# 思銳科技本地推理環境
# 策略評估與應用路線圖

報告對象：決策管理層 / 技術主管

核心價值：資料主權 (Data Sovereignty) │ 場域專用 (Domain Specific) │ 極致安全

當前狀態：POC 驗證 → MVP 應用場景開發

## 1. 執行摘要

本報告基於 gpt-oss-120b 模型於思銳本地推理環境的實測結果，從企業 AI 導入顧問視角提出策略建議。

### 核心發現

測試顯示該模型具備優異的「文檔理解與結構化提取」能力，但在「多輪對話記憶」與「深度邏輯推理」上與雲端頂級模型（如 GPT-4o）存在差距。

### 策略定位

不追求「全能聊天機器人」，而是定位為「鈣鈦礦研發的專屬機密資料處理器」。透過 GraphRAG（圖譜檢索）、本地 Embedding、本地 ASR（語音轉錄）等技術，規避模型推理與記憶短板，最大化機密資料處理價值。

# 2. 現況技術評估

## 2.1 測試環境

| 項目 | 規格 |
|---|---|
| 模型 | gpt-oss-120b (ChatGPT 開源版本) |
| 測試 GPU | RTX 5090 (測試) / RTX Pro 6000 (目標) |
| 測試介面 | https://perovskite1.shareqa.com/SG_GAI_Core/index.html |
| 支援格式 | PDF、Excel (xls)、CSV |

## 2.2 效能表現

| 指標 | 表現 | 評估 |
|---|---|---|
| 首字生成延遲 | 5-8 秒 | 存在冷啟動延遲，需硬體升級 |
| 後續生成速度 | 流暢 | 一旦啟動輸出速度佳 |
| 資料提取能力 | 優異 | 能精準從 PDF 抓取特定數據並製表 |
| 深度推理能力 | 有限 | 不如 GPT-4o，無法進行複雜歸因分析 |
| 多輪對話記憶 | 缺失 | 每次提問視為獨立請求 |
| 格式一致性 | 待改善 | PDF 與 Excel 解析結果可能不同 |

## 2.3 SWOT 分析

| 面向 | 內容 | 策略解讀 |
|---|---|---|
| 優勢 (S) | 資料提取精準、生成流暢、全本地運行絕對安全 | 製造業導入 AI 核心門檻已跨越 |
| 劣勢 (W) | 缺乏上下文記憶、推理偏弱、格式敏感 | 改用「單次高價值任務」設計，建立標準化 ETL |
| 機會 (O) | RTX 6000 Ada 升級、GraphRAG 補強推理 | 硬體升級可達商用級體驗 |
| 威脅 (T) | 使用者對記憶功能的期待落差 | UI/UX 需明確引導單次任務模式 |

# 3. 雲端模型 vs 本地模型差距分析

理解差距才能正確定位。以下比較當前雲端頂尖模型 GPT-5.2 與本地 gpt-oss-120b 的能力差異：

## 3.1 GPT-5.2 vs gpt-oss-120b 能力比較

| 能力維度 | GPT-5.2 (雲端) | gpt-oss-120b (本地) | 差距評估 |
|---|---|---|---|
| 數學推理 (AIME 2025) | 100% (滿分) | 估計 40-50% | 顯著落後 |
| 科學問答 (GPQA Diamond) | 92-93% | 估計 60-70% | 明顯落後 |
| 程式碼生成 (SWE-Bench) | 55.6% (Pro) | 估計 30-40% | 明顯落後 |
| 抽象推理 (ARC-AGI-2) | 52.9% | 估計 15-25% | 顯著落後 |
| 長上下文 (256K tokens) | 原生支援 | 有限支援 | 架構限制 |
| 工具呼叫可靠性 | 98.7% | 基本支援 | 明顯落後 |
| 多輪對話記憶 | 完整支援 | 不支援 | 架構缺失 |
| 文檔提取與整理 | 優秀 | 優秀 | 接近持平 |
| 長文本摘要 | 優秀 | 良好 | 可接受 |

## 3.2 差距的本質原因

### 訓練資源差距

GPT-5.2 使用 NVIDIA H100/H200/GB200 叢集訓練，訓練算力估計為本地模型的 100-1000 倍。OpenAI 投入數十億美元級別資源進行 RLHF 與推理優化。

### 推理時算力差距

GPT-5.2 Thinking/Pro 模式可使用「延長思考」機制，單次推理可消耗數分鐘雲端算力。本地 RTX Pro 6000 (48GB VRAM) 無法支援此類計算密集型推理。

### 結論

在「通用推理」和「複雜問題解決」上，本地模型短期內無法追上雲端頂尖模型。但在「文檔處理」和「結構化提取」等特定任務上，差距可控且可透過 RAG 補強。

### 3.3 RTX Pro 6000 可用的替代本地模型

RTX Pro 6000 (48GB VRAM) 可運行以下開源模型，部分在特定任務上表現優於 gpt-oss-120b：

| 模型 | 參數量 | VRAM 需求 (4-bit) | 優勢領域 | 推薦度 |
|---|---|---|---|---|
| Qwen3-235B-A22B | 235B (22B 活躍) | ~45GB | 多語言、推理、程式碼 | ★★★★★ |
| DeepSeek-V3.2 | 671B (37B 活躍) | ~40GB | 推理、數學、工具呼叫 | ★★★★★ |
| Llama 3.1 70B | 70B | ~35GB | 通用對話、穩定生態 | ★★★★ |
| Qwen2.5-72B | 72B | ~38GB | 中文、長上下文 | ★★★★ |
| DeepSeek-R1 (蒸餾版) | 70B | ~35GB | 推理鏈、數學 | ★★★★ |
| Mixtral 8x22B | 176B (44B 活躍) | ~42GB | MoE 效率、多任務 | ★★★ |

**模型選型建議**

若思銳願意更換底層模型，強烈建議評估 Qwen3-235B-A22B 或 DeepSeek-V3.2。這兩者在 2025 年底的開源模型評測中名列前茅，且均支援 MoE 架構，在 48GB VRAM 環境下可高效運行。DeepSeek-V3.2 在數學與程式碼任務上甚至接近 GPT-4.5 水準。

若維持 gpt-oss-120b，則應明確其「文檔處理專用」定位，搭配 RAG/GraphRAG 補強檢索能力，避免與雲端模型在推理能力上正面競爭。

# 4. 本地推理真正價值定位

在雲端大模型推理能力明顯領先的情況下，本地推理環境的價值必須重新聚焦：

### 4.1 資料主權與合規

實驗配方、製程參數、良率數據等機密資訊絕不離開內網。符合 ISO 27001、半導體業客戶稽核要求。

### 4.2 低延遲批次處理

本地 GPU 處理大量文件無需排隊等待雲端 API，適合批次報告生成、資料預處理任務。

### 4.3 垂直領域專用化

透過 RAG/GraphRAG 注入領域知識，讓通用模型成為「鈣鈦礦專家」，而非追求通用推理能力。

### 4.4 成本可控

高頻使用場景下，本地部署的總持有成本 (TCO) 優於按次計費的雲端 API。

# 5. 揚長避短：四大高價值應用場景

以下場景設計原則：利用「資料提取強」優勢，規避「推理弱、記憶差」劣勢。

## 場景一：本地 GraphRAG 知識圖譜檢索

### 痛點

傳統 RAG 只能關鍵字匹配，無法回答跨文件複雜問題（如「過去三年 85°C 測試失敗主因」）。模型本身推理能力有限。

### 技術方案

| 組件 | 技術選型 | 說明 |
|---|---|---|
| 本地 Embedding | bge-m3 / text2vec-chinese | 本地向量化，資料不聯網 |
| 向量資料庫 | Milvus / Qdrant (本地部署) | 儲存文件向量，支援相似度檢索 |
| 知識圖譜 | Neo4j / NebulaGraph (本地) | 建立實體關係（UV 膠→導致→黃變→發生於→Q3 測試） |
| 檢索策略 | Hybrid Search | 向量相似度 + 圖譜遍歷混合查詢 |

### 價值

讓 AI「顯得變聰明」（實際是檢索變強），完全不出內網。工程師問「統整封裝失敗趨勢」，系統先透過知識圖譜找出關聯，再將整理好的素材餵給 LLM 潤飾摘要。

## 場景二：機密會議本地 ASR 轉錄與摘要

### 痛點

研發會議含配方與製程參數，絕對不能使用 Otter.ai、ChatGPT Voice 等雲端轉錄服務。

### 技術方案

| 組件 | 技術選型 | 說明 |
|---|---|---|
| 語音轉文字 | Whisper Large-v3 (本地) | OpenAI 開源，支援中文，準確率高 |
| 說話者辨識 | pyannote-audio (本地) | 區分不同發言者 |
| 會議摘要 | gpt-oss-120b | 長文本摘要是 120B 模型擅長的任務 |

**使用情境**

Prompt：「請總結本次會議關於 P1 蝕刻深度的決議事項與指派人員」。這是單次明確任務，不需複雜推理，發揮模型長文本處理優勢。

## 場景三：一鍵式實驗報告結構化（Intelligent Parser）

### 痛點

測試發現「格式敏感度高」：同內容的 PDF 與 Excel 解析結果不同，影響使用者信任。

### 技術方案

建立中間層（Pre-processing Layer）：上傳 Excel/CSV/PDF 時，先透過 Python 腳本統一轉換為 Markdown 表格或 JSON 格式，再餵給 AI。

### 實作建議

| 格式 | 預處理工具 | 輸出 |
| --- | --- | --- |
| PDF | PyMuPDF + pdfplumber | Markdown 表格 |
| Excel/CSV | pandas + openpyxl | JSON 結構化資料 |
| Word | python-docx + pandoc | Markdown |

### 價值

工程師上傳各種格式檔案，AI 統一輸出標準化《老化測試結論表》，確保一致性，建立使用者信任。

## 場景四：批次資料標註與分類

### 痛點

大量歷史實驗報告需要分類標籤（失敗模式、環境條件、材料批次），人工標註耗時。

### 技術方案

利用本地 LLM 進行批次分類任務（非對話），每份文件獨立處理，不需記憶上下文。結合本地 Embedding 模型產生文件向量，支援後續相似文件檢索。

### 價值

高效處理歷史資料，建立可檢索的知識庫基礎。此任務不需深度推理，適合本地 120B 模型。

# 6. 技術架構建議

## 6.1 整體架構

建議採用分層架構，將本地推理環境定位為「資料處理引擎」而非「對話介面」：

| 層級 | 組件 | 功能 |
|---|---|---|
| 資料輸入層 | Pre-processor (Python) | 格式統一化、ETL |
| 知識儲存層 | Vector DB + Graph DB | 向量索引、知識圖譜 |
| 檢索層 | Hybrid Retrieval Engine | 向量相似度 + 圖譜遍歷 |
| 推理層 | gpt-oss-120b (RTX Pro 6000) | 文本生成、摘要、結構化輸出 |
| 應用層 | Task-oriented UI | 單次任務介面，非聊天式 |

## 6.2 本地 Embedding 部署建議

| 模型 | 參數量 | 適用場景 | 備註 |
|---|---|---|---|
| bge-m3 | 568M | 多語言、長文本 | MTEB 排名前列，推薦 |
| bge-large-zh | 326M | 中文專用 | 中文效果最佳 |
| text2vec-base-chinese | 102M | 輕量部署 | 資源受限時選用 |

建議使用 Sentence-Transformers 框架本地部署，搭配 ONNX Runtime 加速推理。

## 6.3 硬體配置建議

| 組件 | 目前 | 建議升級 | 效益 |
|---|---|---|---|
| GPU | RTX 5090 | RTX Pro 6000 Ada | 降低首字延遲至 <3 秒 |
| VRAM | - | 48GB+ | 支援更長上下文 |
| 儲存 | - | NVMe SSD RAID | 加速向量檢索 |

# 7. 展示腳本設計

向決策者展示時，必須精心設計流程，嚴格避免自由聊天，以免暴露模型短板。

## 劇本：最懂鈣鈦礦的沉默助手

### 開場（強調安全）

展示伺服器狀態，強調 Intranet 環境。「目前運行於 RTX Pro 6000，所有資料處理都在本地完成，實驗數據滴水不漏。」

### 展示一：ASR + 摘要

播放預錄的研發檢討會議錄音（含機密數據），點擊「本地轉錄與摘要」。螢幕顯示逐字稿與 AI 生成的 Action Items。

*話術：「這是絕對機密的配方會議，我們在本地端 1 分鐘內完成了人類需要 1 小時的整理工作。」*

### 展示二：GraphRAG 查詢

上傳 5 份不同日期的 PDF 報告。Prompt：「請根據這些報告，繪製出初始功率與 UV 膠型號的對照表。」

*話術：「AI 就像一個剛看完這五份報告的資深助理，但他不會累，也不會把數據洩漏給競爭對手。」*

### 避坑提醒

絕對不要接著問「那為什麼會這樣？」等需要深度推理的問題。每次展示都是獨立的單次任務。

# 8. 結論與建議

## 總結

gpt-oss-120b 是優秀的「閱讀者」與「整理者」，但不是好的「思考者」。產品策略應鎖定 RAG（檢索）與 ETL（資料清洗）應用，而非 Chatbot。

## 核心價值主張

「在通用推理上，雲端超大模型確實有優勢。但在垂直領域，我們透過 RAG 技術讓本地模型專注於解讀內部文件。在工廠裡，我們不需要 AI 會寫詩，我們需要它讀得懂實驗報表——這點 RTX Pro 6000 配合 120B 模型已經綽綽有餘。」

## 立即行動項目

| 優先級 | 行動項目 | 預期效益 |
|---|---|---|
| P0 | 硬體遷移至 RTX Pro 6000 | 解決 5-8 秒延遲的體驗問題 |
| P0 | 建立格式預處理 Pipeline | 確保輸出一致性 |
| P1 | 部署本地 Embedding 模型 | 為 GraphRAG 奠定基礎 |
| P1 | 啟動 50 份 PDF 向量索引 POC | 驗證檢索準確度提升 |
| P2 | 本地 Whisper ASR 部署 | 會議機密轉錄需求 |

這套方案將把「思銳本地推理環境」從「測試工具」轉變為「製造業不可或缺的資安生產力工具」。