TCCS v1.5｜Threaded Cognitive Context System

一種語義記憶分線架構的全面提案

# 1. 緣起與設計哲學

TCCS（Threaded Cognitive Context System）源自於對當代大型語言模型（LLMs）在記憶、上下文一致性、與知識延續性方面的根本挑戰的深度反思。 它非單一模組，而是一種重新架構語言模型長期互動與記憶邏輯的思想體系，目標是讓系統如同人類般能分線記憶、語義喚起、遺忘非必要經驗。

# 2. 架構總覽

- Thread Units：每段對話或任務為一個語義執行緒，具備 summary、intent、timestamp、linked\_threads。  
- Semantic Index Layer：將 threads 依主題語義、自我投射、任務目標進行組織。  
- Trigger Engine：判斷當前輸入是否應喚起某一或多個 thread。  
- Memory Router：根據任務目標動態拼裝 context，而非依時間序列。  
- 冷記憶池（Cold Memory Pool）：權重低但備查重要性的 thread 儲存區，模仿潛意識概念。

# 3. 延伸模組設計

- 動態權重：每一 thread 根據互動密度、關聯強度自動升降。  
- 語意反饋：AI 針對模糊召回主動提出反問，如『你是指哪一次…』  
- Thread Audit：召回過程記錄，供追蹤與 debug。  
- 人格建模模組：根據對話歷程建構用戶價值觀與偏好線索，協助決策建議。

# 4. 類比與類人認知模型比較

TCCS 的冷記憶設計對應人類潛意識，語意索引模擬人類語義場連結。與現今 transformer 記憶方式不同，TCCS 採事件線、主題線、角色線三線並存方式建構語意層級記憶體系。

# 5. 應用情境總覽

- 分散式 AI：跨代理 thread 傳遞與共構。  
- 生成式 AI：提升長篇生成上下文穩定度。  
- 可信任 AI：可追溯 thread 來源與召回鏈，提升透明性。  
- 永續 AI：避免遺忘核心資料或強化非預期錯誤記憶。

# 6. 潛在風險與對策

- 記憶污染：thread 設有來源驗證與 checksum。  
- 語意偏誤：語意反饋機制降低歧義。  
- 過召回：可由用戶選擇是否套用 thread。  
- 誘導性建構：每次召回記錄 thread\_id 與源頭，透明追蹤。

# 7. 公開釋出聲明

本提案屬於架構思想釋出，目的為為未來 AI 系統提供參照架構或靈感基礎，歡迎任何研究者、工程師、開源社群自行實作或擴展。