國立臺灣大學工學院工程科學及海洋工程學系

碩士論文

Department of Engineering Science and Ocean Engineering

College of Engineering

National Taiwan University

Master's Thesis

設計任務操作歷史檢索機制以增強基於大型語言模型的網頁瀏覽代理人的效能

Designing a Task Action History Retrieval Mechanism to Enhance the Performance of LLM-Based Web Navigation Agents

林敬翔

Ching-Hsiang Lin

指導教授: 黃乾綱 博士

Advisor: Chien-Kang Huang Ph.D.

中華民國 114 年 3 月

March, 2025

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

謝辭

感謝一路走來，我所遇到的人事物。

感謝爸媽對我的信任，讓我在人生的道路上可以沒有太多後顧之憂的試錯，也讓我學會了什麼叫做無私的愛。感謝靖雯，總是在我沮喪的時候給我打氣，給我靈感，並且不停地陪著我。感謝弟，和我討論很多有趣的議題，一起在探索世界的旅途中教學相長。感謝姐，讓我學會在面對自己熱愛的事物前，永遠可以理直氣壯。感謝網路的力量，更感謝 LLM 領域的先賢壯士，是你們的努力讓我有了這一篇論文。感謝實驗室的各位，讓我在研究所的生活中有一群一起奮鬥的戰友。感謝我在歐洲所交到的朋友，讓我了解每個人都有自己的時區。感謝口委張瑞益教授以及馬尚彬教授的指導，提供了專業的意見，點出了論文可以修正的方向，讓我見識到了自己的不足。感謝黃乾綱教授教會我什麼是做研究的態度、什麼是客觀、什麼是邏輯、什麼是一個學者應該具備的高度。

當然，也要感謝總是堅持和別人走不同道路的自己，雖然偶有顛簸，但這樣的顛簸很美。雖然常常被他人質疑，但我絕不後悔我所做的種種決定

因為需要感謝的人太多了，就感謝天罷。一直以來都很希望把這句話用上，也終於等到這個時候了。所以，最後當然也要感謝陳之藩。

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

摘要

隨著大型語言模型（Large Language Models, LLMs）的迅速發展，基於 LLM 的人工智慧代理人（AI Agents）在網頁瀏覽任務（Web Navigation Tasks）中展現出高度潛力。然而，如何有效整合過往經驗以提升代理人的決策能力與泛化表現，仍是一項待解決的挑戰。

在過去的作法中，Synapse 提出軌跡做為範例（Trajectory-as-Exemplar, TaE）機制，透過語意相似性檢索與任務描述相近的範例作為提示；代理人工作流記憶機制 (Agent Workflow Memory, AWM）則進一步整合常見工作流作為提示使用。然而，AWM 僅依任務所處環境檢索資訊，容易導致檢索到與當前任務無關的資訊，限制效能提升。

本研究提出記憶整合式相似性機制（Memory-Integrated Mechanism with Similarity, MIMS），在 AWM 架構上結合動作目標預測與工作流的語意相似性檢索。任務執行時，MIMS 首先會根據任務描述、歷史軌跡與觀察資訊預測當前動作目標（Action Objective），再從向量資料庫中檢索與該目標最相近的工作流，並納入 LLM 的提示上下文中，輔助動作預測與任務執行。

在 Mind2Web 資料集上的實驗顯示，MIMS 在 Cross-Task 任務中相較於 Synapse 與 AWM 有顯著提升，於 Element Accuracy（EA）、Step Success Rate（SSR）與 Success Rate（SR）分別達到 40.6、37.0 與 5.1。雖然在 Cross-Website 與 Cross-Domain 任務中的表現與現有方法相近，但本研究展現了基於語意相似性的工作流檢索在網頁瀏覽任務中的潛力。

關鍵字：大型語言模型、網頁瀏覽、代理人

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

Abstract

With the rapid advancement of large language models (LLMs), LLM-based AI agents have demonstrated significant potential in web navigation tasks. However, how to effectively leverage prior experience to enhance agents' decision-making and generalization capabilities remains an open challenge.

Among prior approaches, Synapse proposed the Trajectory-as-Exemplar (TaE) mechanism, which retrieves semantically similar trajectories as prompts based on task descriptions. The Agent Workflow Memory (AWM) framework further incorporates common workflows as prompt components. Nevertheless, AWM relies solely on environment-based retrieval, which often results in selecting contextually irrelevant information, thereby limiting performance gains.

To address this issue, we propose a Memory-Integrated Mechanism with Similarity (MIMS), which extends AWM by incorporating action objective prediction and semantically guided workflow retrieval. During task execution, MIMS first predicts the current action objective based on the task description, trajectory history, and current observation. It then retrieves the most semantically relevant workflows from a vector database and incorporates them into the LLM' s prompt context to support action prediction and task completion.

doi:10.6342/NTU202500827

Experiments conducted on the Mind2Web dataset show that MIMS significantly outperforms Synapse and AWM in Cross-Task settings, achieving 40.6 in Element Accuracy (EA), 37.0 in Step Success Rate (SSR), and 5.1 in overall Success Rate (SR). While the performance in Cross-Website and Cross-Domain settings is comparable to existing methods, our results highlight the potential of semantically driven workflow retrieval in enhancing LLM-based agents for web navigation tasks.

Keywords: Large Language Model, Web Navigation, Agent

doi:10.6342/NTU202500827

目次

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

圖次

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

表次

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

專有名詞與縮寫對照表

縮寫	英文全稱	中文對照	講解頁數
LM	Language Model	語言模型	1
LLM	Large Language Model	大型語言模型	1
ICL	In-Context Learning	上下文學習	1
SaaS	Software as a Service	軟體即服務	7
TaE	Trajectory-as-Exemplar	軌跡作為範例	14
AWM	Agent Workflow Memory	代理人工作流記憶機制	16
WF	Workflow	工作流	16
EA	Element Accuracy	元素準確率	21
Op. F1	Operation F1 Score	操作 F1 分數	22
SSR	Step Success Rate	步驟成功率	22
SR	Success Rate	任務成功率	22
MIMS	Memory-Integrated Mechanism with Similarity	記憶整合式相似性機制	24

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

第一章　緒論

1.1　LLM 於網頁瀏覽代理任務中的背景與研究目標

語言模型（Language Model, LM）一直是人工智慧領域的重要研究方向，其應用涵蓋翻譯、語意分析與知識問答等領域。其中，大型語言模型（Large Language Model, LLM）因擁有更龐大的參數量，相較於小型 LM 能夠學習並儲存更多知識，展現出更強的推理能力與廣泛的應用潛力。

OpenAI 於 2022 年推出的 ChatGPT[5]，基於 Transformer 架構 [6]，展現了卓越的自然語言處理能力。隨著 ChatGPT 的問世，人們逐漸發現 LLM 在日常生活與工作中的潛力。只需將任務以文字描述為一組提示詞（Prompt），LLM 便能理解任務並做出回應。此外，LLM 可透過上下文學習（In-Context Learning, ICL）提升推理與執行效能，並無須為每個任務進行額外微調（Finetuning），大幅降低開發與部署成本。這些特性使得 LLM 的應用不再局限於研究領域，任何擁有電腦與網際網路的使用者皆可輕鬆存取並利用其能力。在此背景下，相關應用迅速發展，各種基於 LLM 的技術如雨後春筍般湧現。

於網頁瀏覽（Web Navigation）情境中的人工智慧代理人（AI Agent）為 LLM 應用中的重要研究方向之一，諸如寄送 email、訂購車票等皆屬於相關範疇。為完成任務，系統需設計合適的提示詞，引導 LLM 做出正確回應。然而，多數傳統方法僅依賴靜態資訊，忽略了模型與環境之間的互動資訊對推理結果的影響。

為改善此問題，Synapse 系統提出 Trajectory-as-Exemplar（TaE）機制。該方法透過在 Prompt 中加入環境與模型之間的互動軌跡 (Trajectory)，以增強 Agent 的理解能力與任務執行效果。此外，Synapse 還基於「語意相似性」設計了一套檢索機制，使 Agent 能夠透過檢索與當前任務目標或環境相似的 Trajectory 來進一步提升其表現。

doi:10.6342/NTU202500827

在 Synapse 的基礎上，[4] 提出了代理人工作流記憶機制 (Agent Workflow Memory, AWM)，將多筆任務軌跡進行歸納，生成可重複使用的共同工作流（Common Workflows），並根據任務環選取合適的 Workflow 作為提示。然而，由於其機制的限制，Agent 所檢索到的 Workflow 雖然在環境上和當前任務相關，但實際內容可能和當前情境無關，甚至可能造成 Agent 表現下降。

從 Synapse 的研究結果可觀察到，當檢索的 Trajectory 描述與當前任務描述之間的語意差距增大時，其檢索機制對 Agent 表現的提升效果也隨之下降。甚至，在使用 CodeLlama[7] 作為 LLM 時，基於任務描述相似度進行檢索的機制可能會導致 Agent 表現下降。

另一方面，我們發現 AWM 所提出的 Workflow 檢索機制存在侷限。由於其僅依據任務所屬環境進行檢索，可能選出與當前任務語意無關的 Workflow，進而干擾 Agent 判斷，影響整體表現。

基於上述觀察，我們認為若能設計一套能夠根據語意相似度進行 Workflow 檢索的機制，將有助於提升 Agent 人在陌生環境下的任務執行能力，並改善現有系統面臨的困境。

1.2 方法概述

本研究提出一個新架構，稱為記憶整合式相似性機制（Memory-Integrated Mechanism with Similarity, MIMS）。該架構結合 Synapse 所使用的基於「語意檢索」的 TaE 檢索機制，並搭配我們提出的基於「語意檢索」的 Workflow 檢索機制。

透過預測 Agent 當前的動作目標，並以此作為查詢基準，MIMS 能從向量資料庫中檢索與動作目標語意最相近的 Workflow，並納入 Prompt 作為決策參考。

1.3 研究貢獻

本研究的主要貢獻如下：

1. 提出一種結合動作目標預測與語意相似性檢索的 Workflow 機制，使 Agent 能夠根據語意關聯性選取 Workflow。

doi:10.6342/NTU202500827

2. 將 Workflow 檢索與 Synapse 的 Trajectory-as-Exemplar 機制整合，並透過參數調整，在 Mind2Web[2] 資料集上於 EA、SSR、SR 指標皆優於 Synapse 與 AWM。

1.4 章節安排

本論文章節安排如下：第二章：介紹網頁瀏覽代理人相關資料集與現有代表性系統。第三章：說明本研究方法，包含 Synapse 與 AWM 的分析，以及 MIMS 系統的設計與架構。第四章：探討 MIMS 在不同參數設定下於 Mind2Web 任務中的表現，並與 Synapse 以及 AWM 進行比較。第五章：總結本研究成果，並討論未來可能的研究方向。

doi:10.6342/NTU202500827

doi:10.6342/NTU202500827

第二章　相關文獻探討

本章旨在釐清 LLM-Based Agent 在網頁瀏覽任務中的發展脈絡與技術挑戰，作為本研究方法設計之依據。首先，我們回顧 AI Agent 的發展歷程，並聚焦於基於大型語言模型代理人的應用潛力。接著，說明目前最具代表性的網頁瀏覽任務資料集 Mind2Web，並分析其任務設計與泛化評估機制。最後，我們深入探討三個被廣泛應用於網頁瀏覽任務的系統：MindAct、Synapse 與 AWM，理解其檢索邏輯與生成機制，作為後續系統改良與實驗設定的參考依據。

2.1　人工智慧代理人（Artificial Intelligence Agent, AI Agent）

根據 [8]，在人工智慧（AI）領域，Agent 指的是一種能夠透過感測器感知環境、進行決策，並透過執行動作來回應環境變化的智能系統。換言之，AI Agent 是一種能夠自動與環境互動，以完成指定任務的 AI 系統。

自古以來，人類不斷尋求提高工作效率的方法。自 AI Agent 概念提出以來，研究人員和企業致力於發展 AI Agent，以實現自動化與智能化。例如，Siri[9]、Alexa[10] 及微軟的 Cortana[11] 均為 AI Agent 的經典應用。

根據 [1]，AI Agent 在任務自動化方面的發展可概括為以下四種模式：

1. **Template-based Programming**：在此模式中，函數會被預先定義，執行時，Agent 會將使用者指令映射到最相關的模板，並依據預設步驟完成任務（如圖 2.1）。

   開發新自動化功能時，工程師需參照官方 API 文件撰寫模板（例如 Google Assistant API[12]、SiriKit[13]）。此模式具備高穩定性，因所有模板皆已預先

doi:10.6342/NTU202500827

定義，但在遇到新任務時，須重新撰寫模板，導致缺乏彈性，進而影響可擴展性。

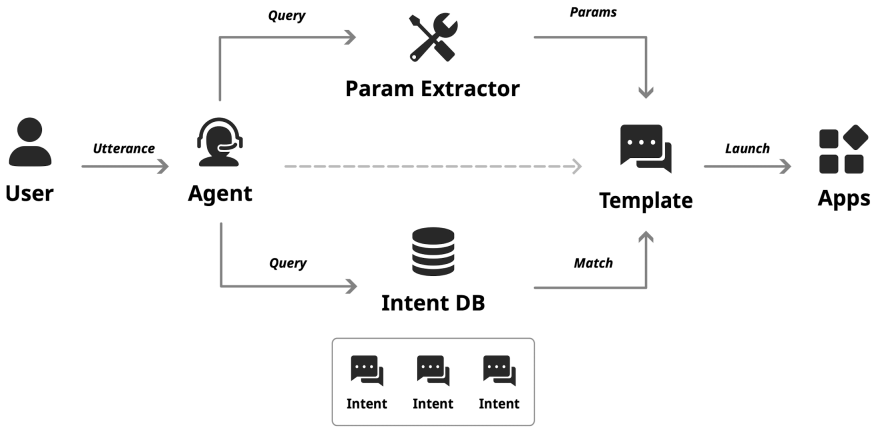

圖 2.1: Template-Based Programming[1]

2. **Supervised Learning Methods**：該方法透過輸入當前狀態（如 GUI 畫面、用戶指令等）來預測下一個最適合的動作。主要研究方向包括學習 GUI 元素的語意資訊、互動行為的建模，以及如何提升模型對不同任務的適應能力。

   相較於 Template-based Programming，此方法具備更大的彈性，能夠適應變化較快的使用情境。然而，模型訓練仍高度依賴大量人工標記的示範數據，且在未見過的任務或介面上，泛化能力（Generalization Ability）有限。

3. **Reinforcement Learning Methods**：透過持續與環境互動來學習任務，不需大量訓練樣本。然而，此方法面臨以下挑戰：

   - 獎勵函數（Reward Function）難以定義。
   - 行動空間（Action Space）龐大，增加學習難度。
   - 泛化能力（Generalization Ability）較低，與 Supervised Learning 方法面臨相同問題。

4. **Early Adoption of Foundation Models**：根據 Scaling Law[14]，語言模型的表現隨著參數規模增加而提升。訓練大型語言模型時，通常使用大規模開放領域文本資料（large-scale open-domain text data）進行無監督學習，隨後透過 Instruction Fine-tuning 及 Reinforcement Learning with Human Feedback（RLHF）來調整模型的對齊度（Alignment）並提升效能。ChatGPT 即為此類模型的典型代表。儘管此類模型的訓練與維護成本高，但一旦完成訓練，即可持續應用，並具備強大的泛化能力。

doi:10.6342/NTU202500827

2.2 大型語言模型代理人 (Large Language Model-Based Agent, LLM-Based Agent)

自 ChatGPT[5]、Claude[15] 等 LLM 出現後，使用者不再需要自行部署模型，而是可以透過 AI 公司提供的軟體即服務（Software as a Service, Saas）平台直接存取 LLM 服務。這一變革使得 AI 技術更加普及，人們逐漸發現 AI 在日常生活與工作中的潛力。

在此基礎上，LLM-Based Agent 逐漸受到關注。同時，學者發現，透過為 Agent 添加額外的模組，能使其執行超越語言處理的任務。例如，ChatGPT Plugin 及 Bing AI 皆屬於此類應用。

2.2.1 LLM-Based Agent 於網頁瀏覽任務中的應用

根據 [8]，網頁瀏覽任務（Web Navigation）是指在網路環境中，代理使用者完成特定目標的一類任務。例如，填寫表單、進行網路購物或寄送電子郵件等，皆可視為網頁瀏覽任務的應用場景。

為有效執行此類任務，Agent 不僅需具備理解複雜網頁場景中指令的能力，還必須能夠靈活應對環境變化，以滿足使用者需求。

學者發現，只要透過良好的提示詞設計，就能增強 LLM 對 HTML 的閱讀以及理解能力，使其能夠面對複雜且高度變動的環境。這項特性恰好符合 Agent 應具備的核心能力，進一步提升 LLM-Based Agent 於網頁瀏覽任務中的潛力。

2.3 網頁瀏覽任務資料集

[2] 指出，網頁瀏覽代理人應能在各類型網站中穩定運作，具備解析複雜且雜訊繁多網頁內容的能力，並能執行多樣且細緻的互動操作。為了準確評估 Agent 在上述情境中的表現，資料集本身必須涵蓋豐富的任務類型與環境特徵，才能有效檢驗其泛化與實用能力。

doi:10.6342/NTU202500827

2.3.1 網頁瀏覽任務類型

網頁瀏覽任務依據其與訓練資料的差異程度，可以劃分為三種類型：Cross-Task、Cross-Website 與 Cross-Domain，用以評估 Agent 在不同層次的泛化能力。

在 Cross-Task 任務，任務來自與訓練資料集中相同的網站，但為不同任務，旨在評估 Agent 在面對相同網站中未見過任務時的泛化能力。例如，在訓練資料集中，有一項任務為在 United Airlines 上執行：「Find one-way flights from New York to Toronto.」而在測試時，Agent 需要解決在同一網站上的另一項任務：「Search receipt with the eTicket 12345678 for the trip reserved by Jason Two.」

在 Cross-Website，任務來自與訓練資料集中相同領域，但來自訓練資料中未曾出現過的網站，旨在評估 Agent 在面對陌生網站時的泛化能力。例如，在訓練資料集中，有一項任務為在 United Airlines 上執行：「Find one-way flights from New York to Toronto.」而在測試時，Agent 需要解決來自相同領域、但訓練資料中未曾出現的網站 American Airlines 的任務：「Find a flight from Chicago to London on 20 April and return on 23 April.」

在 Cross-Domain，任務與訓練資料來自不同領域，即測試任務的網站與任務內容皆未曾出現在訓練資料中，旨在評估 Agent 在全新應用場景下的泛化能力。例如，在訓練資料集中，有一項任務為在 WebMD 上執行：「Search for the interactions between ibuprofen and aspirin.」而在測試時，Agent 需要解決來自不同領域、未見過網站 DMV Now 的任務：「Open page to schedule an appointment for car knowledge test.」

2.3.2 常見網頁瀏覽任務資料集

根據我們的分析，WebArena[16] 雖然涵蓋六個領域，但每個領域僅選擇單一網站進行測試，因此無法評估 Agent 在執行 Cross-Website 任務時，是否能維持其表現。WebShop[17] 則僅專注於購物網站，且任務類型僅限於產品購買，其網頁內容也過於簡化，無法有效檢測 Agent 在真實網站上的執行能力。至於 MiniWob[18]，其任務情境過於單純，並非針對特定網站設計，而是專注於低層次的操作任務，如輸入時間或點擊指定按鈕等，和真實瀏覽情境更是差距甚大。

根據表2.1的比較結果，我們認為相較於其他資料集，Mind2Web[2] 不僅更適

doi:10.6342/NTU202500827

合檢測 Agent 在面對陌生領域時的適應能力，還能有效評估其在真實網頁環境中的思考與執行能力，因此更具實用性與研究價值。

表 2.1: Web Navigation 資料集比較表

	Mind2Web	WebArena	WebShop	MiniWob++
領域	5/31	4	1	-
網站數	137	4	1	100
網站類型	真實	真實	簡化	簡化
任務層次	高	高	高	低

2.3.3 Mind2Web：用於泛化能力評估的資料集

Mind2Web[2] 於 2023 年提出，是第一個專為評估能夠在任何網站上執行語言指令以完成複雜任務的 LLM-based Agent 而設計的資料集。該資料集為靜態資料集，主要儲存網站快取內容（Web Cache）。

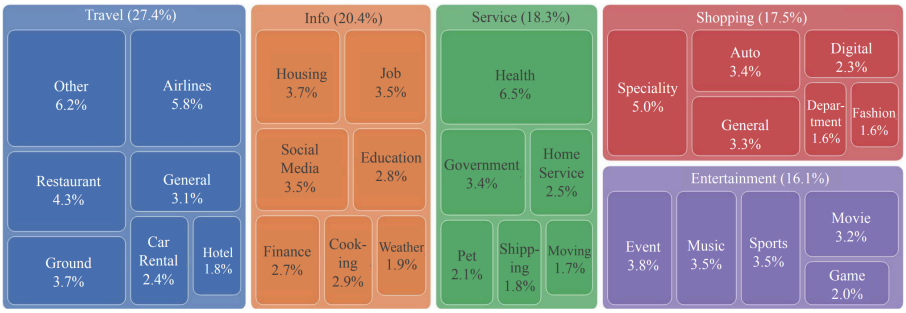

圖 2.2: Mind2Web 資料分布 [2]

如圖 2.2 所示，Mind2Web 資料集共包含 2,350 個任務，涵蓋五大領域（Domains）與 31 個子領域，涉及 137 個真實網站。該資料集具有數項特點：第一，任務設計反映真實世界中多樣且實際的使用者案例；第二，所有操作皆建立於真實網站上，而非模擬環境；第三，資料設計可支援透過不同的任務與環境設定，評估模型於未見情境下的泛化能力。此外，為評估代理人的理解與自動化能力，任務指令多以高階語意描述呈現，而非逐步列出具體操作指令。

2.3.3.1 子資料集分類

Mind2Web 資料集分為訓練集與測試集，且為評估代理人在面對不同泛化挑戰時的表現，測試集進一步被劃分為三個子集：Cross-Task、Cross-Website 與

doi:10.6342/NTU202500827

Cross-Domain，以測試 Agent 在面對陌生環境的能力。各子資料集的詳細資訊列於表 2.2 中。

表 2.2: Mind2Web 子資料集分類

資料集	子資料集	任務數	任務描述	範例網站	範例任務
訓練	-	1009	-	United Airlines (Travel 領域)	Find one-way flights from New York to Toronto.
測試	Cross-Task	252	任務來自與訓練資料集相同的網站，旨在評估 Agent 在面對陌生任務時的泛化能力。	United Airlines (Travel 領域)	Search receipt with the eTicket 12345678 for the trip reserved by Jason Two.
	Cross-Website	177	任務來自與訓練資料集相同領域，但不同網站，旨在評估 Agent 於陌生網站上的泛化能力。	American AIrlines (Travel 領域)	Find a flight from Chicago to London on 20 April and return on 23 April.
	Cross-Domain	912	訓練與測試資料集來自不同領域，旨在評估 Agent 於陌生領域的泛化能力。	DMV Now (Service 領域)	Open page to schedule an appointment for car knowledge test.

2.3.3.2 資料集內容

Mind2Web 資料集的每筆任務包含三個主要部分：任務識別資訊、確認任務（confirmed_task），以及動作序列（actions）。其設計目的為模擬代理人在真實網頁環境下完成特定任務時，所需理解的任務目標與執行步驟。

任務識別資訊包含網站名稱、所屬領域與子領域等結構性欄位；確認任務（confirmed_task）明確說明了使用者的目標；動作序列（actions）則記錄了達成任務所需的一連串互動步驟。其資料結構與範例內容如表 2.3 所示。

表 2.3: Mind2Web 任務資料結構與範例內容

欄位名稱	**資料型別**	**說明**	**範例（以 sports.yahoo 上之任務為例）**
`annotation_id`	string	任務唯一識別碼	`37564222- bb58-4a55-b47b-e9ffbbc1d160`
`website`	string	網站名稱	`sports.yahoo`
`domain`	string	網站主領域	`Entertainment`
`subdomain`	string	網站子領域	`Sports`
`confirmed_task`	string	任務描述	`"Find the results of the most recent NFL games."`
`actions`	list[dict]	動作步驟列表。每個元素皆為單一步驟的資訊結構，詳細內容如表 2.4 所示。	詳細內容如表 2.4 所示

doi:10.6342/NTU202500827

在上述結構中，actions 為資料集中最關鍵的欄位，記錄了代理人完成任務時的實際操作步驟。每一步操作包含網頁的 HTML 狀態（過濾前與過濾後）與所執行的操作內容，欄位說明如表 2.4 所示。

表 2.4: Mind2Web 任務中 actions 欄位之結構與範例

欄位名稱	**資料型別**	**說明**	**範例（以 sports.yahoo 上之任務為例）**
action_uid	string	單一步驟的唯一識別碼	c62fa753- fdf3-4a97-a464-d6e1a2d7c20f
raw_html	string	操作前的完整 HTML 結構，包含所有原始標籤與屬性	過長略去
cleaned_html	string	經過濾機制處理後所保留的 HTML，僅保留可視且具語意價值的元素。該處理流程將於 2.4.1 詳述。	過長略去
operation.op	string	操作類型，包含 CLICK, TYPE, SELECT	CLICK
operation.value	string/null	操作參數，表示輸入文字或選取項目（若無則為空字串）	""

2.3.3.3 任務流程

Mind2Web 資料集中的每個任務由一系列固定步驟組成，每個步驟均對應一組標準答案，形式為一個 (Target Element, Operation) 配對。整體流程可分為三個階段：初始化、執行與評估。

在任務初始化階段，Agent 會獲取一組完整的輸入資訊，包含任務描述（Task Description）、網頁快照（Webpage Snapshot），以及歷史行為軌跡（Action History）。任務描述用以說明使用者的需求與目標，網頁快照則提供 HTML 結構與視覺內容等當前頁面資訊，而歷史行為軌跡則記錄了任務執行過程中的每個先前步驟，包括該步驟的網頁快照與正確的 (Target Element, Operation) 對。

接著進入任務執行階段。在此階段中，Agent 須根據當前任務描述、網頁狀態與累積的操作歷程，逐步預測下一個應執行的 (Target Element, Operation)。值得注意的是，無論 Agent 前一步的預測是否正確，系統都會提供完整的歷史標準答案作為輸入。此設計使得每一個步驟的預測可以獨立進行，避免前一步錯誤造成的連鎖影響，有助於更精確地評估單步執行準確率。

最後在任務評估階段，系統會將 Agent 在每一步的預測結果與對應的標準答案進行比對，據此計算整體任務的執行效能。

doi:10.6342/NTU202500827

2.4 LLM-Based Agents on Web Navigation 研究與發展

本節回顧三個具代表性之 LLM-Based Web Navigation Agents，包括 MindAct、Synapse 與 AWM。透過了解這些系統的運作與檢索機制，我們得以釐清其潛在限制，並據此作為本研究系統架構設計與改良的依據。

2.4.1 MindAct：具資訊過濾機制的網路瀏覽代理人

MindAct 由 Mind2Web 團隊提出，是最早期專注於 LLM-Based Web Agents 在 Web Navigation 任務中的應用研究。其主特色為利用兩階段的過濾（Filtering）機制，解決 HTML 結構複雜、無關資訊過多等等挑戰，以提升 Agent 的表現。

如圖 2.3 所示，在解任務的每個步驟中，Agent 會接收當前頁面的 HTML Document 及任務描述（Task Description）。首先，Agent 會對 HTML 內容進行過濾（Filtering），以減少無關資訊並提取關鍵元素。隨後，經過過濾的 HTML 片段、過去動作（Previous Actions）與任務描述將被組合為 Prompt，並輸入至 LLM 以預測該步驟的 (Target Element, Operation) 對。

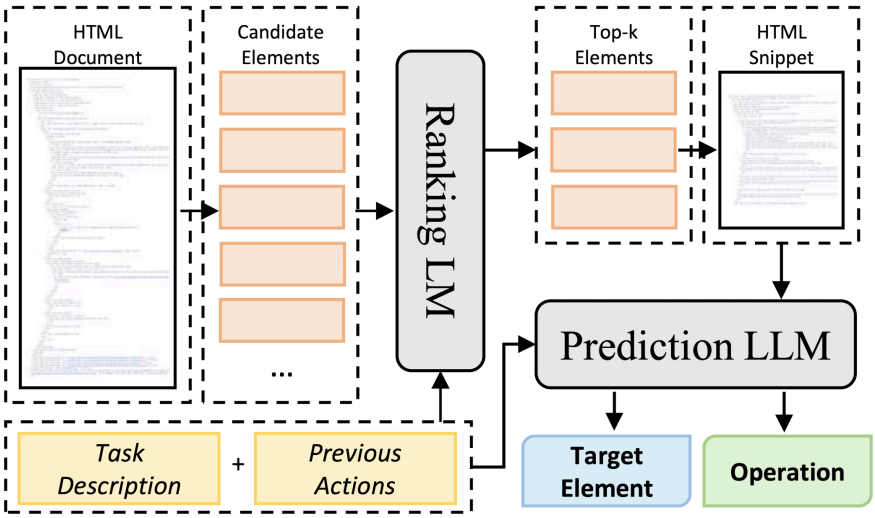

圖 2.3: MindAct 主要架構 [2]

為了提升代理人在網頁瀏覽任務中的決策效率，MindAct 設計了一套兩階段的 HTML 過濾機制，目的在於從複雜且冗長的網頁結構中，篩選出對任務執行最具關聯性的元素。

第一階段為初步篩選。系統在任務執行前，會先對原始 HTML 結構進行過

doi:10.6342/NTU202500827

濾，僅保留可視且具語意價值的元素。此階段可將平均元素數量從 1135 減少至 580，並在 94.7% 的任務中成功保留目標元素（Target Element）。

第二階段則是基於 DeBERTa 的元素篩選。如圖 2.4 所示，MindAct 採用微調後的 DeBERTa[19] 模型，對第一階段保留下來的候選元素進行進一步排序。模型輸入包含候選元素表示（Candidate Representation）、任務描述（Task）以及過去動作（Previous Actions），輸出為該元素是否為目標操作對象的二元標籤。雖然訓練目的是分類，但系統在推論階段會利用分類模型所輸出的機率分數進行排序，並保留前 k 個得分最高的元素，以構成最終的精簡 HTML（Clean HTML）。當 $k = 50$ 時，精簡 HTML 仍能以 86.6% 的機率保留 Target Elements，顯示該方法能在保留關鍵資訊的同時，有效降低干擾因素

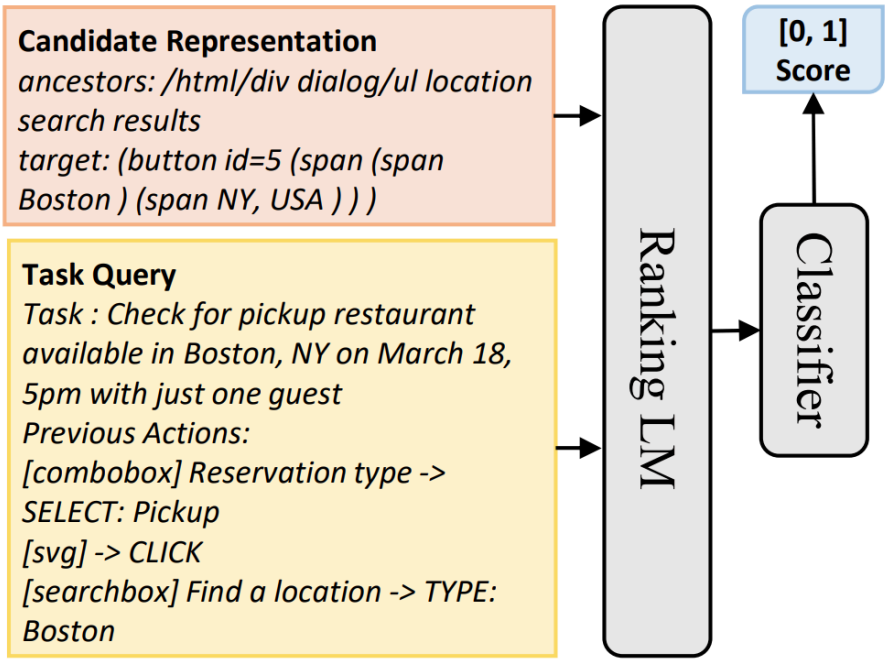

圖 2.4: MindAct 基於 DeBERTa 的元素篩選 [2]

研究團隊除了使用經針對訓練資料進行微調（Fine-tuning）的 FLAN-T5[20]，同時亦嘗試使用 GPT-3.5 搭配上下文學習（In-Context Learning, ICL）執行 Web Navigation 任務。

雖然 GPT-3.5 的表現略遜於經微調的 FLAN-T5，但結果顯示，即使未經微調，Foundation Model 仍可在 Web Navigation 任務中展現一定程度的能力，顯示 LLM 在此類應用的潛力。

值得一提的是，MindAct 所提出的雙階段過濾機制，在資訊縮減與關鍵元素

doi:10.6342/NTU202500827

保留之間展現出良好平衡。因此，本研究亦採用其過濾機制作為本系統預處理階段的重要組件。

2.4.2 Synapse：以軌跡做為範例的語義檢索機制

除了處理資訊過載，另一條提升 LLM-Based Agent 表現的關鍵方向，在於如何有效設計提示（Prompt）結構，使模型能在任務執行過程中適當參考過往經驗並做出合理推論。Synapse 即為此方向的重要代表，其核心特色在於提出了軌跡做為範例（Trajectory-as-Exemplar, TaE）機制將環境與 Agent 互動軌跡作（Trajectory）為 In-context Learning 的範例，以增強模型理解與任務執行能力。此外，Synapse 設計了一套基於「語義相似度」（Semantic Similarity）的 TaE 檢索機制，稱為 Exemplar Memory，使 Agent 能動態檢索與當前任務或環境相似的軌跡，進一步提高表現。

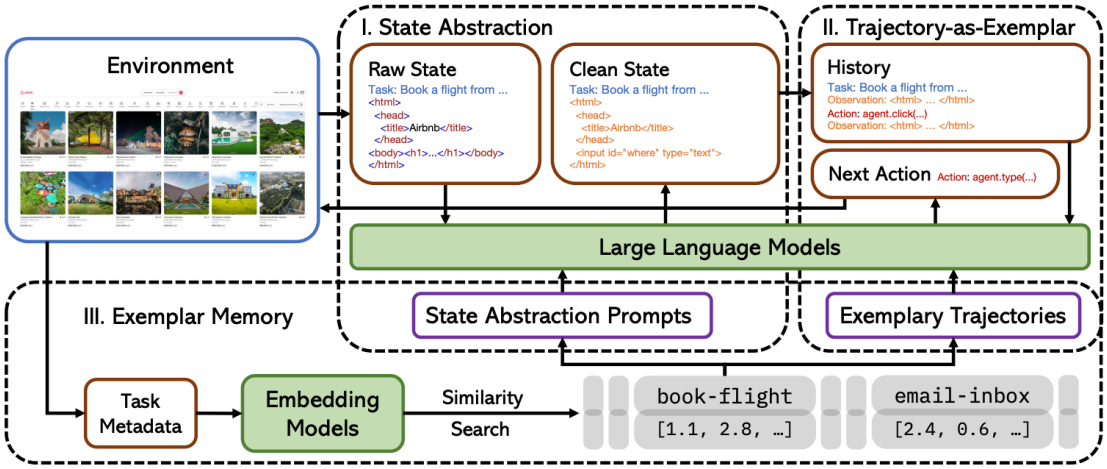

圖 2.5: Synapse 主要架構 [3]

TaE 的運作方法，為將代理人在任務歷程中逐步觀察與執行的資訊編排為一條完整軌跡（Trajectory），並作為大型語言模型的提示內容。如圖2.6所示，每條軌跡包含任務描述（Task）、每個步驟的觀察結果（Observation），以及所執行的操作（即目標元素 Target Element 與操作類型 Operation 的配對）。

TaE 提示來源可分為兩種方式。如表2.5所示，第一種方式是採用預先指定的固定範例，將相同數量的軌跡套用於所有任務，不具備動態性；第二種方式則透過 Exemplar Memory 模組，在每次任務開始前，根據任務情境即時檢索出 n 筆軌跡。

doi:10.6342/NTU202500827

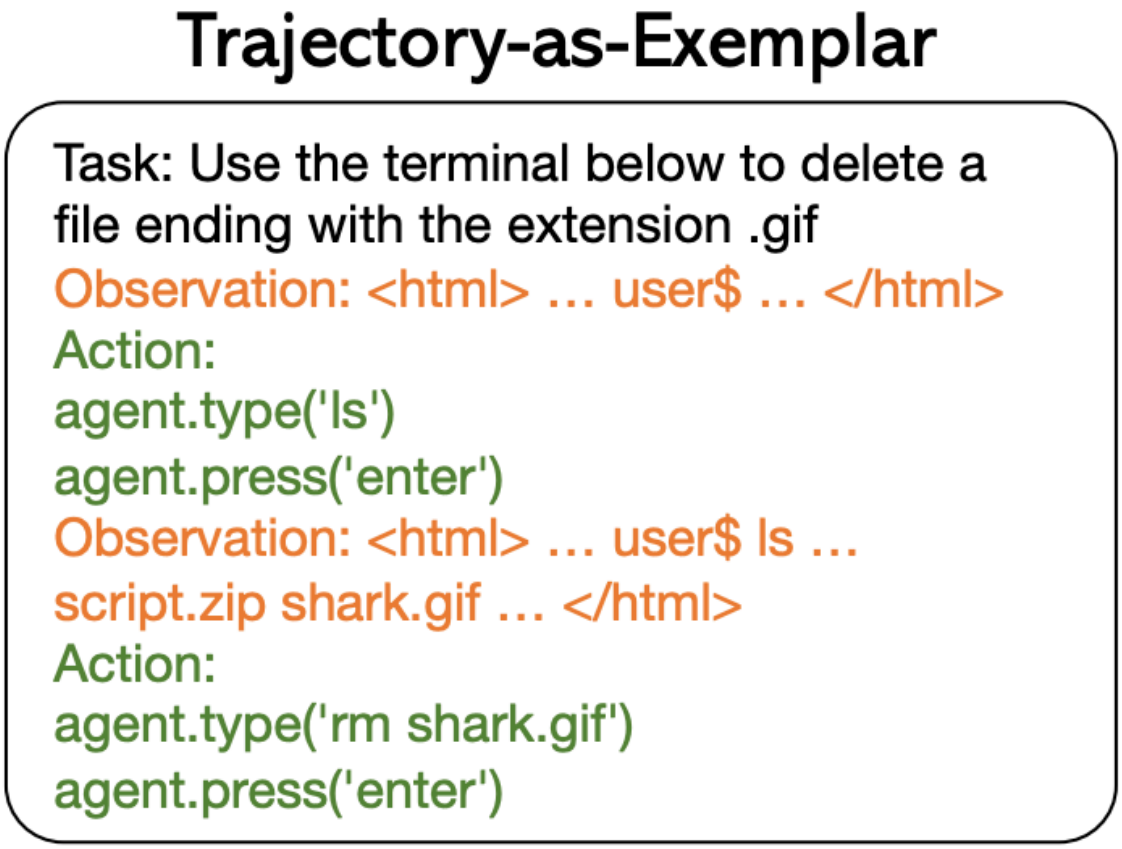

圖 2.6: Trajectory 內容 [3]

表 2.5: Synapse 系統中兩種 TaE 使用方式比較

	不搭配 Exemplar Memory	搭配 Exemplar Memory
Trajectories 來源	預先指定的 n 筆 Trajectories，於所有任務中固定使用	每次任務前由 Exemplar Memory 根據任務情境動態檢索 n 筆 Trajectories
機制詳述	無額外記憶模組，直接將固定 Trajectories 放入 Prompt	透過 Exemplar Memory 機制檢索合適的 Trajectories 放入 Prompt

doi:10.6342/NTU202500827

除了 TaE 機制外，Synapse 進一步提出了 Exemplar Memory 機制，透過語意相似度（Semantic Similarity）檢索自動挑選與當前任務語意最接近的 Trajectories，作為 Prompt 中的範例，協助模型進行 In-context Learning。在使用時，該機制可分為建構記憶階段以及使用記憶階段：

1. **建構記憶階段（針對訓練資料中的每筆任務）**：在階段，系統會先將每筆任務的 Metadata（包含 Task Description、Domain、Subdomain、Website）串接為一段文字描述，並使用嵌入模型將該描述進行編碼，取得其語意向量（Vector Representation）。最後，將此向量與對應的完整軌跡一併儲存至向量資料庫（Vector Database）中。

2. **使用記憶階段（針對每次測試任務）**：在階段，系統會針對當前任務的 Metadata 建立與訓練階段相同格式的文字描述。接著，使用嵌入模型將該描述進行編碼，取得其語意向量。於向量資料庫中執行語意相似度檢索，選取與該描述最相近的前 n 筆 Trajectories。最後，再將這些檢索結果作為當前任務的 TaE，嵌入於 Prompt 中，作為提示內容提供給模型。

此種基於語意相似度的動態檢索機制，使 Synapse 能依據任務語境自動選出最適合的學習範例，增強模型在類似任務上的泛化能力。因此不僅接下來介紹的 AWM 會使用，在本研究中，也使用 Synapse 所提出的 TaE 機制以及 Exemplar Memory 機制。

2.4.3 Agent Workflow Memory：使用共同工作流的提示機制

受人類透過過往經驗建立例行公事（Routine）解決問題的啟發，代理人工作流記憶機制 (Agent Workflow Memor, AWM）[4] 在 Synapse 的基礎上引入工作流（Workflow, WF）機制以提升 Agent 的學習與適應能力。

AWM 能從不同任務中歸納出共同工作流（Common Workflow），並在任務開始前，透過環境檢索工作流來輔助執行任務。例如，無論是在 Amazon 搜尋最便宜的貓食，或將最便宜的狗飼料加入購物車，都涉及商品搜尋與價格排序，這類跨任務可重複使用的流程即為 Common Workflow，使 Agent 能更高效適應多樣化任務。

doi:10.6342/NTU202500827

如圖 2.7 所示，AWM 的 Workflow 包含三個元素。第一為 Workflow 名稱，用以標示其功能，例如圖中第一行；第二為功能描述，說明該流程所對應的任務目的，對應圖中第二行；第三則為一系列操作步驟，由一系列 (Target Element, Operation) 配對組成，描述具體的操作步驟，如圖中的第三至第四行。

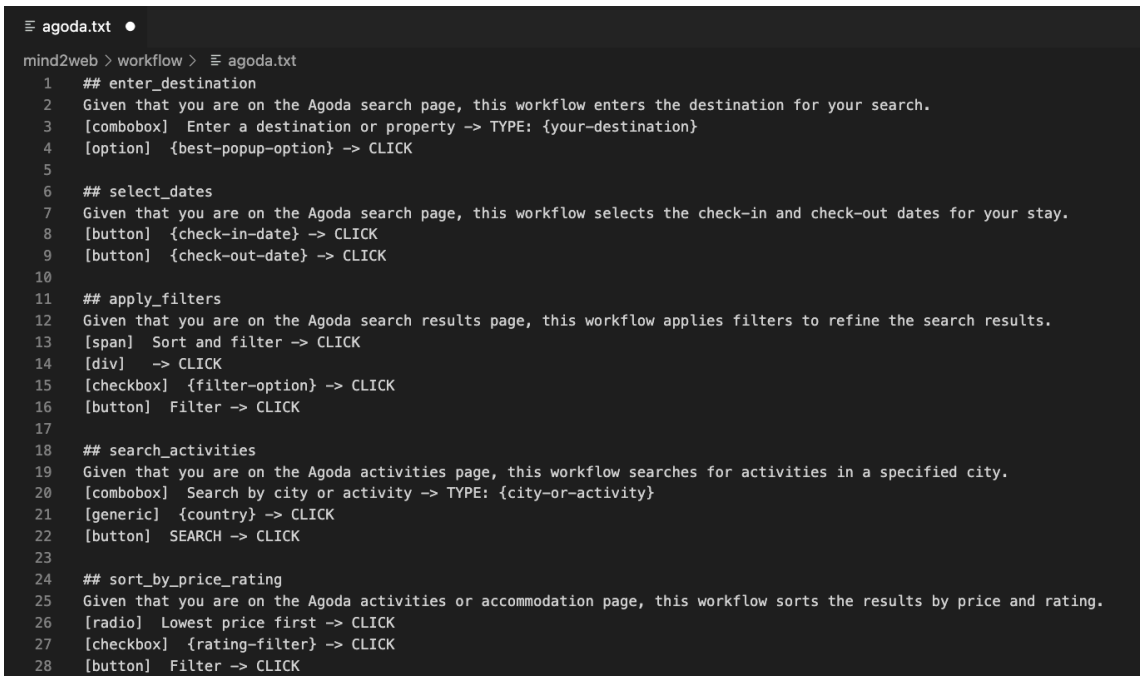

圖 2.7: AWM 中以 Agoda 任務與解答所歸納出的 Common Workflows 範例

Workflow 的建立與應用分為兩個階段，如圖 2.8 所示。在訓練階段（Induce Workflows），系統會將來自同一網站的所有任務及對應解答合成為一組 Prompt，交由 LLM 歸納出該網站的 Workflow Set，並儲存為文字檔。例如，若訓練資料中在 Agoda 網站上包含 15 筆任務，系統會將其整理為單一輸入，交由 LLM 歸納並產生 7 個代表性 Workflow，並儲存為 'Agoda.txt'。在此過程中，Prompt 會特別要求使用抽象化語言描述元素（Element），例如「價格欄位」而非具體 DOM ID，以提升泛化能力。

接著，在應用階段（Apply Workflows），當進入測試階段時，系統會根據任務的 Metadata（如 Website 或 Domain）檢索相應的 Workflow，並將其整合進 Prompt 中，作為提示詞的一部分輸入 LLM，以輔助任務推理與執行。

AWM 的 Workflow 檢索策略依任務類型而有所不同。在 Cross-task 設定中，系統會選取與當前任務相同網站的完整 Workflow 檔案；在 Cross-website 設定中，從相同領域中其他網站的 Workflow 中隨機選取（原論文未說明選取數量）；至於 Cross-domain，則從所有領域的 Workflow 中進行隨機挑選（同樣未明示選取筆

doi:10.6342/NTU202500827

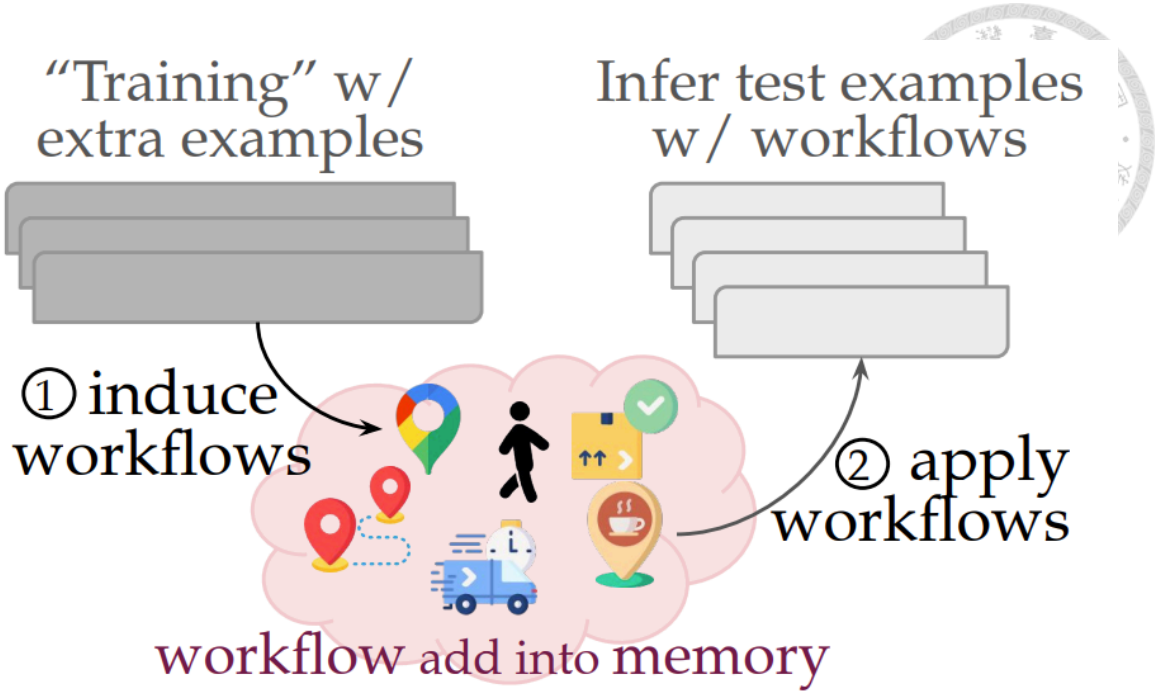

圖 2.8: AWM 的 Workflow 機制 [4]

數）。

雖然 AWM 提出 Workflow 機制以強化 Agent 表現，然而，由於其檢索機制僅依據任務的環境資訊，導致在做 Workflow 檢索時可能選取到和當前任務情境不相關的範例。因此，本研究將延續 AWM 工作流設計的理念，提出一套新的 Workflow 檢索機制，期能在保留通用性之餘，提升 Workflows 與任務語意之間的相關度，進而提升任務表現。

除了引入 Workflow 機制外，AWM 同時延續 Synapse 所提出的 Trajectory-as-Exemplar（TaE）設計，作為提示內容的一部分。但與 Synapse 基於語義相似性檢索範例不同，AWM 則根據任務所處的環境條件進行檢索。在 Cross-task 設定中，AWM 會從與當前任務相同網站的 TaE 中隨機選取 1 筆；在 Cross-website 設定中，從相同領域（Domain）中的其他網站中隨機選取 1 筆；而在 Cross-domain 中，則從所有 TaE 中隨機挑選 1 筆。

這種以任務環境為依據的檢索邏輯雖簡便，但在實際應用中可能無法充分考量當前任務與檢索範例之間的關聯性，進而影響 Agent 表現。因此，本研究將於第三章進一步比較兩種 TaE 檢索策略的效能表現，並選擇較具優勢的機制作為本研究之選擇。

doi:10.6342/NTU202500827

第三章　研究方法

本研究針對 LLM-Based Agent 在網頁瀏覽任務中的表現，提出一套整合式架構——Memory-Integrated Mechanism with Similarity（MIMS）。MIMS 結合兩項關鍵設計：其一，延續 Synapse 所提出的語意檢索方式，用於選取具代表性的 TaE；其二，新增一項基於動作目標語意相似性的 Workflows 檢索策略。

為釐清上述方法設計之必要性與潛在效益，本章將首先對 Synapse 以及 AWM 進行分析，並說明其機制的限制與啟發。接著，我們將詳述 MIMS 系統之整體架構、核心模組與設計原理。

3.1　現有系統分析：Synapse 與 AWM

為評估現有系統在不同任務類型下的效能差異，本節將分析兩套代表性方法——Synapse 與 AWM——的架構特性與檢索策略。具體而言，我們將聚焦於兩項關鍵機制進行比較分析：（1）TaE 檢索方式的異同與表現；（2）Workflow 檢索機制的潛在侷限。

為確保分析具備客觀性與可重現性，我們首先說明所依據之實驗設定與評估方法，並據此展開實證比較與結果討論。這些觀察結果亦將作為後續提出 MIMS 系統設計的依據。

3.1.1　實驗設定與評估方法

本小節將依序說明進行比較分析所依據的實驗環境、資料集來源，以及使用的效能評估指標。相關設定與配置將作為後續實驗中不同方法公平比較的基礎。

doi:10.6342/NTU202500827

3.1.1.1 實驗環境

如表3.1所示，在本研究中，我們採用 OpenAI 所推出的語言模型 gpt-4o-mini-2024-07-18 [21] 作為主要的 LLM 模型進行調用。GPT-4o（其中“o”代表”omni”）是 OpenAI 的多模態能力語言模型 [22]，相較於 GPT-4 [23] 不僅在回應速度上表現更佳，在多項標準測試（benchmark）中亦展現出更優異的性能。GPT-4o mini 為其小型版本，不僅具備更低的推理延遲與使用成本，亦延續了 GPT-4o 優秀的多模態推理能力。根據官方測試結果，儘管 GPT-4o mini 在所有任務的整體表現略遜於 GPT-4o 本體，但其在多數任務上仍優於 GPT-3.5 Turbo，因此被 OpenAI 官方稱為「最具成本效益的小型模型（cost-efficient small model）」。

與 GPT-4 相比，GPT-4o mini 的推理成本僅為其約 1%；與 GPT-3.5 Turbo 相比，也僅需約 2/3 的成本。因此，基於效能與成本之間的平衡考量，我們選擇使用 gpt-4o-mini 作為本研究的主要語言模型。

為確保方法間比較的公平性，我們亦基於相同的 gpt-4o-mini-2024-07-18 模型，重新實作並執行 SYNAPSE 與 AWM 等現有方法的實驗。此外，在嵌入模型與向量資料庫的選擇上，我們亦延續 Synapse 所採用的配置，以維持設定一致性。

為了降低模型回應的隨機性，我們將溫度參數（Temperature）固定為 0。考量成本限制，所有實驗皆僅執行一次。

表 3.1: 實驗環境設置

API	
LLM	OpenAI gpt-4o-mini-2024-07-18 [21]
Embedding	OpenAI text-embedding-ada-002[24]
資料庫	
向量資料庫	FAISS[25]
參數設定	
模型溫度 (Temperature)	0

doi:10.6342/NTU202500827

3.1.1.2 資料集

在本研究中，我們分別使用了 Mind2Web 的訓練以及測試資料集。表 3.2 與表 3.3 分別列出訓練集與測試集的統計資訊。

訓練資料共包含 1,009 筆任務，分布於 73 個真實網站中，涵蓋 3 個主要領域與 18 個子領域。整體平均每網站約有 13.8 筆任務，任務密度介於 5 至 24 筆之間，顯示資料具一定多樣性。

表 3.2: 實驗訓練資料集資訊

總任務數	總網站數	總領域數	總子領域數	每網站平均任務數	每網站最少/最多任務數
1009	73	3	18	13.8	5 / 24

測試資料集則進一步分為三個泛化層級：Cross-Task、Cross-Website 與 Cross-Domain。透過這樣的資料分層設計，我們能全面評估模型在熟悉網站、類似網站、以及全新領域下的表現差異，並為後續的系統比較與泛化分析提供明確基準。

表 3.3: 實驗測試資料集資訊

子測試資料集	任務數	網站數	領域數/子領域數	每網站平均任務數	每網站最多/最少任務數
Cross-Task	252	69	3 / 18	3.6	10 / 1
Cross-Website	177	10	3 / 10	17.7	23 / 13
Cross-Domain	912	54	2 / 13	16.8	33 / 5

3.1.1.3 評估指標

本研究參考 Mind2Web 所提出的四項評估指標，分別為元素準確率 (Element Accuracy, EA)、操作 F1 分數 (Operation F1, Op. F1）、步驟成功率 (Step Success Rate, SSR) 以及任務成功率 (Success Rate, SR)，用以衡量代理人在任務執行過程中的表現。

以下將依序介紹這四項評估指標的數學定義與設計理念，說明其在 Web Navigation 任務中的評估重點。需要說明的是，在後續實驗中，實際僅採用 EA、SSR 與 SR 作為效能指標。至於 Op. F1，則因其評估方式存在潛在偏誤，未納入本研究之分析範圍，相關排除原因亦將於指標說明中詳述。

1. 元素準確率 (Element Accuracy, EA)：此指標檢測代理人是否正確預測了該步驟中應操作的元素（Target Element）。其數學定義如下：

doi:10.6342/NTU202500827

$$\text{EA} = \frac{1}{T}\sum_{i=1}^{T} 1[\hat{e}_i = e_i], \tag{3.1}$$

其中 T 為測試任務中的總步驟數，$\hat{e}_i$ 與 e_i 分別代表代理人預測與標準答案中的目標元素，指示函數 $1[\cdot]$ 的值在條件成立時為 1，否則為 0。

2. 操作 F1 分數 (Operation F1, Op. F1)：Op. F1 用於衡量代理人預測之操作（Operation）的準確性。其數學定義為：

$$\text{Operation F1} = \frac{1}{T}\sum_{i=1}^{T} \text{F1}(\hat{o}_i, o_i), \tag{3.2}$$

其中 $\hat{o}_i$ 與 o_i 分別為第 i 步預測與標準的操作指令，$\text{F1}(\cdot)$ 表示 token-level 的 F1 分數函數。

3. 步驟成功率 (Step Success Rate, SSR)：SSR 同時檢驗代理人在每個步驟中是否正確預測了目標元素與操作行為。其數學定義為：

$$\text{SSR} = \frac{1}{T}\sum_{i=1}^{T} 1[(\hat{e}_i, \hat{o}_i) = (e_i, o_i)], \tag{3.3}$$

此處 $(\hat{e}_i, \hat{o}_i)$ 與 (e_i, o_i) 分別為第 i 步的預測與標準 (Target Element, Operation) 配對。

4. 任務成功率 (Success Rate, SR)：SR 衡量的是代理人是否能在整個任務中，每一個步驟皆正確完成操作。若任務中有任一步驟失誤，該任務則被視為整體失敗。其定義如下：

$$\text{SR} = \frac{1}{M}\sum_{j=1}^{M} 1[\text{SSR}_j = 1], \tag{3.4}$$

其中 M 為總任務數量，SSR_j 為第 j 筆任務中所有步驟是否皆正確的指標值。

雖然本研究完整介紹了 Mind2Web 所提出的四項指標，但考量 Op. F1 僅針對操作行為進行評估，且即使預測目標元素錯誤，該指標仍可能回傳正確結果，導致與實際任務表現脫節。在 Mind2Web 原始研究中的分析，Op. F1 僅被列入報告結果，卻未進行進一步的探討與詮釋；而在 SYNAPSE 後續的研究中，更是選擇

doi:10.6342/NTU202500827

將此指標完全移除。有鑑於此，本研究亦採取相同策略，排除 Op. F1 作為評估指標，以提升整體評估準確性。

此外，雖然 SR 能反映任務是否整體成功，卻無法區分部分正確與完全錯誤的細節差異。例如，在共 9 步的任務中，無論是一步錯，或是九步錯，都會得到當前 SR 為 0 的結果。而在 SSR 的分析中，則會分別出現 8/9 以及 0/9 的差異。

EA 雖然也具備細部檢驗能力，但由於其與 SSR 呈現高度一致的趨勢。因此，本研究將 SSR 作為主要評估指標，用以觀察 Agent 於細部步驟與整體任務上的執行效能。

3.1.2 TaE 檢索機制分析

在第二章中，我們說明了 Synapse 與 AWM 系統均採用了 TaE（Trajectory-as-Exemplar）檢索機制。為了選取表現較為良好的機制作為本研究使用，本節將會對兩者的表現做比較與分析。Synapse 採用基於語意相似性的檢索方式，而 AWM 則透過網站或領域作為條件進行隨機檢索。本研究稱前者為「語意檢索」，後者為「環境檢索」，如表 3.4 所示。

表 3.4: TaE 檢索機制種類及說明

TaE 檢索機制	說明
語意檢索	透過語意相似性，檢索 n 個和當前任務描述最相似的 Trajectory
環境檢索	於 Cross-Task：隨機檢索 1 個與當前任務同一網站的 Trajectory 於 Cross-Website：隨機檢索 1 個與當前任務同一領域的 Trajectory 於 Cross-Domain：隨機檢索 1 個 Trajectory

我們在實驗中停用 Workflow 檢索機制，控制變因，以 SSR 作為指標比較無 TaE、語意檢索與環境檢索三種情境的結果。此外，為使語意與環境檢索在數量上公平比較，均設定檢索 1 筆 Trajectory。

如表 3.5 所示，我們首先可以看到，無論是基於「語意檢索」還是「環境檢索」，只要為 Agent 提供額外的 TaE 資訊，均能為 Agent 帶來正向影響。接著，我們可以看到，「語意檢索」相較於「環境檢索」，在所有的情境都有更好的表現。特別是在 Cross-Website 與 Cross-Domain 任務中，展現出更明顯的表現提升。這樣的趨勢顯示，在面對較陌生的網站或領域任務時，語意檢索所提供的 Trajectories 更可能對代理人行動產生實質幫助。

doi:10.6342/NTU202500827

表 3.5: 以 Step Success Rate 為評估指標，在 Mind2Web 的三個子測試資料集中比較無使用 TaE 檢索、使用語意檢索及使用環境檢索的表現

TaE 檢索機制	Cross-Task	Cross-Website	Cross-Domain
無	27.6	21.9	24.7
語意檢索	**34.3**	**26.1**	**28.1**
環境檢索	33.7	24.8	27.0
相對提升（語意 / 環境）	+1.78%	+5.24%	+4.07%

3.1.3 Workflow 檢索機制分析

本節分析 Workflow 選取方式對模型表現的潛在影響，並進一步指出其限制與改進可能。現有方法在 Cross-Task 任務中仰賴來自相同網站的 Workflows 作為檢索來源，雖非基於語意相似性挑選，但實際上，這些 Workflows 往往仍具有與目標任務相當程度的語意重疊。以 Target 網站為例，某測試任務為：“Add a set of queen-sized bed sheets with at least a 4-star rating to the cart.” 而 Workflow Set 中某項 Workflow 則為：“Given that you are on the product detail page, this workflow adds a specific product to the shopping cart.” 由此可見，儘管上述 Workflow 並非透過語意檢索方式獲得，其所屬網站與任務一致，仍展現出高度語意相關性，並可為代理人提供結構相近之 Trajectory 參考。然而，由於僅依照環境來選取，Workflow Set 中也不乏語意無關之實例。

已有研究指出，當模型接收到過多無關資訊時，將可能干擾判斷，降低整體效能 [26]。因此，我們推測：若能導入更精準的語意檢索機制，主動篩選與當前任務語意高度相關的 Workflows，將有助於進一步提升代理人於任務執行時的推理準確性與成功率。

3.2 MIMS 系統架構

綜合前述分析結果，本研究提出一種基於「語意檢索」的 Workflows 檢索機制，並進一步與基於「語意檢索」的 TaE 檢索機制整合，形成名為 Memory-Integrated Mechanism with Similarity（MIMS）之系統架構。

MIMS 的整體架構如圖 3.1 所示。相較於 AWM 系統，MIMS 在流程上最大的差異為：在執行 Workflows 檢索之前，新增一個「動作目標預測」步驟。代理人

doi:10.6342/NTU202500827

會先透過任務描述、歷史軌跡與觀察內容，預測目前應執行的動作目標，接著再以該預測結果作為語意查詢依據，進行 Workflows 的語意相似度檢索。此外，需特別說明的是，Workflows 與 TaEs 的檢索程序在邏輯上屬於同步進行，兩者並無先後順序之別。

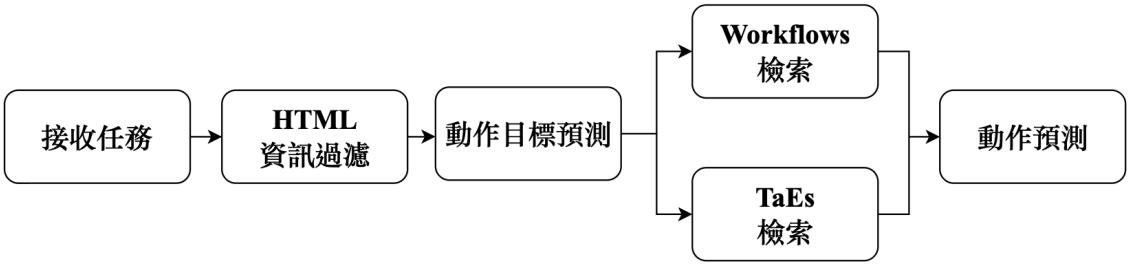

圖 3.1: MIMS 系統架構

在接下來的小節，我們會針對 HTML 資訊過濾、動作目標預測、Workflows 檢索、TaEs 檢索以及動作預測做詳細解說。

3.2.1 HTML 資訊過濾

在圖3.1中我們可以看到，在接收任務後，Agent 便會對所接收到的 HTML 資訊做過濾。和 Synpase 以及 AWM 一樣，我們採用了 MindAct 的兩階段過濾方法，透過排名機制，取 HTML 中前 k 個最具相關性的元素。

為了降低資訊量，我們參照 Synapse，將每一步的觀察結果（Observation）僅保留 k=5 個元素（在 MindAct 中 k=50）。雖然將 k 從 50 調整為 5 的作法，會讓觀察中包含 Target Element 的機率從 83% 下降到 53%，但在過去的實驗中，Synapse 卻因爲資訊量變少而導致 SSR 的表現提高了，因此我們在這邊也採用相同的作法。

3.2.2 動作目標預測

如圖3.1所示，在過濾完 HTML 資訊後，為了提升 Workflow 語意檢索的精準度，本研究引入「動作目標」（Action Objective）預測機制，作為查詢語句的生成依據，協助檢索出語意上更相關的 Workflow 描述。本節說明動作目標預測的任務定義、設計理念，以及我們如何透過大型語言模型（LLM）實現此預測機制。

doi:10.6342/NTU202500827

在每個時間戳記 t，代理人所處的任務狀態定義為：

$$S_t = \{\text{Task},\ H_{1:t-1},\ O_t\}, \tag{3.5}$$

其中，Task 表示任務描述；$H_{1:t-1}$ 為歷史軌跡（包含所有過去的觀察與動作）；O_t 為當前觀察。同時，為降低輸入長度與計算成本，在此沿用自 Synapse 的設定，令歷史軌跡 $H_{1:t-1}$ 中的 HTML Element 保留 $k = 3$ 個元素。

我們觀察到，一段 Workflow 描述本質上可視為一系列動作 $\{a_1, a_2, ..., a_n\}$ 的動作目標描述。因此，若能從當前狀態 S_t 準確預測出與該序列的目標 o_t，便可將該目標作為查詢基準，透過語意相似性檢索與當前情境最相符的 Workflows 描述。

理想情況下，系統應根據 S_t 預測出一整段動作序列 $\{a_t, a_{t+1}, ..., a_{t+r}\}$ 所對應的語意目標 $o_{t:t+r}$，並據此進行語意檢索。然而，實務上 Workflow 長度 r 不定，無法精確預測，這使得直接預測完整 Workflows 目標 $o_{t:t+r}$ 成為具挑戰性的任務。因此，本研究簡化問題設定，僅關注下一步動作所對應的語意目標 o_t，並將此問題形式化為使用 LLM 進行的預測任務：

$$o_t = \text{LLM}(S_t;\ \text{Prompt}^{\text{obj}}), \tag{3.6}$$

LLM$(\cdot)$ 表示大型語言模型函式，其輸入除了當前狀態 S_t 外，還需搭配預先設計好的 Prompt 模板 Prompt$^{\text{obj}}$，以明確引導模型生成具備上下文語意的一段自然語言動作目標 o_t。

另外，在模板中，我們會透過提供回應範例，請 Agent 在回應中判斷當前的網站資訊。例如，在「Given that you are on the Under Armour website, the next objective is to navigate to the appropriate product category.」這個範例中，我們可以觀察到其前半部分”Given that you are on the Under Armour website”強調當前所處的網站環境，而後半部分”the next objective is to navigate to the appropriate product category”描述下一步可達成的目標。這樣的設計源於我們在 Workflow 檢索機制分析中的發現——「環境相似性」對於準確檢索適當的 Workflow 至關重要。此外，此格式也與 Workflow 本身的結構相符。

圖 3.2 展示了使用者輸入與 LLM 之間互動，進而生成動作目標的過程。整體流程可分為兩個區塊：User 與 LLM。在 User 區塊中，輸入內容由預先定義的

doi:10.6342/NTU202500827

提示模板 $\text{Prompt}^{\text{obj}}$ 組成（圖中紅色文字），其中包含代理人的動作空間（Action Space）、回應格式（Response Format），以及數個範例回應（Example Responses），用以指引模型理解與生成回應。接著，輸入中會附上任務描述（Task）、歷史軌跡（Trajectory）與觀察資訊（Observation）（圖中黑色文字）。上述內容會整合為一筆輸入，送入 LLM 作為生成基礎。LLM 區塊則為模型的輸出結果，指出當前所在網站與下一步應執行的動作目標 o_t。

3.2.3 Workflows 檢索

在圖3.1中我們可以看到，在完成動作目標預測後，agent 會同時執行 Workflows 檢索及 TaEs 檢索。以下會分別說明在資料前處理階段以及測試階段的運作流程。

3.2.3.1 資料前處理

在正式使用 Workflows 協助代理人執行任務前，系統需先對 Workflows 進行前處理，可分為以下兩個階段：

1. **Workflow 歸納**：根據訓練資料中標註的任務軌跡，彙整出具代表性的 Workflows。此步驟與 AWM 的方法相同。

2. **向量資料庫建構**：將每條 Workflow 儲存至向量資料庫（Vector Database），流程如下：

   (a) 對 Workflow 的描述（Workflow Description）進行語意嵌入（Embedding），得到向量表示 w_i。

   (b) 將 Workflow 的所有欄位（包含標題、描述、步驟序列）作為對應的 Metadata，與向量 w_i 一併存入資料庫。

舉例來說，圖 3.3 所示為一條從 Agoda 任務中歸納的 Workflow。在我們對第二行 Workflow Desccription 進行 Embedding 後取得向量表示，將所有資訊（一到四行）作為對應的 Metadata，一併存入資料庫。

doi:10.6342/NTU202500827

1. User

You are a large language model trained to navigate the web.
Identify the current webpage and determine the objective of
the next action based on the given task, trajectory, and observations.
Here is the action space:
1. CLICK [id]: Click on an HTML element with its id.
2. TYPE [id] [value]: Type a string into the element with the id.
3. SELECT [id] [value]: Select a value for an HTML element by its id.
Response format:
Provide a single simple sentence that identifies the current webpage and
states the next action's objective while ensuring that specific entities
(such as product names, locations, or categories) are abstracted
without losing their essence, and the website name remains unchanged.
Example responses:
'Objective: Given that you are on the Under Armour website,
the next objective is to navigate to the appropriate product category.'
'Objective: Given that you are on the United Airlines website,
the next objective is to select the correct departure city from the dropdown menu.'
'Objective: Given that you are on TicketCenter,
the next objective is to select a sports team from the list of search results.'

Task:
As a Verizon user, finance a new blue iPhone 13 with 256gb along with monthly apple care
Trajectory:
Observation:
'<html> <nav main><button id=195 button menu>
<svg id=196 img />Menu </button></nav><form frmsearch>
<input id=99 text type to search. navigate forward st hi, james! what can we />
<button clear search text></button>
<button submit submit search></button><input _dyncharset utf-8 />
</form> </html>'
Action:
'TYPE [99] [Iphone 13]'
([textbox] Type to search. Navigate forward to hear suggestio... ->TYPE: Iphone 13)
Observation:
'<html> <nav main><button id=3314 button menu>
<svg img />Menu </button></nav><form frmsearch>
<div suggested searches>
iphone 13 <button>Show related products </button>
<ul related products for iphone 13>
<div related products>
<div apple - iphone 13 5g>
Apple - iPhone 13 5G 128GB (Unlocked) - Blue
</form> </html>'

2. LLM

"Objective:
Given that you are on the Verizon website, the next objective
is to click on the link for the specific iPhone model you are interested in."

圖 3.2: MIMS 動作目標預測提示以及 LLM 回應

```
## enter_destination
Given that you are on the Agoda search page, this workflow enters the destination for your search.
[combobox]  Enter a destination or property -> TYPE: {your-destination}
[option]  {best-popup-option} -> CLICK
```

圖 3.3: Workflow 範例（來自 agoda.txt）

doi:10.6342/NTU202500827

3.2.3.2 測試階段運作流程

在測試階段，LLM 會先根據當前任務狀態 S_t 預測出動作目標 o_t，如前節所定義。我們將 o_t 透過相同的語意嵌入模型轉換為查詢向量 q，並以此作為查詢，進行語意檢索。

為了比對查詢向量與資料庫中儲存的 Workflow 向量，我們採用 L2 距離（歐幾里得距離）作為相似度度量。相較於內積（Inner Product），在向量未經正規化的情況下，內積的排序結果會同時受到向量方向與範數（norm）的影響，可能導致語意上相似但範數較小的向量被低估。相對而言，L2 距離可有效減少範數干擾，提供較穩定的一致性排序，因此更適用於未正規化語意嵌入的相似檢索任務。

本研究使用 FAISS 作為向量檢索系統，查詢流程如下：

$$\text{Top-}M(\text{Workflow}) = \underset{w_i \in \mathcal{W}}{\arg\,\text{top-}M} \, \|q - w_i\|_2, \tag{3.7}$$

其中 q 為嵌入後的動作目標向量，w_i 為第 i 條 Workflow 描述的向量表示，M 為檢索數量超參數。

最後，我們將兩不同 Workflows 檢索機制的種類及說明呈現於表3.6。

表 3.6: Workflow 檢索機制種類及說明

Workflow 檢索機制	說明
環境檢索	於 Cross-task：檢索當前網站之 Workflow 之 txt 檔 於 Cross-website：隨機檢索和當前任務相同之領域網站中的 Workflows 於 Cross-domain：隨機檢索來自所有領域的 Workflows
語意檢索	透過語意相似性，檢索 M 個和當前動作目標最相似的 Workflows

3.2.4 TaEs 檢索

如同前一節所述，在圖3.1中我們可以看到，在完成動作目標預測後，agent 會同時執行 Workflows 檢索及 TaEs 檢索。

根據前述實驗分析結果，相較於 AWM 採用的「環境檢索」策略，「語意檢索」機制在各種任務情境中皆展現更優異的表現。因此，在 MIMS 中，我們也選

doi:10.6342/NTU202500827

擇使用「語意檢索」作為 TaE 的檢索方法。

在測試階段，給定一筆新任務，系統會先對其描述 Task 做詞嵌入以取得對應向量表徵：

$$z_{\text{task}} = f(\text{Task}), \tag{3.8}$$

其中 $f(\cdot)$ 表示語意嵌入函數，用以將任務描述轉換為向量表示。

接著，於向量資料庫 $\mathcal{M} = \{(z_i, \text{TaE}_i)\}_{i=1}^{K}$ 中（其中 K 為訓練資料中可用的 exemplar 數量），根據向量間的 L2 距離（Euclidean Distance）進行檢索，選取與 z_{task} 距離最近的前 n 筆 exemplar：

$$\text{Top-}n(\text{TaE}) = \underset{(z_i, \text{TaE}_i) \in \mathcal{M}}{\arg\,\text{top-}n} \; \|z_{\text{task}} - z_i\|_2, \tag{3.9}$$

這些被選出的 Top-n(TaE) 會被嵌入至 Prompt 中，作為 In-context Learning 的例子，輔助大型語言模型完成後續任務推理。

在本研究中，我們與 Synapse 相同，採用 L2 距離作為語意檢索的距離度量方式，並使用相同的嵌入表示與向量比對策略。而為降低輸入長度與推理成本，系統亦延續 Synapse 的資料預處理方式：Trajectories 中每一步的 Observation 僅保留 $k = 3$ 個 HTML Elements。

3.2.5 動作預測

在完成 Workflow 與 TaE 的檢索後，系統將這些資訊與任務上下文組合為 Prompt，輸入至大型語言模型（LLM），以預測下一步應執行的動作 a_t，其中 a_t 表示代理人在時間步 t 的預期動作。此過程可形式化為：

$$a_t = \text{LLM}\left(\text{Task},\ \text{Workflows},\ \text{TaEs},\ H_{1:t-1},\ O_t;\ \text{Prompt}^{\text{act}}\right) \tag{3.10}$$

圖 3.4 展示了完整的動作預測提示以及 LLM 產生的動作預測。整體流程包含四個步驟，分別以圖中 1. User 至 4. LLM 四個區塊呈現。在 1. User 區塊中，系統首先引入預先定義的動作預測提示模板 Prompt$^{\text{act}}$（紅色文字），包含代理人目標

doi:10.6342/NTU202500827

以及 Agent 的動作空間（Action Space）。接著，系統逐步附加三部分資訊：首先是 MIMS 先前所檢索到的 M 個 Workflows 與以及 n 個 TaEs，再來是當前任務描述 Task 與歷史軌跡，最後為當前觀察環境的 Observation（黑色文字）。這些內容組成了 LLM 動作預測的完整輸入。2. LLM 區塊為語言模型的第一次回應，根據輸入資訊預測動作。接下來的 3. User 區塊呈現上述動作在環境中執行後所產生的 Observation O_t，系統將其與 Prompt^{act}、新檢索之 M 個 Workflows、任務一開始檢索的 n 個 TaEs、任務歷史軌跡 $H_{1:t-1}$ 組合，作為下一輪推理的上下文。最後，4. LLM 區塊顯示模型基於更新後觀察結果所做出的下一步動作預測。此預測結果即為 LLM 在 t 時的動作預測 a_t 。

doi:10.6342/NTU202500827

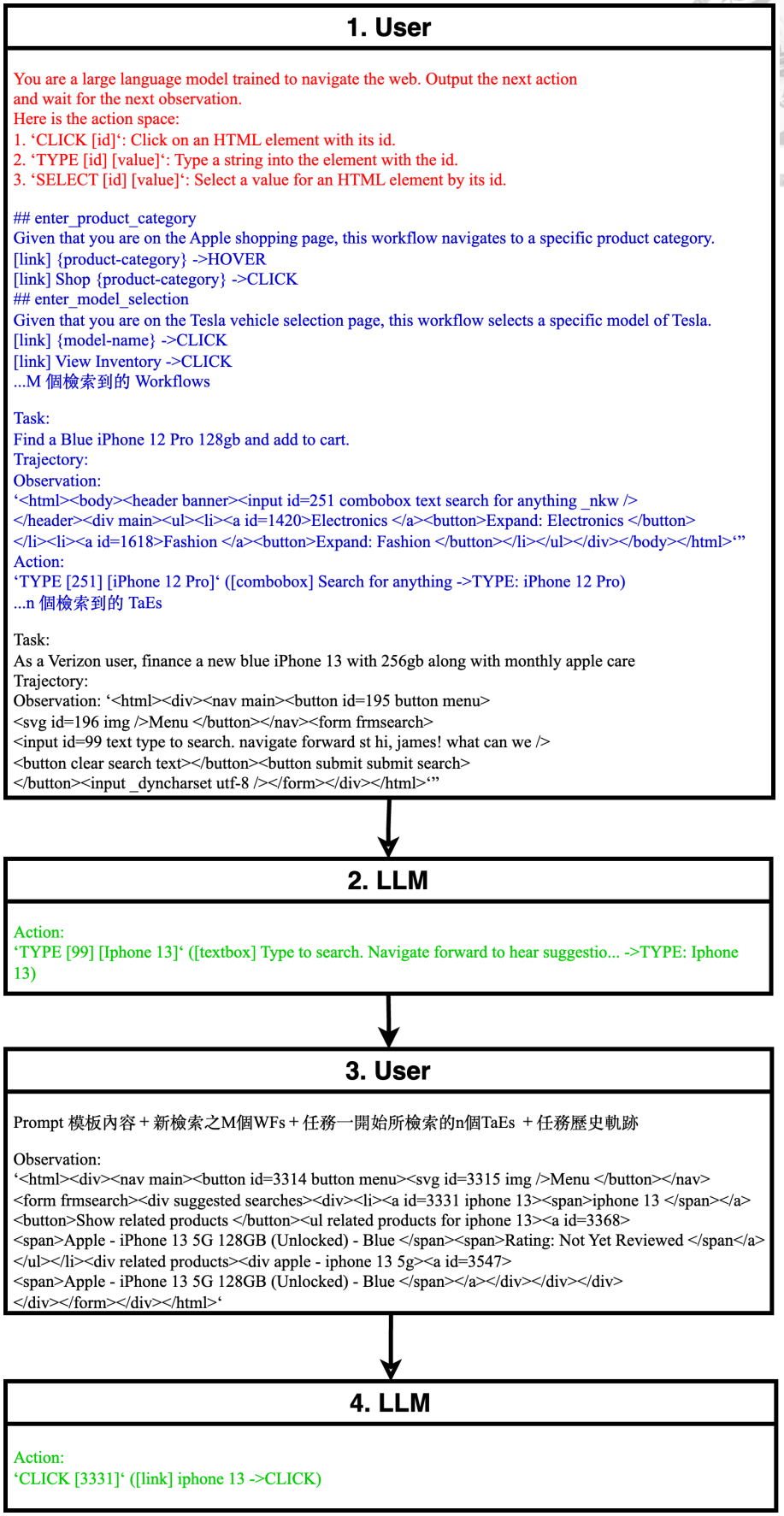

圖 3.4: MIMS 動作預測提示以及 LLM 回應

doi:10.6342/NTU202500827

第四章　實驗結果與討論

本章旨在透過實驗評估本研究所提出之 MIMS 系統的效能表現，並與現有兩套代表性系統——Synapse 與 AWM——進行比較分析。

首先，我們將聚焦於 Workflows 數量與 TaE 組合對 Agent 之影響進行系統性探討。接著，透過完整測試資料集，進一步比較 MIMS 系統與 Synapse、AWM 在三種泛化任務設定下的整體效能，並分析其優勢與潛在限制。

4.1　Workflows 數量與 TaE 組合對 SSR 表現影響之分析

為了探討 Workflows 數量與 TaE 組合如何影響 Agent 的表現，我們設計一組實驗，觀察不同 Workflows 數量搭配 TaE 數量（$n = 1, 2, 3$）於三種泛化任務類型（Cross-Task、Cross-Website、Cross-Domain）中的表現變化。

以下首先說明實驗環境與資料設定，接著逐一呈現三種任務類型下的結果與分析，最後以整合圖表歸納主要趨勢。

4.1.1　實驗設定與評估方法

本章實驗環境設定與評估方法皆與第三章相同，惟在測試資料集部分，為降低實驗成本，我們於涵蓋所有領域任務的前提下，從子資料集中選取部分網站進行測試，如表 4.1 所示。

如前所述，Workflows 檢索機制會根據訓練資料歸納出若干 Workflow Set，並應用於所有與 Workflows 相關的實驗中。表 4.2 展示了本研究所建立之 Workflow Set 的統計資訊。

doi:10.6342/NTU202500827

表 4.1: 完整測試資料集與選取資料集資訊

子測試資料集		Cross-Task	Cross-Website	Cross-Domain
選取說明		每個子領域 各選取一個網站任務	每個領域 各選取一個網站任務	每個領域 各選取兩個網站任務
任務數	選取	70	55	53
	總數	252	177	912
網站數	選取	18	3	4
	總數	69	10	54
每網站任務數	平均	3.6	17.7	16.8
	最多	10	23	33
	最少	1	13	5
領域數	總數	3	3	2
子領域數	總數	18	10	13
選取網站		united, budget, resy, spothero, expedia, us.megabus, marriott, ign, discogs, rottentomatoes, sports.yahoo, eventbrite, underarmour, bookdepository, tesla, apple, target, kohls	trip, shopping.google, tiktok.music	linkedin, allrecipes, fedex, ca.gov

表 4.2: Workflow Set 資訊

平均每 Set Workflow 數量	每 Set 最少 Workflow 數量	每 Set 最多 Workflow 數量	Workflow 總數量	領域數	子領域數	橫跨網站數
8.8	3	14	646	3	18	73

doi:10.6342/NTU202500827

所有實驗均採用「語意檢索」的 TaE 檢索策略，檢索數量 n 介於 $1 \sim 3$。此選擇基於 Synapse 與 AWM 均未探索 $n > 3$ 的設定，且 Synapse 團隊於 OpenReview 提及，為避免 LLM 因上下文過長而效能下降，選擇 $n = 3$ 作為上限。為此，我們比較 $n = 1, 2, 3$ 於不同 Workflow 數量設定下的影響。

Workflow 數量 M 則採費波那契數列進行設定：$M = 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89$，以平衡參數覆蓋與實驗成本，探索最具代表性的效能趨勢。

4.1.2 於 Cross-Task 之 SSR 表現分析

圖 4.1 顯示於 Cross-Task 任務設定中，當 Workflows 數量逐步提升時，各種 TaE 設定（TaE = 1, 2, 3）對 SSR 表現的影響趨勢。

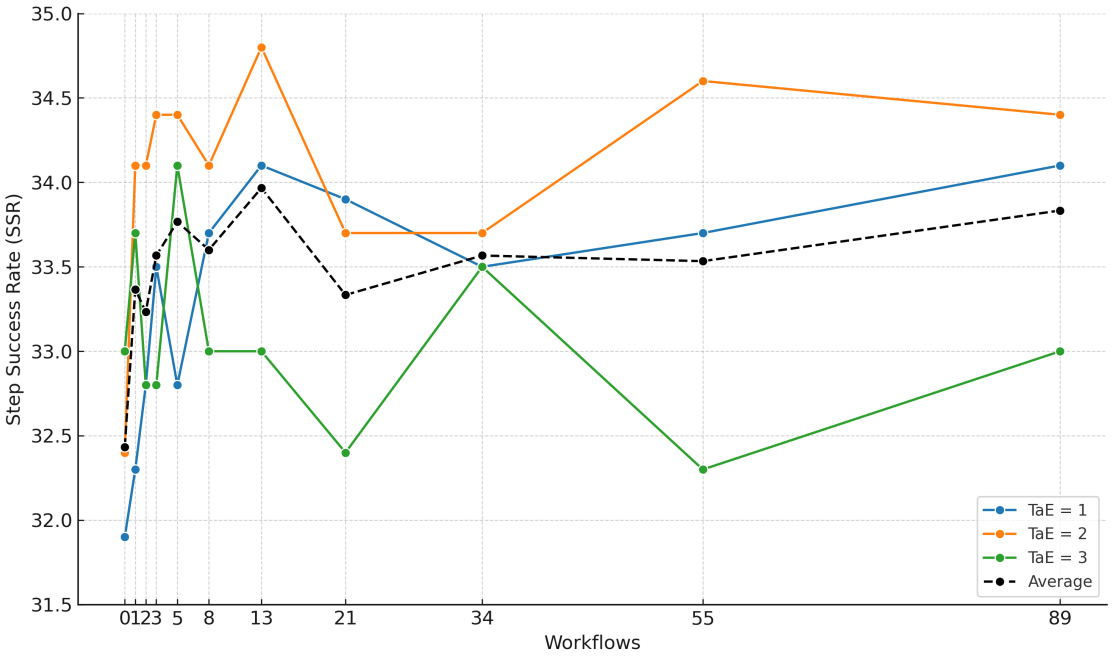

圖 4.1: 於 Cross-Task 任務情境中，當 Workflows 數量逐步提升時，各種 TaE 設定對 Step Success Rate 表現的影響趨勢

實驗結果： 如圖所示，在 Cross-Task 任務設定中，當 Workflow 數量從 0 增加至 13 時，三種不同的 TaE 設定（TaE = 1、2、3）皆有不同程度的表現變化。整體而言，TaE = 1 與 TaE = 2 在 SSR 表現上呈現明顯上升趨勢，然而，當 Workflow 數量超過 13 之後，SSR 未再出現穩定增長趨勢，甚至出現波動。而在 TaE = 3 的情況下，整體 SSR 表現波動較大，缺乏明確趨勢。

doi:10.6342/NTU202500827

結果分析： 上述結果顯示，在 TaE = 1 與 TaE = 2 的情境下，適量擴增 Workflow 有助於提升模型於 Cross-Task 任務下的決策效率。然而，當 Workflow 數量進一步擴增，效能不再穩定上升，顯示大幅擴增提示量未必持續帶來效能提升。

4.1.3 於 Cross-Website 之 SSR 表現分析

4.2顯示於 Cross-Website 任務設定中，當 Workflows 數量逐步提升時，各種 TaE 設定（TaE = 1, 2, 3）對 SSR 表現的影響趨勢。

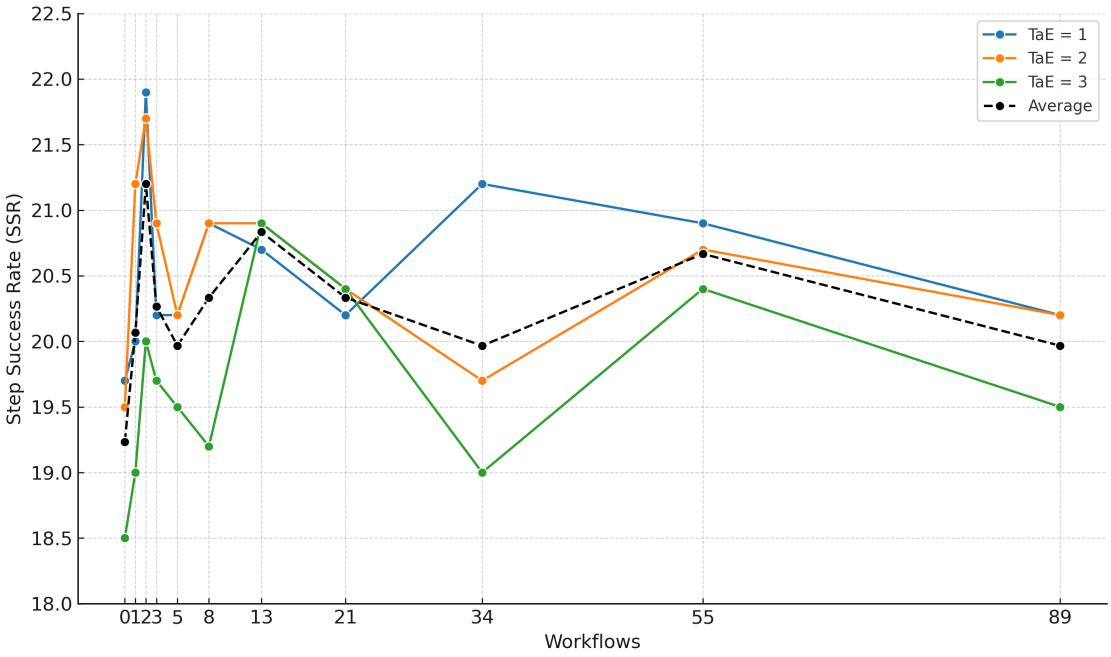

圖 4.2: 於 Cross-Website 任務情境中，當 Workflows 數量逐步提升時，各種 TaE 設定對 Step Success Rate 表現的影響趨勢

實驗結果： 整體而言，當 Workflow 數量由 0 增加至 2 時，TaE = 1 與 TaE = 2 呈現明顯的 SSR 提升，分別達到約 21.9 與 21.7 的峰值。隨後，兩者表現皆出現下滑及震盪，未再出現明顯提升。TaE = 3 則在 Workflow = 13 時達到最高點 20.9，但整體波動幅度相對較大。

結果分析： 觀察結果顯示，少量高相關性的 Workflows 有助於提升代理人在 Cross-Website 任務中的執行表現，尤其是在 TaE = 1 與 TaE = 2 的設定下。然而，當範例量繼續擴增時，整體 SSR 並未持續成長，甚至出現不穩定的下滑趨勢，顯示提示資訊量與效能之間並非線性正比。與此同時，TaE = 3 在 Workflow = 13 時出現較佳表現，顯示其在特定條件下仍有潛力，但整體穩定性相對較低。

doi:10.6342/NTU202500827

4.1.4 於 Cross-Domain 之 SSR 表現分析

4.3顯示於 Cross-Domain 任務設定中，當 Workflows 數量逐步提升時，各種 TaE 設定（TaE = 1, 2, 3）對 SSR 表現的影響趨勢。

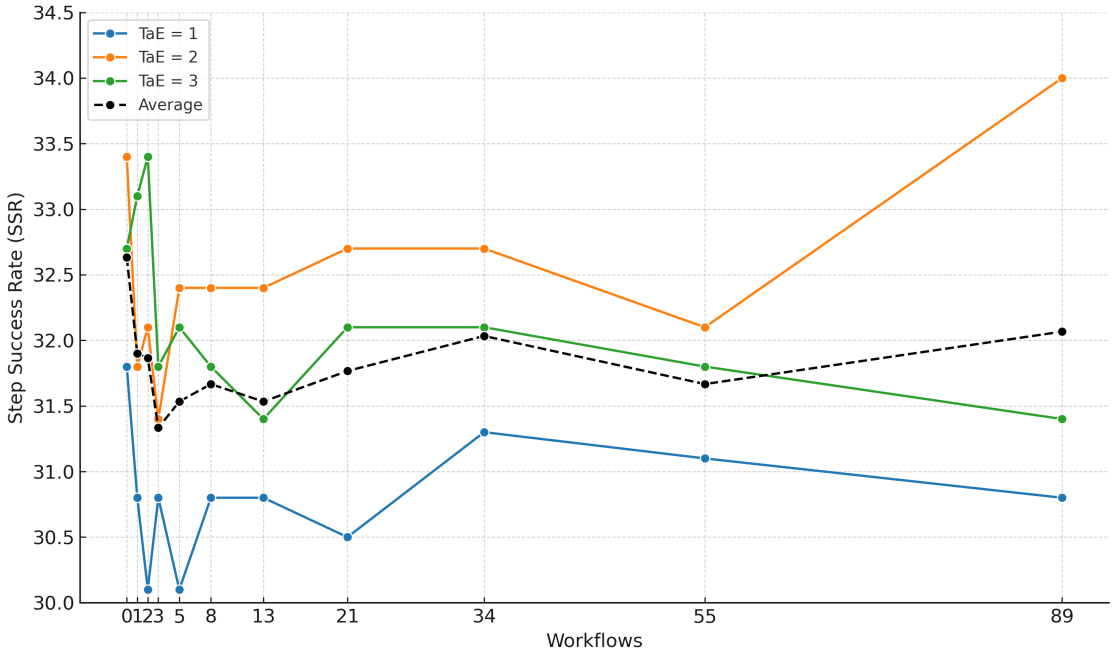

圖 4.3: 於 Cross-Domain 任務情境中，當 Workflows 數量逐步提升時，各種 TaE 設定對 Step Success Rate 表現的影響趨勢

實驗結果：整體來看，TaE = 1 與 TaE = 2 在加入少量 Workflows 後，SSR 有明顯下降趨勢，分別於 Workflow = 2 與 3 時達到最低點。雖然隨著 Workflow 數量增加，SSR 有小幅回升，但整體表現仍未超越未加入 Workflow 的基準點。TaE = 3 則在 Workflow = 2 時短暫達到局部高點，之後呈現持續下滑趨勢。

結果分析：上述結果顯示，在 Cross-Domain 的任務設定下，Workflows 對模型推理效果的正面影響相對有限。特別是在 TaE = 1 與 TaE = 2 的情況下，Workflow 可能因環境相關性太低，反而成為干擾來源，導致模型在決策時無法聚焦於當前任務需求。即便 TaE = 3 於特定 Workflow 數量下短暫提升表現，整體仍缺乏穩定性，顯示在高度異質任務間使用先前經驗的效果受限。

doi:10.6342/NTU202500827

4.1.5 不同任務情境下，Workflows 數量對 Step Success Rate 影響之表現

綜合 Cross-Task、Cross-Website 與 Cross-Domain 三種任務設定之結果（圖 4.1 至圖 4.3），可發現 Workflows 的提示效果與任務情境呈現高度相關。當訓練資料集合測試任務間的異質性愈高，增加 Worklfows 所能帶來的效益便愈有限，甚至可能成為干擾源。

為補充上述分析，圖 4.4 整合不同 TaE 設定下的 SSR 平均表現，進一步可視化 Workflows 數量變化於三種任務類型中的整體趨勢。圖中顯示，Cross-Task 呈現最穩定的正向成長趨勢，Cross-Website 的表現則較為震盪，而 Cross-Domain 在加入提示後整體表現仍低於基準，顯示提示資訊與任務環境落差過大時，易對推理造成干擾。此外，根據 [26]，即使輸入長度未達模型上限，若提示中混入大量無關資訊，仍可能使 LLM 推理效能下降。這再次說明，Workflows 數量的選擇應視任務異質性而定，並非愈多愈好。

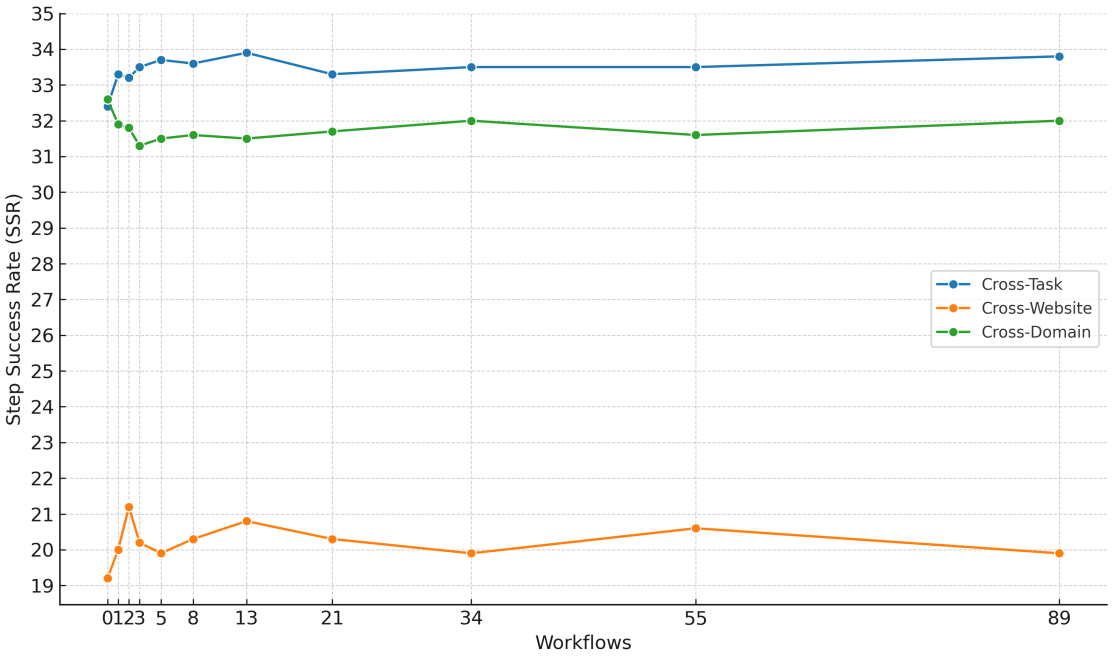

圖 4.4: 於不同任務情境中，Workflows 數量與 SSR 關係（各 TaE 平均）

最後需強調，本研究設計雖能觀察不同參數組合下的趨勢，但尚未針對最佳參數設定進行窮舉式搜尋，未來研究可進一步釐清不同任務情境下的最佳配置。

doi:10.6342/NTU202500827

4.2 MIMS 與現有方法之比較分析

為全面評估 MIMS 系統的效能，本節將其與目前代表性的兩種方法——Synapse 與 AWM—進行比較，涵蓋三種不同類型的任務泛化設定。

4.2.1 實驗設定與評估方法

為全面比較 MIMS 系統與現有方法在不同泛化任務下的表現，本節設計了一組完整且一致的實驗設定。實驗環境與第三章相同，惟為提升比較之代表性與可信度，本節不再僅採用部分測試資料，而是使用完整的測試資料集，以更準確地衡量各系統在實際應用情境中的效能差異。

本研究納入的比較對象 AWM，其原始論文中提及在 Cross-Website 與 Cross-Domain 設定下，Workflows 是從相同領域或整體資料集中隨機挑選。然而，作者並未說明實際的選取數量，亦未提供程式碼實作細節。為補足此不足，本研究根據 Workflow Set 中的平均數（8.8），選擇最接近的整數 9，作為 AWM 在本實驗中的預設檢索數量。

此外，AWM 採用的 Workflow「環境檢索」策略需依賴任務之 Metadata（如 Domain 與 Website）來判斷任務環境，進而進行相對應的 Workflow 選取。但在實際應用中，系統往往無法準確取得此類環境資訊，使得 AWM 難以有效應對陌生任務，也無法動態調整 Workflow 檢索數量，降低了系統的使用彈性。

相對而言，MIMS 採用語意檢索策略，不需依賴任何外部 Metadata，即可直接根據任務描述進行 Workflow 語意檢索。此一設計使系統可依任務需求自由調整檢索數量，提升了系統在實務中的彈性。

綜合前述設計與前章實驗結果，MIMS 在不同測試資料集下所採用的最終設定如下，係根據 EA 與 SSR 指標表現最優之參數組合所決定：

- Cross-Task：2 TaEs + 13 Workflows
- Cross-Website：2 TaEs + 2 Workflows
- Cross-Domain：2 TaEs + 89 Workflows

doi:10.6342/NTU202500827

4.2.2 實驗結果與分析

從表 4.3 可觀察到，MIMS 系統在三種任務泛化設定中皆展現出一定程度的優勢，尤以 Cross-Task 任務中的提升最為顯著：

- **Cross-Task**：MIMS 在 EA、SSR、SR 三項指標上皆優於 Synapse 與 AWM，分別達到 40.6、37.0 與 5.1，顯示透過語意檢索結合適量 Workflows 可有效強化模型對任務步驟的理解與執行能力。
- **Cross-Website**：MIMS 在 SSR 上表現最佳（26.1），但與 AWM（25.9）差距極小，EA 略低於 AWM，三系統在 SR 上則無差異（皆為 1.6）。此結果顯示，MIMS 雖具競爭力，但在跨站任務中提升幅度有限。
- **Cross-Domain**：MIMS 僅於 SR（2.1）與 AWM 並列最佳，其餘指標皆略低於 AWM，顯示在跨領域任務下，MIMS 所導入的基於語義檢索的 Workflows 雖具一定幫助，但提升效果有限，未能全面超越現有方法。

表 4.3: MIMS 和現有系統的比較

系統	TaE 檢索機制	Workflow 檢索機制	*Cross-Task*			*Cross-Website*			*Cross-Domain*		
			EA	*SSR*	*SR*	*EA*	SSR	SR	*EA*	*SSR*	*SR*
-	無	無	31.9	27.6	0	26.9	21.9	0	28	24.7	0
Synapse	語意檢索	無	39.3	36	4.3	30.8	25.2	**1.6**	32	28.5	2
AWM	環境檢索	環境檢索	38.7	34.6	3.1	**32.1**	25.9	**1.6**	**32.7**	**29.1**	**2.1**
MIMS	語意檢索	語意檢索	**40.6**	**37**	**5.1**	31.9	**26.1**	**1.6**	32.1	28.8	**2.1**

整體而言，MIMS 對 Cross-Task 任務帶來顯著提升，而在 Cross-Website 與 Cross-Domain 中雖無明顯優勢，仍展現出與既有系統相近的表現水準。此結果亦與前文分析一致：當測試任務與訓練資料間存在較大之**環境差距**（例如網站結構、互動方式或任務流程顯著不同）時，Workflows 所帶來的輔助效果將可能減弱，甚至產生干擾，導致效能下降。

doi:10.6342/NTU202500827

第五章　結論與未來展望

5.1　結論

本研究首先分析了現有代表性系統 Synapse 與 AWM，識別其在 Workflow 檢索策略上的潛在改進空間。基於此分析，我們提出一基於「語意檢索」的 Workflow 檢索機制。

透過在不同泛化任務設定下的實驗，本研究從 TaE 檢索數量與 Workflow 檢索數量兩個面向進行細緻比較，並找出可於各任務類型下取得相對良好效能的參數組合。結果顯示，所提出之 MIMS 系統在 Cross-Task 任務中展現出明顯優勢，能有效提升代理人在熟悉網站中處理新任務的能力；然而，在 Cross-Website 與 Cross-Domain 任務中，MIMS 的表現則與現有方法相近，僅在部分指標上略有提升，甚至在某些情境還有表現的下降，並未展現全面優勢。

整體而言，MIMS 展現了在語意檢索框架下整合 Workflow 的潛力，特別是在任務環境與訓練資料相對接近時。然而，當面對較為陌生的任務場景時，其效能仍受限，亦揭示了未來進一步優化 Workflow 檢索策略與參數設定的研究空間。

5.2　未來展望

儘管本研究成功提升了 LLM-Based Agent 在 Web Navigation 任務的效能，但仍有多項值得進一步探討與改進之處。以下列舉幾個未來可能的研究方向：

1. 探索多樣化的 Workflow 檢索策略：從實驗結果觀察，即使調整 Workflows 數量，系統效能並未呈現穩定且一致的變化趨勢。雖然本研究提出一種創新的語意檢索機制，但尚未與其他潛在的檢索方法進行全面比較。因此，未來

doi:10.6342/NTU202500827

研究可進一步探索不同類型的 Workflow 檢索策略，以尋求更具穩定性與泛化能力的檢索機制。

2. 最佳化參數搜尋策略的精進：本研究基於資源限制，採用費波那契數列作為 Workflow 數量的選取依據，藉此平衡探索範圍與實驗成本。然而，該方法在參數範圍擴展時會因參數變化不夠細緻，導致最佳參數組合難以被精確捕捉。未來可嘗試引入其他搜尋策略，以更有效率地探索參數空間，提升系統效能的穩定性與最終表現。

3. 動作目標判斷機制的強化：在做 Workflows 檢索的過程，我們需要先讓 Agent 預測 Action Objective，再根據該目標檢索適合的 Workflows。然而，若 Agent 預測的動作目標不正確，即使檢索邏輯本身是正確的，仍無法有效提升效能。因此，未來可嘗試設計動作目標驗證機制，以判斷 Agent 預測的 Action Objective 是否合理，進而提升檢索準確性。

4. Workflow 檢索機制的效率提升：由於動作目標相似性機制需額外執行動作目標預測，因此每次執行任務時，會比 AWM 或 Synapse 至少增加一倍的時間與計算成本。雖然本研究證明了該機制的效能提升，但其高昂的資源需求仍可能影響實際應用的可行性。若能改善，將會使此系統更佳的實用。

5. 在動態網頁環境上的適應性：本研究使用 Mind2Web 作為測試資料集，然而該資料集屬於靜態網頁快照，雖然已儘可能模擬真實網站情境，但仍缺少許多動態內容變化。在真實應用場景中，Agent 需要適應這些動態變化，因此未來可進一步探討如何讓 Agent 在動態環境下仍能有效檢索與決策，提升其在真實世界應用中的適用性。

doi:10.6342/NTU202500827

參考文獻

[1] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun, R. Kong, Y. Wang, H. Geng, J. Luan, X. Jin, Z. Ye, G. Xiong, F. Zhang, X. Li, M. Xu, Z. Li, P. Li, Y. Liu, Y.-Q. Zhang, and Y. Liu, "Personal llm agents: Insights and survey about the capability, efficiency and security," arXiv preprint arXiv:2401.05459, 2024.

[2] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su, "Mind2web: Towards a generalist agent for the web," Advances in Neural Information Processing Systems, vol. 36, pp. 28 091–28 114, 2023.

[3] L. Zheng, R. Wang, X. Wang, and B. An, "Synapse: Trajectory-as-exemplar prompting with memory for computer control," in The Twelfth International Conference on Learning Representations, 2023.

[4] Z. Z. Wang, J. Mao, D. Fried, and G. Neubig, "Agent workflow memory," 2024. [Online]. Available: https://arxiv.org/abs/2409.07429

[5] OpenAI, "Introducing chatgpt," November 2022, accessed: 2025-02-10. [Online]. Available: https://openai.com/index/chatgpt/

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[7] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu,

doi:10.6342/NTU202500827

R. Sauvestre, T. Remez et al., "Code llama: Open foundation models for code," arXiv preprint arXiv:2308.12950, 2023.

[8] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou et al., "The rise and potential of large language model based agents: A survey," Science China Information Sciences, vol. 68, no. 2, p. 121101, 2025.

[9] Apple Inc., "Siri – Apple," 2025, accessed: 2025-02-18. [Online]. Available: https://www.apple.com/siri/

[10] Amazon.com, Inc., "Alexa, the Voice Assistant," 2025, accessed: 2025-02-18. [Online]. Available: https://www.alexa.com/

[11] Microsoft, "Cortana support," 2025, accessed: 2025-02-18. [Online]. Available: https://support.microsoft.com/en-us/cortana

[12] Google Inc., "Google Assistant SDK RPC Reference," 2025, accessed: 2025-02-18. [Online]. Available: https://developers.google.com/assistant/sdk/reference/rpc?hl=zh-tw

[13] Apple Inc., "SiriKit Documentation," 2025, accessed: 2025-02-18. [Online]. Available: https://developer.apple.com/documentation/sirikit/

[14] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," arXiv preprint arXiv:2001.08361, 2020.

[15] Anthropic, "Claude (oct 8 version)," [Large language model], 2023. [Online]. Available: https://www.anthropic.com/

[16] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, Y. Bisk, D. Fried, U. Alon et al., "Webarena: A realistic web environment for building autonomous agents," arXiv preprint arXiv:2307.13854, 2023. [Online]. Available: https://webarena.dev

[17] S. Yao, H. Chen, J. Yang, and K. Narasimhan, "Webshop: Towards scalable real-world web interaction with grounded language agents," in ArXiv, preprint.

doi:10.6342/NTU202500827

[18] E. Z. Liu, K. Guu, P. Pasupat, T. Shi, and P. Liang, "Reinforcement learning on web interfaces using workflow-guided exploration," ArXiv, vol. abs/1802.08802, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:3530344

[19] P. He, X. Liu, J. Gao, and W. Chen, "{DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}," in International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=XPZIaotutsD

[20] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma et al., "Scaling instruction-finetuned language models," Journal of Machine Learning Research, vol. 25, no. 70, pp. 1–53, 2024.

[21] OpenAI. (2024, 7) Gpt-4o mini: Advancing cost-efficient intelligence. Accessed: 2025-04-07. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[22] ——. (2024, 5) Hello gpt-4o. Accessed: 2025-04-07. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[23] ——. (2023, 3) GPT-4. Accessed: 2025-04-07. [Online]. Available: https://openai.com/index/gpt-4/

[24] ——. (2023, 12) New and improved embedding model. Accessed: 2025-04-07. [Online]. Available: https://openai.com/index/new-and-improved-embedding-model/

[25] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The faiss library," arXiv preprint arXiv:2401.08281, 2024.

[26] M. Levy, A. Jacoby, and Y. Goldberg, "Same task, more tokens: the impact of input length on the reasoning performance of large language models," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 15 339–15 353.

doi:10.6342/NTU202500827