

# Final Project Report — AI Governance Simulator

## 摘要 (Abstract)

本專案提出並實作一個面向「全自動城市治理 (AI Governance Simulator)」的簡化研究路徑：從未來願景出發，分析所需成分，並設計一個可實作的 model problem (簡化城市交通—能源模擬)，以驗證核心概念。報告包含願景描述、成分討論、機器學習類型判斷，以及完整的可執行實作與結果分析。所附程式碼位於 `traffic_energy_rl.py` (含環境、fallback policy、測試、訓練/評估流程)。

---

## 1. AI 的未來能力 (Problem 1)

能力名稱：全自動城市治理模擬與政策生成系統 (AI Governance Simulator)。

二十年後，我預期 AI 能夠在城市層級整合多來源 (交通、能源、環境、經濟、健康) 時空資料，自主建構高維因果模型，並在虛擬環境中模擬、評估多方案政策 (含短期衝擊與長期副作用)，最終產出兼顧效率、公平與永續的政策建議。此系統不僅做因果預測，還能建議需要新增的觀測或實驗以驗證假說，並提供可解釋的決策理由。

應用場景 (舉例)：

- 城市級碳減量策略：模擬碳稅、公共運輸投資、停車管制等對排放、通勤與房市的交互影響。
- 危機應對：在傳染病或颱風情境下模擬醫療負載、避難動線與供應鏈穩定性，協助資源調度。

此能力重要性在於：城市系統為高度耦合的複雜系統，單一領域的優化常造成跨領域外部性；能進行跨域、多目標、長期預測的 AI 有助於更科學化的公共治理，並提高政策透明性與公民參與品質。

---

## 2. 所需的成分與資源 (Ingredients)

為實現上述能力，系統需整合以下成分，並說明其角色與相互依存：

### 2.1 資料 (Data)

- **多模態時空資料**：交通感測器、智慧電表、遙測影像、人口動態、衛生醫療資料、經濟統計、社群媒體事件流。需高時間解析度（分一小時級）與空間解析度（街區/網格），並包含長期歷史紀錄（數年）以捕捉趨勢。
- **標註與干預記錄**：政策實施時間、受影響區域、資源分配紀錄，作為因果推論的外生衝擊資料。

### 2.2 工具 (Tools)

- **時空表徵學習**：Graph Neural Networks (GNN) 或 spatio-temporal transformers，建構城市網路上的低維表示。
- **生成與模擬工具**：基於 agent-based modelling 的城市模擬器（或 SUMO/CityLearn 接口）來產生交互行為樣本。
- **強化學習 (RL) 與多目標最佳化**：在模擬環境中尋找政策序列，利用多目標 reward（效率、公平、永續）進行折衷。
- **因果推論套件**：以差分法、工具變數 (IV)、結構方程 (SEM) 檢驗政策因果效應，避免僅憑相關性優化。

### 2.3 硬體與環境 (Hardware / Environment)

- 分布式雲端計算與 GPU/TPU 用以訓練大型時空模型與 RL 代理。
- 感測器網路與資料串流基礎建設，若要部署需與城市 IoT 系統整合。

### 2.4 學習架構 (Learning Setup)

- **自監督學習 (self-supervised)** 作為表示學習的基礎；**非監督式/對比學習** 用於抽取潛在因子。
- **強化學習 + 模擬器** 負責試驗與策略優化；**meta-learning** 協助跨城市快速 adaptation。
- **\*\*人類回饋 (RLHF / human-in-the-loop) \*\*** 用以注入倫理/法律偏好與審查。

各成分互依：表示學習提供給 RL 更好的狀態空間；因果推論驗證 RL 建議的實際可行性；模擬器基於生成模型補足稀疏干預資料。

---

### 3. 涉及的機器學習類型（Methods）

主體為 非監督式/自監督學習 + 模擬驅動的強化學習（RL）：

- 自監督/非監督：從海量無標籤的城市感測資料中學到穩健表徵（資料來源為感測器與遙測），目標為壓縮與重建誤差或對比式目標。
  - 強化學習：在模擬環境中做政策搜尋，目標訊號為多目標 reward（例如 weighted sum of travel\_time、emissions、inequality 指標），存在明確環境互動與長期回饋。
  - 輔以監督學習（對於特定預測任務如需求預測）與因果推論方法做驗證。
- 

### 4. 第一個可實作的模型問題（Solvable Model Problem）

#### 問題設計

簡化為「

在一個 toy 城市中，同時控制交通（aggregate congestion）與能源需求，目標最小化通勤時間與能源排放的加權和。

」

此問題保留了多目標權衡、代理互動與政策一回饋迴圈，是通往完整治理系統的核心子問題。

#### 輸入/輸出/目標

- 輸入：狀態向量  $s_t = (\text{traffic\_load}, \text{energy\_demand}, \text{time\_of\_day})$
- 輸出：動作  $a_t \in \{\text{invest\_transport}, \text{demand\_response}, \text{price\_increase}, \text{do\_nothing}\}$
- 目標：最小化累積  $\text{cost} = \sum_t [\alpha * \text{emissions}_t + \beta * \text{commute\_time}_t]$
- 資料形式：由 Python 模擬器生成的時序資料（可控制隨機種子）

#### 模型與方法

- 使用簡單 Gym（或 Gymnasium）環境實作動態規則；
- 若 stable-baselines3 可用，採用 PPO；否則提供 deterministic FallbackModel 作為 baseline。

## 實作與結果（程式碼與實驗）

- 已實作 traffic\_energy\_rl.py：包含 CityTrafficEnergyEnv、FallbackModel、訓練/評估流程與單元測試。
- 在缺少 stable-baselines3 的環境下，fallback policy 保證可執行並能產生基準結果；若安裝 SB3，則可執行 PPO 訓練並觀察 reward 下降與策略改善。

## 討論

從此簡化實作學到：

1. **reward 設計敏感度高**：不同  $\alpha, \beta$  導致截然不同的策略傾向。
2. **模擬真實性限制**：toy dynamics 雖可驗證方法，但真實城市需更精細行為模型與因果干預資料。
3. **可擴展性議題**：從單區域到多區域、從單代理到多代理，計算與穩定性挑戰明顯。

---

## 結論與下一步

本專案提供了一個自治的研究路徑：從願景→成分→方法→可執行 model problem。下一步建議：

1. 若要進一步提升可信度，將環境替換為 SUMO/CityLearn 並加入異質 agent；
2. 將 fallback policy 換成離線 Q-learning 或自實作簡易 RL 演算法，以無需外掛訓練動機；
3. 設計因果檢驗（A/B 或合成控制法）來驗證模擬策略在真實世界的外推性。

---

## 附錄（檔案結構）

- `traffic_energy_rl.py` — 主程式（環境、策略、測試、訓練、評估、繪圖）
- `requirements.txt` — 建議套件
- `README.md` — 使用說明
- 實驗圖表判讀說明