下面我把你要的三份附錄 **G／H／I** 全文生出來（可直接存為 docs/V19/ 下的三個 .md 檔）。

# **Appendix\_G\_Sparse\_Recognition\_V19.md**

**題名**：少量資料辨識／GNSE 實作規格（V19）  
 **目的**：在**小樣本、低訊噪比**下，仍能穩定提出可檢驗的候選解與「最小可辨識補充」建議。  
 **硬體預設**：RTX 3090 + 一般 CPU（可遷移同級邊緣設備）

## **G.1 觀念總覽**

* **GNSE（Graph-Native Signature Engine）**＝「圖譜原生特徵引擎」
* **三要素**：  
  1. 結構表徵：typed-walk Node2Vec 派生的**結構向量** e\_struct
  2. 語義表徵：文本／圖片的**語義向量**（CLIP/文本嵌入）e\_text / e\_image
  3. 融合打分：任務自適應權重 score = w\_s·S(e\_struct) + w\_t·S(e\_text) + w\_i·S(e\_image) + bias

## **G.2 typed-walk Node2Vec（結構向量）**

### **G.2.1 轉移機率（typed-walk）偽程式**

Input: graph G=(V,E), node\_types Tn, edge\_types Te

walk\_len L, walks\_per\_node R

p[type], q[type] // return/in-out bias by edge-type

α[type\_u→type\_v] // type-conditioned preference

Output: sequences of typed random walks

for each v in V:

repeat R times:

walk = [v]; prev = None; prev\_edge\_type = None

for step in 1..L-1:

cand = neighbors(walk[-1])

probs = []

for u in cand:

e\_type = edge\_type(walk[-1], u)

// Node2Vec p/q bias (second-order) with type-specific params

if prev is None:

bias = 1

else:

if u == prev: bias = 1 / p[e\_type]

else if dist(prev,u)=1: bias = 1

else: bias = 1 / q[e\_type]

// type preference

n\_type = node\_type(u)

bias \*= α[ node\_type(walk[-1]) → n\_type ]

probs.append( (u, bias) )

u\_next = sample\_by(probs\_normalize(probs))

prev = walk[-1]; walk.append(u\_next)

emit all walks

**建議參數**（3090／通用圖譜）

* L=40、R=20；p[·]=1.0（中性返程）、q[·]=2.0（鼓勵探索）
* 對「概念→實例」與「因果→事件」邊型，把 α 稍微>1（如 1.2）以強化結構語義

### **G.2.2 向量學習**

* 工具：gensim word2vec / torch Skip-gram（負採樣 5）
* 維度：dim\_struct=128（預設）
* 覆寫鍵：.env: GNSE\_DIM\_STRUCT, GNSE\_WALK\_LEN, GNSE\_WALKS\_PER\_NODE, GNSE\_P\_JSON, GNSE\_Q\_JSON, GNSE\_ALPHA\_JSON

## **G.3 Δ-Refresh（局部重算）**

### **G.3.1 觸發事件**

* 新增／更新節點或邊（L2 micro-fact，或 L3 重要連結）
* 刪改重導（redirect / supersede）
* 你在駕駛艙執行「重加權」或「結構校正」

### **G.3.2 流程圖（ASCII）**

[變更事件] → [定位影響子圖 K-hop]

→ [產生增量 walks] → [局部embedding學習]

→ [向量庫原地更新] → [寫入審計] → DONE

* K=2..3（預設 2）
* 夜間批次：合併白天增量，跑一次「冷凝調整」（對齊分佈均值）

## **G.4 融合打分（結構 × 語義 × 圖像）**

### **G.4.1 定義**

* **結構相似**：S\_struct = cos(e\_struct(q), e\_struct(k))
* **文本相似**：S\_text = cos(e\_text(q), e\_text(k))
* **圖像相似**：S\_image = cos(e\_image(q), e\_image(k))（無圖像則跳過）
* **任務權重**：w\_s + w\_t + w\_i = 1，由任務（搬運/危害/運輸…）決定（見附錄 H 任務表）

### **G.4.2 融合與校準（Python 伪碼）**

def fusion\_score(q, k, w\_s, w\_t, w\_i, bias=0.0):

s = 0.0

if q.struct and k.struct:

s += w\_s \* cos(q.struct, k.struct)

if q.text and k.text:

s += w\_t \* cos(q.text, k.text)

if q.image and k.image:

s += w\_i \* cos(q.image, k.image)

return s + bias

# 校準（可選）：以小型人工標註對 (q,k,label∈{0,1})

# 估計 logistic 回歸，把 raw\_score → calibrated\_prob

## **G.5 Top-2 分差與「最小可辨識補充」**

* 規則：abs(score1 - score2) < 0.05（5%）→ 回傳**補充建議**而非要你裁示
* 格式：

{ type: "need\_more\_signal",

hints: ["請提供耳朵形狀與鼻口比的照片"],

if\_user\_cannot: "資料不足，無法進一步收斂。",

policy: "禮貌收束，不嘗試臆測" }

## **G.6 快取與資源**

* 以 (query\_hash, index\_version) 作快取 key；GNSE 查詢預設 1200ms 時間箱、共 4 步探索
* Δ-Refresh 與召回查詢分離佇列；RST 失敗自動降級「語義-only」模式

## **G.7 驗收卡（最小）**

* **A-GNSE-1**：typed-walk 能覆現近鄰（人工抽驗 20 筆，相似概念回傳率 ≥ 80%）
* **A-GNSE-2**：Δ-Refresh 對單點變更在 5 分鐘內更新完成
* **A-GNSE-3**：Top-2 <5% 時能回補充建議，且不誤觸推理
* **A-GNSE-4**：融合打分在三任務場景下相較單向量基線提升 ≥ 8%（小測集）

# **Appendix\_H\_Concept\_Spectrum\_V19.md**

**題名**：概念光譜（模糊概念的量化與推理）  
 **目標**：把「雨很大／有點遠／非常重」等模糊語，轉為可計算、可引用的**光譜值**與**區間語義**。

## **H.1 表徵模型**

* **一維概念**：採用\*\*分段線性（Piecewise Linear）\*\*隸屬函數（輕量、穩定）  
  + 以 3–5 個錨點 (x, μ(x)) 定義；μ ∈ [0,1]
  + 類別詞（小/中/大）以**三角形/梯形**隸屬分割
* **多維概念**：各維度各自計算 μ\_i，再以**可換 t-norm**聚合  
  + 預設 **加權幾何平均**：μ = ∏ μ\_i^{w\_i}（穩定、可調權）

不用 KDE/高斯混合，避免 3090/CPU 額外負擔；推理時以錨點插值。

## **H.2 基本 API（伪）**

// 註冊光譜

POST /spectrum/register

{ "name":"rain\_intensity\_tw",

"unit":"mm/h",

"anchors":[ [0,0.0], [5,0.2], [10,0.5], [20,0.8], [50,1.0] ],

"labels": {

"小": { "shape":"tri", "range":[0,8,12] },

"中": { "shape":"tri", "range":[8,18,28] },

"大": { "shape":"trap", "range":[20,35,999,999] }

}

}

// 評分

POST /spectrum/eval

{ "name":"rain\_intensity\_tw", "x": 12 } → { "μ":0.56, "label":"中(0.62)" }

// 多維聚合

POST /spectrum/eval\_multi

{ "specs":[

{"name":"size\_generic","x": 0.7}, // 標準化 0..1

{"name":"weight\_kg","x": 25}

],

"weights":[0.4,0.6] }

→ { "μ":0.73 }

## **H.3 台灣預設（V19-12）**

你剛剛拍板的「台灣光譜預設」落地如下：

### **H.3.1 降雨 rain\_intensity\_tw（mm/h）**

* Anchors: [0,0], [5,0.2], [10,0.5], [20,0.8], [50,1.0]
* Labels: **小**(0-8-12)、**中**(8-18-28)、**大**(20-35-∞)

### **H.3.2 風力 wind\_level\_tw（m/s，對應蒲福）**

* Anchors: [0,0], [5,0.2], [10,0.5], [17,0.8], [25,1.0]
* Labels: **微風**(0-4-6)、**強風**(8-14-20)、**烈風**(17-22-∞)

### **H.3.3 能見度 visibility\_tw（km）**

* Anchors: [0,1.0], [1,0.8], [5,0.5], [10,0.2], [20,0.0]（越小越危險 → 以「清晰度」反向）
* Labels: **低**(0-1-3)、**中**(2-6-12)、**高**(10-15-∞)
* 評分時自動做 clarity = 1 - f(distance)

## **H.4 任務權重表示例（用於融合打分與決策）**

| **任務** | **w\_struct** | **w\_text** | **w\_image** | **備註** |
| --- | --- | --- | --- | --- |
| 搬運（尺寸/重量） | 0.3 | 0.5 | 0.2 | 文字描述較重要；圖像輔助 |
| 危害（天氣/能見） | 0.5 | 0.3 | 0.2 | 結構（歷史關聯/規範）權重高 |
| 運輸（路況/裝載） | 0.4 | 0.4 | 0.2 | 均衡 |

這張表與 GNSE 的 w\_s,w\_t,w\_i 對齊；你可在駕駛艙覆寫。

## **H.5 與推理的銜接**

* L2 Preflight 先做單位與量綱檢查（用 Pint），將數值轉為光譜系統標準單位
* 系統二在多假設排序時，把 μ 作為**軟證據分**的一部份（非決定性），並入審計

## **H.6 驗收卡（最小）**

* **A-SPC-1**：三個預設光譜可正確插值與回標籤
* **A-SPC-2**：多維聚合在缺一維時能自動降級不報錯
* **A-SPC-3**：融合打分能讀取權重表影響排名（小測集可觀）

# **Appendix\_I\_Preflight\_V19.md**

**題名**：L2 公理 Preflight（前置審查）  
 **任務**：在「引擎展開推理」之前，把**不可能／自相矛盾／單位錯配／紅線風險**先擋下。

## **I.1 資料入口與正規化**

* **入口**：Logic\_Axioms/（自然語言公理）、Facts/（候選事實）、Maybe\_right/（評論）
* **管線**：去噪 → 分句 → 模板映射（HORN/EQ/Constraint/Temporal） → 型別/單位抽取 → 形式化
* **禁止硬刪**：不通過者進 needs\_revision 佇列（60 天未動自清）；時間性覆蓋走 SUPERSEDE

## **I.2 形式化模板（精簡集）**

* **等式邏輯（EQ）**：a=c ∧ b=c → a=b；以**同餘閉包**維護等價類
* **命題/Horn**：p(X) ∧ q(X,Y) → r(Y)；用 SAT/ASP 做互斥檢查
* **約束（Constraint）**：單位與量綱一致、溫壓界、地理層級
* **時間（Temporal）**：t\_cause < t\_effect、時間粒度一致

## **I.3 檢查器實作（輕量）**

* **單位**：Pint（量綱與換算）；同欄位需同量綱
* **時間**：dateutil + 自訂粒度欄位（year/month/day…）
* **地理**：鄰接關係表（城市→州/縣→國）；不可跨國雙歸屬
* **因果**：簡單 DAG 方向檢查；倒置即否
* **紅線**：SafetyAxiom（Neo4j 類型：:SafetyAxiom {id, rule, scope}）先行否決
* **等式**：Union-Find + Congruence Closure（小型）

## **I.4 回報結構（JSON）**

{

"ok": false,

"violations": [

{

"rule\_id": "AX\_EQ\_001",

"type": "logic|unit|time|geo|redline",

"span\_or\_fact\_id": "f12345",

"reason": "Equality contradiction via c",

"minimal\_counterexample": ["a=c","b=c","¬(a=b)"]

}

],

"advice": "請將 km 與 m 統一為 m"

}

## **I.5 與 ERR／版本化的銜接**

* **重覆**：DUPLICATE → 合併 source/alias（不增節點）
* **互斥**：新=高權重（E1/E2）→ 舊 RETRACTED 並 redirect\_to 新
* **時間性**：effective\_at 較新者 → 舊 SUPERSEDED
* **審計**：作者/時間/hash/回歸結果必記

## **I.6 性能／資源**

* 主跑 CPU；以 (context\_hash,candidate\_hash,ruleset\_version) 作快取
* 超時降級：只跑「單位＋時間＋等式」三件最要緊的

## **I.7 驗收卡（最小）**

* **A-L2-1**：100 條自然語言公理→可正規化入庫（含 form/vars/type\_sig）
* **A-L2-2**：km 與 m 混算被攔下並建議轉換
* **A-L2-3**：因果倒置能被事後驗證抓回、進仲裁佇列
* **A-L2-4**：紅線 SafetyAxiom 能否決危險輸出
* **A-L2-5**：ERR 橋接：新舊互斥→舊轉向或撤回；重複→合併 alias

### **備註（對工程）**

* 模板與檢查器都以「**可替換開源實作**」為原則；你只要封裝統一接口：  
  + normalize\_rule(text) -> {form, vars, ...}
  + preflight(candidate, context) -> {ok, violations[]}
* 任何拒絕必附「**最小反例**」以便你快速修文或裁示。

這三份就是 V19 需要的 **G/H/I 附錄**。  
 你若要，我接著可以幫你把這三份再生成「最小測資（/tests/acceptance/）」的小檔，或是再補一份 .env.v19.example 骨架。