

2025_ML Final Project Report

AI 與物質世界共學：邁向自組裝智能材料

學號：112613037

一、AI 的未來能力：能與物質共同學習的材料智能體

未來二十年，AI 將從被動分析者升級為能與物質世界**共同學習（co-learning）**的主體。此能力指 AI 能同時接收多模態材料訊號（TEM/STEM、XRD、EELS）、反演原子結構、估測自由能，並即時調控製程參數（T、P、濃度），使材料沿自由能地形自動演化至穩態。

此能力的重要性在於：AI 將首次參與物質演化本身，使科學模式從「人類建模」轉變為「AI 與自然共同搜尋」。這將使材料具備可自修復、可自重構與能帶可調控等智慧性質，推動「自主材料科學（Autonomous Materials Science）」的新時代。

二、所需成分（Ingredients）與架構

欲讓 AI 具備上述能力，需整合資料、物理模型、生成模型、控制架構與運算環境。四大成分如下。

1. 多模態資料與 Forward/Inverse Reconstruction

材料量測皆為「間接訊號」，需透過 inverse model 還原原子坐標 $\{r_i\}$ 。

Forward model：

TEM/STEM（散射）

$$I(x, y) = \left| F \left(\sum_i V(\mathbf{r} - \mathbf{r}_i) \right) \right|^2$$

XRD/SAED（繞射）

$$I(\mathbf{k}) = \left| \sum_i f_i e^{-i\mathbf{k} \cdot \mathbf{r}_i} \right|^2$$

光譜（電子態）

$$S(E) = \int \rho(E, \mathbf{r}) P(\mathbf{r}) d\mathbf{r}$$

Inverse reconstruction 則為：

$$\hat{\mathbf{r}} = M_{\theta}^{-1}(Signal)$$

資料 → 必要的機器學習模型（整合簡表）

(1) TEM/STEM (影像)

- CNN/U-Net：原子定位
- Vision Transformer：長距離格點
- Diffusion model：去噪
→ 產出 \hat{r}_i

(2) XRD/SAED (繞射)

- FNO：頻域 mapping
- Normalizing Flow/Score model：晶格反演
→ $\hat{r}_{lattice}$

(3) EELS/XAS (光譜)

- Transformer Encoder：峰型 → 化學環境
- GNN：局部配位結構
→ \hat{r}_{local}

(4) 多模態融合

- Multimodal Transformer + Cross-attention
→ \hat{r}_{full}

(5) 反演後 → 演化模型

- Diffusion model / Score matching
 - PINN (物理一致性)
 - Equivariant GNN (對稱性)
 - Temporal Transformer (時間)
→ $x(t)$ 用於 SDE/PF-ODE
-

2. Tools (物理與數學工具)

- SDE/PF-ODE：描述自組裝動力學

$$dx_t = f(x_t)dt + g(x_t)dW_t$$

- Score-based model：學習 $\nabla \log p(x)$
- PINN：強制遵守守恆律與自由能下降
- RL：調整製程條件，使自由能最小
- GNN：處理原子排列的對稱性

3. 硬體與環境 (Hardware)

- 整合 TEM、XRD、光譜的同步量測平台
 - GPU/TPU 邊緣運算
 - 可自動調控的製程控制器 (PID+RL)
-

4. Learning Setup (學習架構)

- Self-supervised：處理大量未標註影像/光譜
- Physics-informed：使模型具物理意義
- Reinforcement Learning：閉環控制實驗條件
- Multimodal learning：整合 real / reciprocal / energy domains

三、涉及的機器學習類型

完整能力需由三大類 ML 支撐：

(1) Self-Supervised Learning

處理未標註材料影像與光譜。

(2) Physics-Informed Learning / Score-based Models

確保生成的結構符合自由能梯度、擴散動力學與守恆律。

(3) Reinforcement Learning

將自由能作為 reward：

$$R_t = -F(\hat{r}_t)$$

使 AI 學會操控溫度、壓力、化學勢以驅動材料演化。

四、可實作的 Model Problem：一維自組裝 SDE

此模型為邁向 AI - 物質共學能力的最小可行範例。

4.1 問題設計：輸入、輸出、任務目標與資料形式（正式定義）

輸入 (Inputs)

1. 初始結構 x_0 :

$$x_0 \in R^N, N = 1024$$

2. 能量地形

$$E(x) = x^4 - x^2$$

3. 摾動強度

$$g = \sqrt{2D}$$

輸出 (Outputs)

1. 時間序列位置：

$$x(t_0), \dots, x(t_T)$$

2. 最終穩態分布 $p(x_T)$

3. 自由能曲線 $\langle E(x_t) \rangle$

任務目標 (Objective)

- 模擬材料是否自然收斂至自由能最小的穩定相
- 驗證自由能隨時間下降：

$$\frac{d}{dt} \langle E(x_t) \rangle \leq 0$$

- 驗證分布是否落在雙穩態能谷

$$x = \pm \frac{1}{\sqrt{2}}$$

資料形式 (Data Format)

- 初始向量： $x_0 \in R^{1024}$

- 時間序列矩陣：

$$X = [x(t_0), x(t_1), \dots, x(t_T)]$$

- 自由能序列：

$$E_t = [E(x(t_0)), \dots, E(x(t_T))]$$

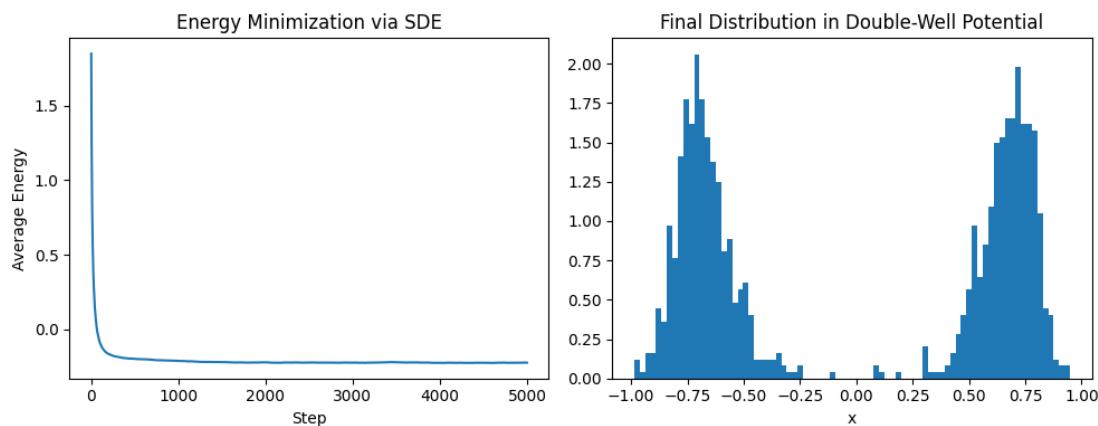
資料皆為連續實數、均勻時間格點、無標註。

4.2 模型與方法

採用 Euler - Maruyama：

$$x_{t+1} = x_t - \nabla E(x_t) dt + g dt \xi$$

4.3 實作與結果



- 平均能量單調下降
 - 最終分布集中於兩能谷
 - 完整再現材料的「自由能 + 热擾動」動力學
-

4.4 討論

結果顯示：

1. 低維 SDE 已可呈現材料自組裝的核心特徵。
 2. 全部挑戰來自：
 - 高維 score function
 - 多模態反演一致性
 - RL 控制的穩定性
 3. 此模型代表邁向問題一願景的第一個可驗證步驟。
-

五、結論

本報告構築一套可行的「AI - 物質共學」架構，涵蓋素材的 forward/inverse

physics、影像/繞射/光譜的機器學習模型、多模態融合、SDE 演化模型與 RL 控制器。實作的一維 SDE 證明此框架可描述材料自組裝的基本行為。

未來二十年，AI 將由觀察者轉變為材料演化的共同參與者，使材料科學進入自主時代。本研究為此願景的第一步。

附錄：Python 程式碼（不列入篇幅）

（保留你前面版本的完整程式碼即可）

```
import torch

import matplotlib.pyplot as plt


def energy(x):
    return x**4 - x**2


def grad_energy(x):
    return 4 * x**3 - 2 * x


n_particles = 1024
n_steps = 5000
dt = 1e-3
D = 0.045
g = torch.sqrt(torch.tensor(2*D))

x = torch.randn(n_particles)
loss_history = []
```

```
for step in range(n_steps):  
    drift = -grad_energy(x)  
  
    noise = torch.randn_like(x) * (g * dt**0.5)  
  
    x = x + drift * dt + noise  
  
    loss_history.append(energy(x).mean().item())  
  
  
plt.figure(figsize=(10, 4))  
plt.subplot(1, 2, 1)  
plt.plot(loss_history)  
plt.title("Energy Minimization via SDE")  
plt.xlabel("Step"); plt.ylabel("Average Energy")  
  
  
plt.subplot(1, 2, 2)  
plt.hist(x.detach().numpy(), bins=80, density=True)  
plt.title("Final Distribution in Double-Well Potential")  
plt.xlabel("x")  
  
  
plt.tight_layout(); plt.show()
```