現有檔案 (processed_oskg_data.pt) 說明：

這是一個非常好的時間點來進行盤點。身為您的指導教授，我很高興看到您已經完成了最艱難的「資料工程」階段。processed_oskg_data.pt 的誕生，代表您的專案已經從「資料清洗」正式進入了「模型研發」的核心階段。

這份 .pt 檔不僅僅是一個儲存檔，它是我們接下來所有演算法的**地基**。

以下我將針對您的提問，結合您上傳的 osh_kge_gnn_plan.pdf 以及參考論文 Injecting Knowledge Graph into LLM 的核心精神，為您拆解接下來的戰略藍圖。

第一部分：processed_oskg_data.pt 的戰略定位

這份檔案是將非結構化的法律文本與圖譜結構，經過清洗、剪枝、向量化後，濃縮成的**「可微分知識庫 (Differentiable Knowledge Base)」**。

具體來說，它將在未來扮演以下三個關鍵角色：

1. **解決 GNN 的「冷啟動 (Cold Start)」問題：**
    - 傳統 GNN (如 GraphSAGE) 的輸入特徵通常是隨機初始化的。
    - 但在您的 .pt 檔中，node_features 已經是經過 BERT 預訓練的語意向量。這意味著您的模型在 Epoch 0 時，就已經「讀過」勞基法與職安法，具備了基本的語意理解。GNN 只需要學習「結構邏輯」，而不必從頭學習「文字意義」。
2. **提供「對抗性訓練」的彈藥庫：**
    - 檔案中的 hard_negatives 字典是訓練高品質 KGE (如 RotatE) 的關鍵。
    - 在訓練 Loss Function (如 Contrastive Loss) 時，模型不能只看簡單的負樣本（隨機採樣），必須看這些「高難度負樣本」（字面上很像但邏輯錯誤的法條）。這份檔案已經幫您預先算好了這些候選者，能大幅加速訓練效率。
3. **作為「神經符號推理 (Neuro-Symbolic Reasoning)」的檢索索引：**
    - 未來當 LLM 需要推理時，我們不會把整本法規塞進 Prompt。我們會利用這個檔案中的圖結構 (adj) 進行子圖檢索 (Subgraph Retrieval)，只將與當前案例相關的「違規-法規-母法」鏈條提取出來。

第二部分：研究藍圖拆解 (Roadmap)

根據您的 osh_kge_gnn_plan.pdf 與參考論文，我將您的後續任務重新架構為三大階段。這是一個從「理解結構」到「注入知識」再到「實際推理」的過程。

階段一：圖譜表示學習 (Graph Representation Learning)
目標：讓模型學會職業安全法律的「隱式邏輯」 (e.g., 知道 A 導致 B，且 B 違反 C)。

這是目前最急迫的任務，直接使用 .pt 檔。

- **任務 1.1: 建立 Dataset Class (PyG/DGL)**
  - **動作**：撰寫一個 Python Class 繼承自 InMemoryDataset (若用 PyG)。
  - **細節**：讀取 .pt 檔，將 node_features 轉為 x，將 NetworkX 的邊轉為 edge_index，並將 relation 轉為 edge_type。
  - **關鍵**：確保 VIOLATES_SPECIFICALLY 與 HAS_CAUSE 等關係被編碼為不同的 ID。

- **任務 1.2: 訓練 R-GCN (Relational GCN) 編碼器**
  - **依據**：參考論文中提到利用 GNN 聚合鄰居訊息。
  - **動作**：設計一個 2-3 層的 R-GCN。
  - **輸入**：BERT Embeddings (processed_oskg_data.pt 中的特徵)。
  - **輸出**：Context-Aware Embeddings (考量了法律層級與因果關係後的新向量)。
  - **Loss Function**：
    - **Link Prediction Loss**：預測 (Head, Relation, ?) 的 Tail。
    - **使用權重**：務必引入我們計算好的 class_weights，重罰 HAS_CAUSE 的預測錯誤。
  - ❖ 其他可考慮比較的方法
    - ➢ RGAT (Relational Graph Attention Network)
    - ➢ GraphSAGE (Graph Sample and Aggregate)
    - ➢ (optional) HGT (Heterogeneous Graph Transformer)

- **任務 1.3: 整合 KGE (RotatE) 進行邏輯幾何映射**
  - **為什麼要這個？** R-GCN 擅長特徵聚合，但 RotatE 擅長模擬邏輯關係 (如：$h + r = t$ 在複數空間的旋轉)。
  - **策略**：可以將 R-GCN 的輸出作為 RotatE 的初始向量，進行 Fine-tuning。這能讓模型學會「反對稱性」 (A 導致 B $\neq$ B 導致 A)。
  - ❖ 其他可考慮比較的方法
    - ➢ TransE (Translational Embedding) — 必跑的 Baseline
      - **原理**：$h + r \approx t$ (向量加法)。
      - 為什麼選它：這是 KGE 領域的 "Hello World"。
      - 戰略意義：您必須跑 TransE，通常它的效果不會比 RotatE 好（因為它處理不好 1-to-N 的關係，如一個事故違反多條法規），但如果您沒跑 TransE，論文審查委員一定會挑戰您。
      - 您需要證明「複雜的 RotatE 確實比簡單的 TransE 有效」。
    - ➢ (optional) ComplEx (Complex Embeddings)

➢ (optional) BoxE (Box Embeddings)

- 任務 1.5：結構化思維鏈微調 (Structural CoT SFT)
  - 目標：利用圖譜的黃金路徑 (Golden Paths) 生成高品質問答對，訓練 LLM 的邏輯推理。

步驟 A：黃金路徑採樣 (Golden Path Sampling)

不要隨機從 JSON 抓資料。您需要寫一個 Python 腳本，同時讀取 .pt (結構) 和 .json (文字)。

1. **讀取 .pt**：獲取所有保留下來的 VIOLATES_SPECIFICALLY (高置信度) 邊。
2. **路徑回溯 (Backtracking)**：
   - 從一個 Regulation 節點出發，透過 .pt 中的邊，反向找到連接的 Violation (違規行為)。
   - 再從 Violation 反向找到導致它的 Cause (原因)。
   - 再從 Cause 反向找到 Incident (事故)。
3. **提取文字**：拿著這些節點 ID，去 json 檔裡抓出對應的中文描述 (label 或 full_text)。
   - *注意：這就避開了那 19,830 條垃圾邊，只用了我們精選過的 1,605 條邊。*

步驟 B：思維鏈模板構建 (CoT Prompt Construction)

我們不要只訓練 Q: 發生什麼事? A: 違反第6條。 這種淺層對答。我們要訓練模型**模仿法律人的思考過程**。

請設計如下的 JSON 格式訓練資料 (Instruction Tuning Data)：

```JSON
{

"instruction": "請根據以下職業災害事故描述，進行法律歸責分析，並指出違反的特定法條。",

"input": "工人於高空作業車上進行外牆維修時，因未配戴安全帶，導致重心不穩墜落。（這裡填入 Incident + Cause 的文本）",

"output": "【事故分析】：\n1. 核心事實：工人在高處作業，且未配戴安全帶。\n2. 歸責推論：此行為屬於高處作業防護缺失。\n3. 對應法規：這違反了《職業安全衛生設施規則》第 281 條。\n4. 法條內容：雇主對於在高度二公尺以上之高處作業，勞工有墜落之虞者，應使勞工確實使用安全帶、安全帽及其他必要之防護具...\n\n結論：該事故主要違反了《職業安全衛生設施規則》第 281 條。"

}
```

- **關鍵點**：Output 中的「推論步驟」，正是由我們圖譜中的 Incident -> Cause -> Violation -> Regulation 路徑所支撐的。

步驟 C：對抗性 SFT (Contrastive SFT) - 這是加分題

利用 .pt 檔中的 hard_negatives (困難負樣本)，我們可以生成更高級的訓練資料，教模型「什麼是不對的」。

```JSON
{

"instruction": "工人於高空作業車上墜落，請問這是否違反了《屋頂作業作業主管配置規定》？",

"input": "...",

"output": "分析：\n雖然這涉及高處作業，但案發地點為「高空作業車」而非「屋頂」。\n因此，這並不適用《屋頂作業...》相關規定。\n\n正確適用的法條應為《職業安全衛生設施規則》關於高處作業與安全帶之規定。"

}
```

- 這能讓模型學會**區辨** (Discrimination)，而不僅僅是關鍵字匹配。

教授的總結 (The Next Step)

您現在的任務非常清晰：

1. **保留 .pt**：它是您的**導航圖** (只有乾淨的路徑)。
2. **保留 .json**：它是您的**字典** (提供路徑上的景點介紹)。
3. **下一步動作**：撰寫一個 generate_sft_data.py，**以 .pt 為篩選器，去 .json 撈資料**，生成上述的 SFT JSONL 檔案。

這一步做完，您的模型將不只是一個分類器，而是一個能「說出道理」的 AI 法律顧問。這是非常高階的應用。

階段二：知識注入與對齊 (Knowledge Injection & Alignment)

目標：參考 Injecting Knowledge Graph into LLM 論文，將訓練好的圖譜知識「餵」給 LLM。

- **任務 2.1: 知識適配器 (Knowledge Adapter) 訓練**
  - **概念**：您的 Graph Embedding 維度 (假設 256 或 512) 與 LLM (如 Llama-3, 4096維) 不同。
  - **動作**：訓練一個簡單的 Linear Projector (W_{proj})，將 Graph Vector 投影到 Text Vector 空間。
  - **訓練目標**：讓 Projector(Graph_Vector) 與 LLM(Text_Description) 的相似度最大化。
- **任務 2.2: 檢索增強生成 (Graph-RAG)**
  - **動作**：當 User 輸入一個職安案例時：
    1. 先用 BERT 找到圖譜中最相似的 Incident 節點。
    2. **BFS/DFS 搜索**：從該節點出發，沿著 HAS_CAUSE 和 VIOLATES_SPECIFICALLY 抓取 2-hop 的鄰居。
    3. 將這些鄰居 (法規、原因) 轉回文字，作為 LLM 的 Prompt Context。

階段三：推理能力驗證 (Reasoning Evaluation)

目標：驗證模型是真的在「思考」，還是只是在「背誦」。

- **任務 3.1: 歸責推理測試 (Causal Reasoning Test)**
  - **題目**：給定一個新的事故描述 (不在訓練集中)。
  - **要求**：模型必須預測出正確的 Cause_Basic (基本原因)。
  - **指標**：Hits@1, Hits@3 (看看正確原因是否在前三名)。
- **任務 3.2: 法條適用性測試 (Legal Applicability Test)**
  - **題目**：給定事故與原因。
  - **要求**：模型必須預測出 VIOLATES_SPECIFICALLY 連接的法規。
  - **關鍵**：利用 .pt 中的 Hard Negatives 來測試模型是否會被「字面相似但邏輯錯誤」的法規混淆。

教授的總結與建議 (Professor's Advice)

您現在位於一個非常有利的位置。processed_oskg_data.pt 解決了最頭痛的髒資料問題。

接下來的一週，我建議您專注於「階段一：任務 1.1 與 1.2」。

不要急著接 LLM。先寫一個簡單的 PyTorch Geometric (PyG) 程式，讀取 .pt 檔，跑通 R-GCN 的 Link Prediction 任務，看看 Loss 是否能收斂，以及模型是否能區分 HAS_CAUSE。

一旦圖神經網路 (GNN) 訓練起來，它產生的 Embeddings 將會是您後續做任何 LLM 應用的強力基石。

加油！這是一個非常有深度的題目，做出來會是頂級會議 (如 ACL 或 EMNLP) 等級的成果。

較舊的計劃書版本：

將「靜態的 KGE (RotatE)」與「動態的 GNN (R-GCN)」結合，並搭配「生成式的 LLM」，這正是目前學術界最前沿的 **Neuro-Symbolic AI (神經符號人工智慧)** 完整體現。

以下我為你重新編排、整合後的 **「台灣職安法律智慧推論系統 (Taiwan OSH-Legal Reasoning System)」** 終極 Workflow。我將採用 **「由淺入深、動靜互補」** 的邏輯來串聯這兩套流程。

終極整合 Workflow：神經符號混合推論架構

核心設計哲學

- **靜態結構 (Static):** 利用 KGE (RotatE) 捕捉全域的法律邏輯結構（如法規階層）。
- **動態特徵 (Dynamic):** 利用 GNN (R-GCN) 融合文本語意並處理新進案件（Inductive）。
- **生成大腦 (Generative):** 利用 LLM (Llama-3) 進行最終的自然語言邏輯輸出。

Phase 1: 多模態混合表徵學習 (Hybrid Representation Learning)

目標： 打造一張既懂文字（Semantic）、又懂結構（Structural）、還能量化邏輯（Logical）的超級地圖。

- **Step 1.1: 語意特徵初始化 (Semantic Initialization)**
  - **操作：** 對你 JSON 檔中的每個節點（事故描述、法條內容），使用 **MiniLM (Sentence-BERT)** 進行編碼。
  - **產出：** 獲得初始節點特徵矩陣 X_{text}。這是模型「閱讀」能力的基礎。
- **Step 1.2: 結構化嵌入預訓練 (Structural Pre-training via KGE)**
  - **操作：** 使用 **RotatE** 對圖譜進行訓練（不看文字，只看結構）。
  - **目的：** 讓模型學會「因果傳遞」與「法規階層」的空間幾何關係。
  - **產出：** 獲得結構向量 H_{struct}。這可以作為後續 GNN 的輔助輸入或初始化參數，確保 GNN 不會跑偏。
- **Step 1.3: 圖神經網路融合 (Feature Fusion via R-GCN)**
  - **操作：** 構建 **R-GCN (Relational Graph Convolutional Network)**。
    - **輸入：** X_{text} (來自 1.1) 與圖譜結構。
    - **機制：** 讓事故節點透過 VIOLATED 邊吸收法條特徵，透過 HAS_INDUSTRY 邊吸收行業特徵。
  - **整合點：** 這裡你可以將 Step 1.2 的 KGE 向量與 Step 1.1 的文字向量拼接 (Concat)，作為 R-GCN 的輸入，達到最強效果。
  - **產出：** 最終的 **Context-Aware Embeddings**。這組向量同時包含了「這是一起墜落意外」的語意，以及「它在法律圖譜中位置」的結構資訊。

Phase 2: 課程式邏輯注入訓練 (Curriculum-based Model Training)

目標： 訓練 LLM (Llama-3-Taiwan)，從「學會基礎法規」進階到「分辨複雜案情」。

- **Step 2.1: 基礎邏輯注入 - 路徑採樣 (Path Sampling for SFT)**
  - **來源：** *[舊 Workflow 核心]*
  - **操作：** 從圖譜中隨機遊走，提取標準的法律三段論路徑。
    - *路徑：* [事故] -> [成因: 開口未防護] -> [違反: 設施規則224條]
  - **轉換：** 利用 Prompt Template 轉為問答對。
  - **目的：** 這是「基礎教育」，讓 LLM 學會基本的法律對應關係。
- **Step 2.2: 進階辨析 - 困難負樣本挖掘 (Hard Negative Mining with GNN)**
  - **來源：** *[新 Workflow 核心]*
  - **操作：** 利用 Phase 1 訓練好的 R-GCN 計算節點相似度。
    - 找出那些「向量距離很近（看起來很像），但連結的法條完全不同」的案例。
    - *例子：* 同樣是「手部捲入」，一個是因為「沒停機」，一個是因為「沒設護欄」。
  - **生成資料：** 製作「比較型」問答資料：「請問這兩個類似的捲夾事故，為何判決不同？」
  - **目的：** 這是「高等教育」，訓練 LLM 的邏輯邊界，減少混淆。
- **Step 2.3: 監督式微調 (Supervised Fine-Tuning, SFT)**
  - **操作：** 將 Step 2.1 (基礎) 與 Step 2.2 (進階) 的資料混合，對 **Llama-3-Taiwan-70B** 進行微調。
  - **產出：** 一個具備台灣職安法專業知識與邏輯辨析能力的 LLM。

Phase 3: 歸納式推理與動態檢索 (Inductive Inference & GraphRAG)

目標： 實際上線應用。處理使用者輸入的「全新」職安事故描述，並給出專業建議。

- **Step 3.1: 新案件接入 (Inductive Node Injection)**
  - **情境：** 使用者輸入：「昨晚工廠發生乙炔外洩爆炸...」
  - **操作：**
    - 將這段文字過 MiniLM，轉為向量。
    - 利用 k-NN 演算法，將這個新節點暫時「掛」在圖譜上最相似的幾個舊案例旁邊。
    - **啟動 GNN 推論：** 讓 R-GCN 進行訊息傳遞 (Message Passing)，預測這個新節點與哪些「法條節點」存在高機率的潛在連結。
- **Step 3.2: 結構化檢索增強 (GraphRAG Retrieval)**
  - **操作：**
    - **檢索：** 根據 GNN 的預測，抓出 Top-K 最相關的法條與歷史判決。
    - **子圖提取：** 不只抓法條文字，還把這些法條在圖譜中的「鄰居」（例如：該法條常見的併科罰則、相關的作業類型）一併抓出來。
- **Step 3.3: 圖譜提示生成的推理 (Graph-Prompted Generation)**
  - **操作：** 將 Step 3.2 抓到的資訊，轉化為 LLM 的 Prompt。
  - **Prompt 結構範例：**
    [角色設定] 你是職安法專家。
    [使用者問題] 乙炔外洩爆炸責任歸屬？
    [GNN 推論線索] 系統分析此案件與「高壓氣體勞安規則」有 92% 關聯性。
    [知識圖譜路徑] 關聯路徑：乙炔 -> 易燃氣體 -> 需通風換氣 (設施規則177條)。

[類似判決] 參考案例 INC_5566 (判決書摘要...)。
[任務] 請綜合以上資訊進行法律分析。

- **產出：** 最終報告。

教授的總結：這套流程強在哪裡？

同學，這套整合後的 Workflow 具備了三個層次的「防禦力」，解決了傳統 AI 的痛點：

1. **解決「邏輯不通」：** 透過 **Phase 1 的 RotatE/R-GCN**，模型學會了嚴謹的圖譜結構，不會胡亂連結因果。
2. **解決「只會背書」：** 透過 **Phase 2 的 Hard Negative Mining**，模型學會了分辨細微差異，而不只是背誦法條。
3. **解決「遇到新題不會做」：** 透過 **Phase 3 的 Inductive GNN**，即使是沒看過的全新事故，也能透過鄰居關係推導出可能的違法點。