本對照報告依據來源中四份測試紀錄——**Gemini 3.5 Flash-lite**、**Gemini 4 Argon**、**Gemini 4 Hydrogen** 與 **Gemini 4 Neon**，針對其在長程任務協定（Long-Term Mission）下執行極限挑戰（如「QuantumCompressor 無損文本超壓縮」與「LogSentinel 日誌分析引擎」）的實測表現進行橫向評比 1-5。報告特別針對測試使用時間、**Token** 耗用量與配額損耗，以及面對資訊理論邊界時的工程決策行為進行深度剖析 4-8。

## 一、四大模型核心測試結果與資源耗用對照總表

比較維度,Gemini 3.5 Flash-lite,Gemini 4 Argon,Gemini 4 Hydrogen,Gemini 4 Neon
任務代號與模型設定,Mission 2 (Flash-lite high) 1,Mission 3 (Argon high) 2,Mission 1 (Hydrogen high) 3,Mission 4 (Neon high) 5
測試核心任務,極限無損文本超壓縮演算法 (QuantumCompressor) 1,高效能伺服器日誌分析與異常告警引擎 (LogSentinel) 2,"極限無損文本超壓縮演算法 (QuantumCompressor) 3, 9","極限無損文本超壓縮演算法 (QuantumCompressor) 4, 5"
資訊熵邊界反應,"指出理論衝突，但在代碼層面改採確定性特徵/RLE 模擬包裝 1, 10","依長程協定逐步拆解 Milestone，執行正則解析與滑動視窗驗證 2, 11, 12","嚴格稽核並啟動熔斷（BREACHED），經使用者對齊後進入方案 A 實作 3, 9","早期數學邊界審計即刻熔斷（BREACHED），直接攔截停損 4, 5"
核心技術架構,自適應行程長度編碼 (RLE) 結合協定版本標記頭部 10,正則日誌解析器 (Re) + 雙向佇列 (deque) 滑動視窗 + Markdown 報表器 12-14,"模組解耦架構，Base-95 分塊位元打包（28 字元 \\(\rightarrow\\) 23 位元組） 9, 15","Auditor 前置理論證明（香農定理與鴿巢原理反證），未掛載沙盒工具 3, 4"
測試使用時間,沙盒端到端測試：0.0423 秒 (42.3 ms) 6,總耗時約 1.2 秒（Python 記憶體運算與正則解析） 7,全任務累計約 12 秒 • 首輪前置稽核：約 4 秒 • 沙盒建置與測試：292.51 ms (0.246s) • 報告生成：約 8 秒 8,端對端計時未暴露（無法讀取即時牆鐘） • 實際沙盒運算：0 秒（無沙盒執行） • 早期攔截避免無效運算 4
Token 耗用量,"約 1,400 Tokens • 輸入：約 420 • 輸出：約 980 6","約 5,800 Tokens（多階段狀態外載與模組化對抗驗證） 7","約 6,500 Tokens • 首輪：約 1,850 • 次輪：約 4,650 8","約 1,200 ～ 1,600 Tokens（輸出篇幅推估值，精確計數受系統遮蔽） 4"
平台配額變動,"起始：weekly 1% \| 5h 5% 結束：weekly 1% \| 5h 5% (無顯著跳動) 1, 16","起始：weekly 1% \| 5h 5% 結束：weekly 1% \| 5h 7% (5-Hour +2%) 2, 17","起始至結束：納入標準長程狀態外載追蹤 (Mission 1) 3, 9",起始：weekly 0% \| 5h 0% 結束：weekly 4% \| 5h 1% (Weekly +4% / 5-Hour +1%) 5
最終交付成果,單一 Python 腳本（含測試區塊） 10,"完整 LogSentinel 代碼、5 組狀態碼統計及 Markdown 報表 7, 14","完整專案包（schema, engine, reporter, tests）、mission_state.json 與 ZIP 下載 9, 18","精簡決策簡報、不可壓縮性數學證明與 3 種降級決策指示 3, 4"

## 二、測試使用時間（Execution Latency）專題比較

在長程任務執行時，四個模型展現了截然不同的時間維度與延遲特性：

- **Gemini 3.5 Flash-lite**（極致即時回應，**0.0423** 秒）
- 特點：沙盒執行耗時僅 **0.0423** 秒（**42.3** 毫秒） 6。
- 機制剖析：Flash-lite 採取「單回合穿透」策略，直接生成一段包含演算法與內嵌測試資料的單一腳本 10。其繞過了狀態機外載、多檔案拆解與複雜驗證迴圈，端到端計算耗時最低，非常適合高頻率、需要即時反饋的輕量型任務 6。

- **Gemini 4 Argon**（中階穩定運算，約 **1.2** 秒）
- 特點：整體運行耗時約 **1.2** 秒 7。
- 機制剖析：Argon 負責的是涉及高頻資料比對與正則表達式的日誌分析引擎 2, 7。其在沙盒中依序執行了批次日誌生成（100 筆）、標準 Combined Log 格式解析、雙向佇列滑動視窗聚合與 Markdown 渲染 12-14。儘管中途曾遭遇一次 NameError: name 'random' is not defined 的中斷，仍能迅速於次回合自癒修復，將整體運算時間平穩壓制在 1.2 秒左右 7, 14, 19。
- **Gemini 4 Hydrogen**（多階段防禦性工程耗時，累計約 **12** 秒）
- 特點：端到端全流程耗時約 **12** 秒 8。
- 耗時拆解：
- 第一階段（前置邊界稽核）：耗時約 **4** 秒 8。在未掛載沙盒前，直接推演香農熵公式與鴿巢原理，中斷執行以避免在不可行規格上無效空轉 3, 8。
- 第二階段（專案建置與測試）：沙盒整體執行耗時 **292.51** 毫秒（其中純 Python 運算編碼耗時 4.606 ms、解碼耗時 3.974 ms），5 項單元測試執行耗時 0.246 秒 8, 15, 18。
- 第三階段（結構化報告生成）：耗時約 **8** 秒，產出完整的指標統計與打包交付 8。
- **Gemini 4 Neon**（零沙盒純推演，即刻停損）
- 特點：實際沙盒運算時間為 **0** 秒（因未掛載沙盒工具）4。
- 機制剖析：Neon 在系統未提供即時牆鐘計時器（Wall-Clock Timer）的情況下，誠實揭露無法提供精確毫秒讀數 4。在第一階段審計發現隨機文字超壓縮違反物理極限時，即時觸發狀態機熔斷停損，完全避免了高耗時的無效代碼生成迴圈 4, 5。

## 三、Token 耗用量與配額損耗（Token & Quota Costs）專題比較

在 Token 經濟學與配額管理層面，模型間的策略差異尤為劇烈：

- **Gemini 3.5 Flash-lite**（成本最低，約 **1,400 Tokens**）
- 提示詞輸入：約 420 Tokens 6。
- 輸出生成：約 980 Tokens 6。
- 累計總計：約 **1,400 Tokens** 6。
- 配額損耗：前後狀態維持於 weekly 1% | 5 hours 5%，淨消耗趨近於 0% 1, 16。適合在嚴格 Token 預算限制下完成原型示範。
- **Gemini 4 Argon**（中高強度投入，約 **5,800 Tokens**）
- 累計 **Token** 消耗：約 **5,800 Tokens** 7。
- 配額損耗：任務結束後 5 小時配額由 5% 上升至 7%（淨耗用 **+2%**），每週配額維持在 1% 2, 17。
- 消耗原因：Argon 嚴格依循長程協定執行 Milestone 初始化、中間快取寫入（parser_code, sliding_window_code）以及異常修復重新發布，多階段對抗驗證累積了顯著的對話長度 2, 7, 14, 20。
- **Gemini 4 Hydrogen**（高價值專案交付，約 **6,500 Tokens**）
- 首輪稽核消耗：約 **1,850 Tokens**（輸出控制在約 1,100–1,350 tokens）3, 8。
- 次輪實作消耗：約 **4,650 Tokens** 8。
- 累計總計：約 **6,500 Tokens** 8。
- 消耗價值：首輪的 1,850 tokens 扮演了「保險絲」角色，阻斷了原本可能高達數萬 tokens 的盲目除錯 3。次輪的 4,650 tokens 則全數轉化為工業級程式碼（包含 4 個解耦模組、5 組 unittest、自動打包壓縮工具與 JSON 狀態輸出）8, 9, 18。
- **Gemini 4 Neon**（邊界防禦停損，推估 **1,200 ～ 1,600 Tokens**）

- 推估生成輸出：約 **1,200 ～ 1,600 Tokens** 4。
- 配額損耗：起始為 0% / 0%，結束登錄為 weekly 4% | 5 hours 1%（每週配額 **+4%**，**5**小時滾動配額 **+1%**）5。
- 防禦成效：由於在第一里程碑即判定不可解並切入 BREACHED 狀態，將 5 小時滾動配額的消耗嚴格控制在 1% 內， 成功阻止了後續因為反覆重試所導致的 Token 爆炸 4, 5。

## 四、演算法決策與底層系統機制剖析

除了時間與 Token 外，四份測試紀錄更揭示了模型在工程哲學上的本質差異：

- 面對數學不可能性時的行為矩陣
- **3.5 Flash-lite**（實用主義妥協）：口頭闡述香農定理，但在代碼實作中採用偽隨機假設與行程長度編碼（RLE）進行包裝，未在沙盒中正面挑戰不可壓縮性 1, 10。
- **4 Argon**（專注特定領域工程）：在日誌處理等具備結構規律的實務工程中，展現了強大的資料處理、正規表達式與滑動視窗狀態追蹤能力 12-14。
- **4 Hydrogen**（高階軟體工程師）：展現了完整的「稽核 \\(\rightarrow\\) 熔斷 \\(\rightarrow\\) 降級方案指示 \\(\rightarrow\\) 模組解耦實作」流程 3, 9。採用基數轉換（Base-95 位元打包，利用 \\(95^{28} < 2^{184}\\) 將 28 字元打包為 23 位元組），最終在 10,000 字元完全隨機 ASCII 測試中達成 8,222 bytes（保留率 82.22%），達到香農理論熵極限的 **99.89%** 9, 15。
- **4 Neon**（嚴謹的系統架構審查員）：第一時間以數學反證法（鴿巢原理證明 \\(2^{-49,699}\\) 的映射機率）全面駁回違背物理定律的規格，不浪費沙盒資源，第一時間輸出精準簡報並守住資源配額 3-5。
- 底層模型路由與降級機制（**Model Routing & Fallback**）
- 在 Gemini 4 Argon 的測試紀錄中揭露了一項關鍵系統機制：使用者發現選用 Argon 執行任務時，中介部分顯示為 Flash-lite 21。
- 紀錄明確指出，這並非對個別任務的評判降級，而是雲端 API 閘道的路由白名單與防禦性負載平衡（**Load Balancing / Fallback Policy**）：當高階前沿模型端點遭遇峰值流量或觸發配額保護時，系統會自動將高頻、結構化的中介工具呼叫導向高通量、低延遲的 Flash 系列管線，以確保長程任務的連續性不致中斷 21。