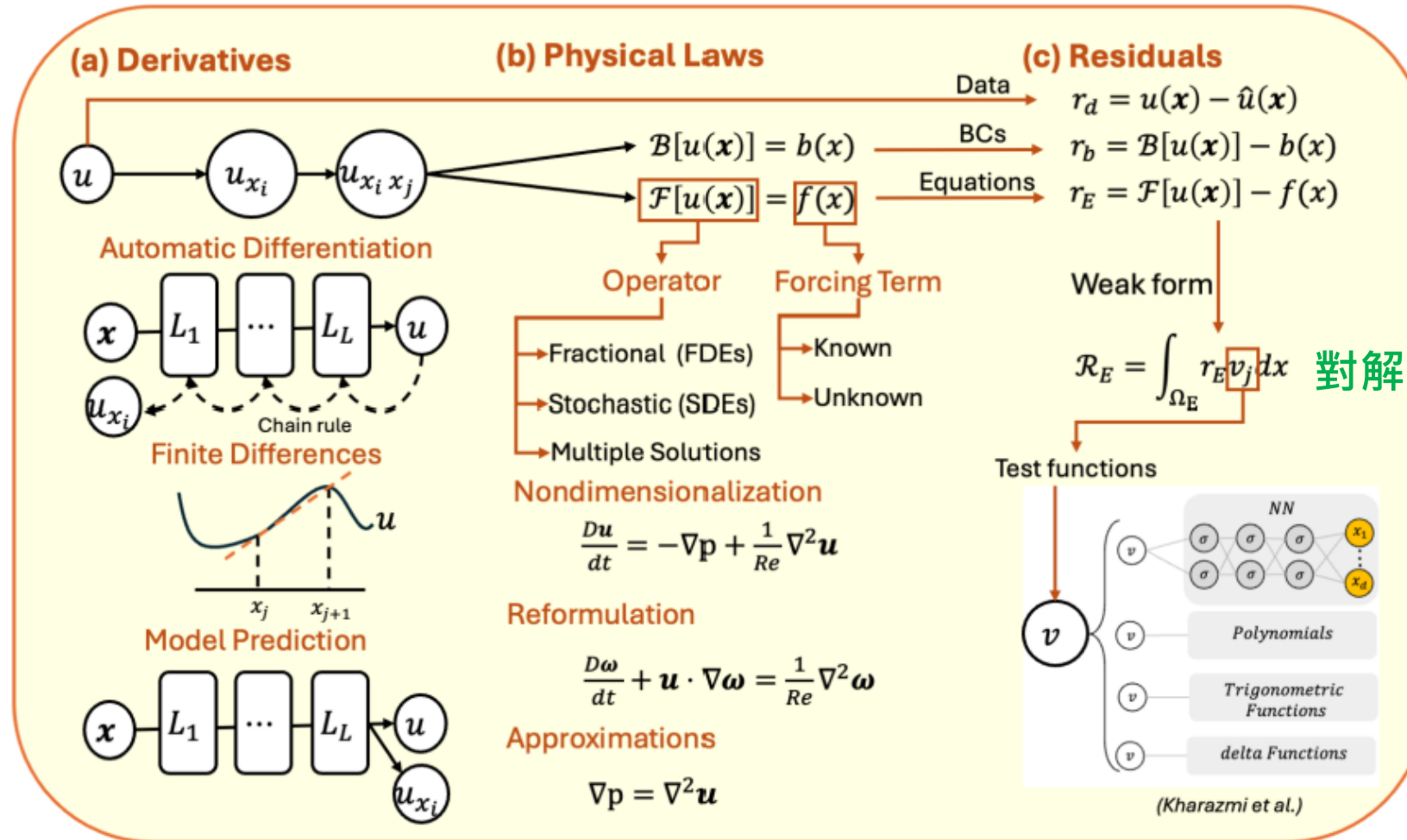
Recurrent Neural Network Support



$$\partial_t \mathbf{q} = \underline{\underline{D}} \nabla^2 \mathbf{q} + \mathbf{R}(\mathbf{q}) \quad \text{擴散+反應}$$

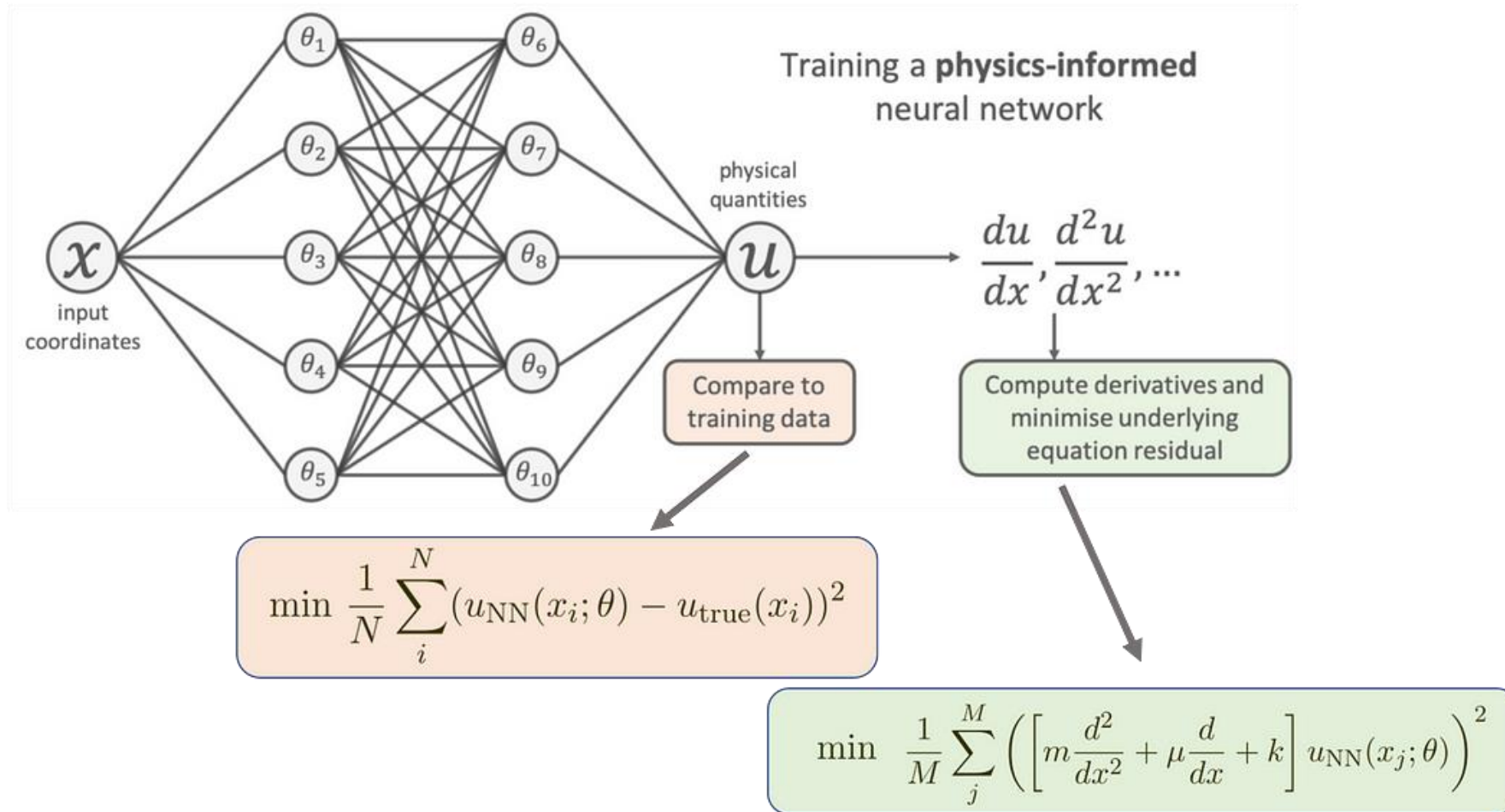


Governing Equations

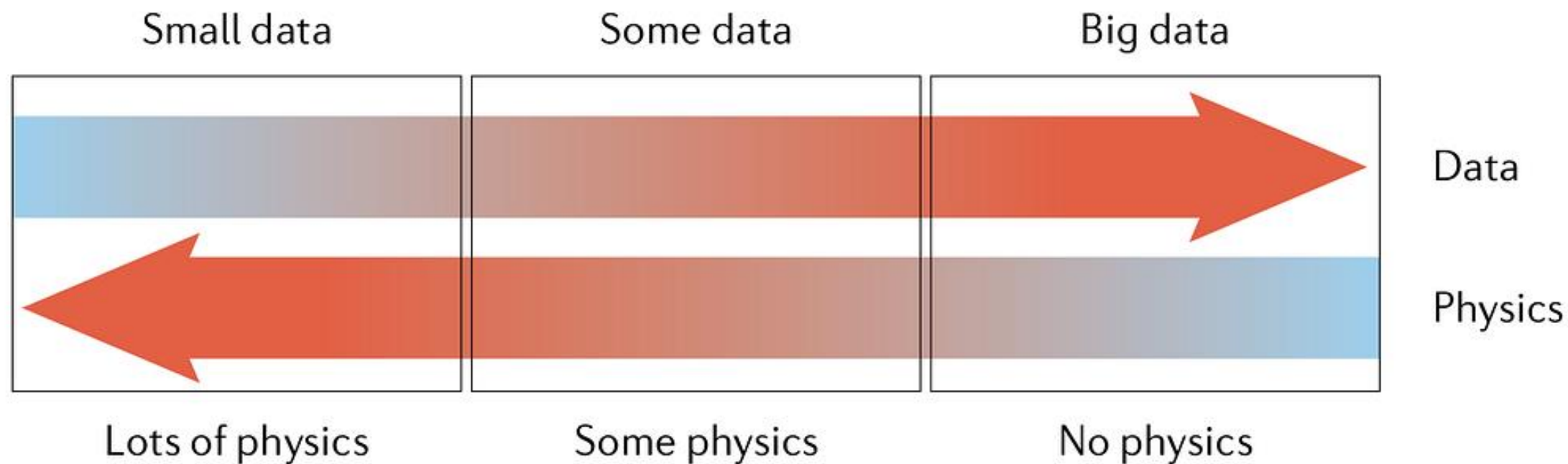


對解的平滑度要求降低

Physics-Informed Neural Networks

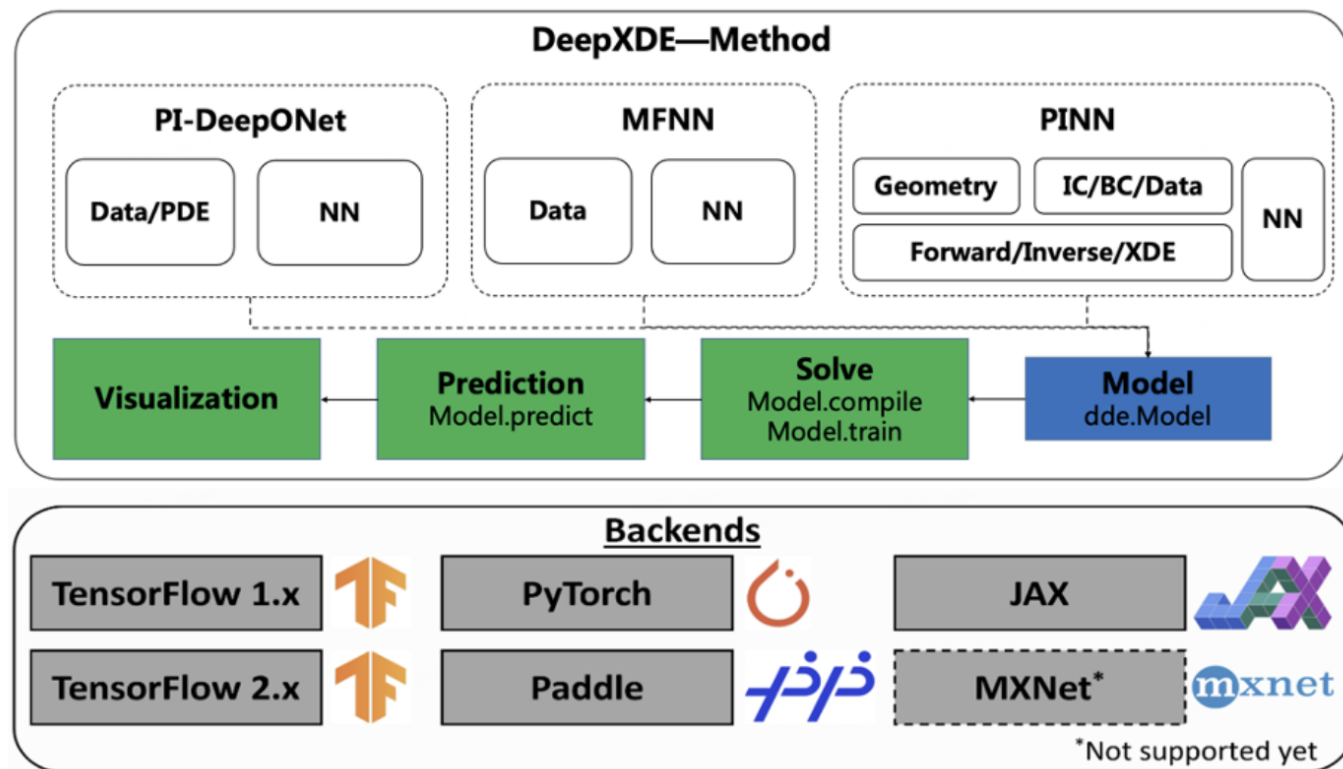


Physics-Informed Neural Networks

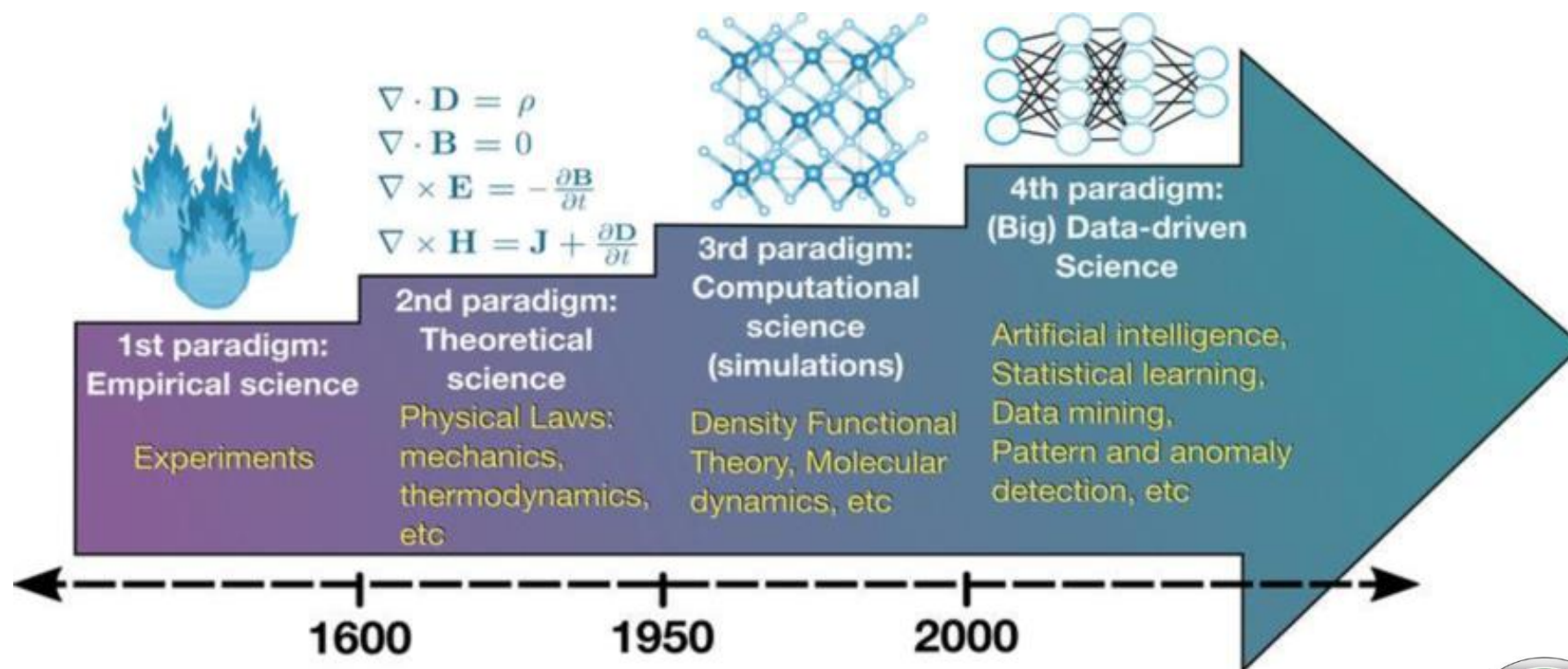


DeepXDE 是一個專為科學機器學習與物理引導學習設計的 Python 函式庫。它提供工具使用神經網絡來解決各類型的微分方程，主要功能包括：

- 物理引導神經網絡 (Physics-Informed Neural Networks, PINNs)：用於求解常微分方程 (ODEs)、偏微分方程 (PDEs)、積分微分方程 (IDEs)、以及分數階偏微分方程的正向與反向問題。
- 深度運算網絡 (Deep Operator Networks, DeepONets)：用於學習運算子以建模複雜系統。
- 多保真度神經網絡 (Multifidelity Neural Networks, MFNNs)：整合不同來源且具不同精度的數據。



Data-driven Science



經驗

理論

計算

資料驅動

牛頓

電腦

機器學習



Recurrent
Neural Network

