

你的專屬 AI 參謀： 以「生成式人工智慧」打造本地端兵棋推 演輔助系統

謝沛學

網路安全與決策推演研究所

壹、前言

美國公司 OpenAI 開發的 ChatGPT 橫空出世，其所展現對自然語言理解、生成與對話能力，於全球範圍內掀起了一場顛覆性的技術革命。在「生成式人工智慧」(Generative AI) 蓬勃發展的背景下，基於轉換器的「生成式預訓練技術」(Generative Pre-trained Transformers) 所開發的「大型語言模型」(Large Language Models, LLMs) 及其相關應用迅速成為全球矚目的焦點。由於出色的內容理解能力，能夠實現有效的上下文感知、語義分析和知識推理，「大語言模型」具有協助想定分析以及「軍事行動方案」(Course of Actions, COA) 生成與規劃，進而建構戰場決策輔助系統的潛能。然而，目前 LLMs 運用於想定及軍事行動方案生成，仍有幾個問題待解決。首先，受限於 LLMs 預訓練所使用資料的品質，使用 LLMs 回答某些特定專業領域的問題時，可能出現「答非所問」，甚至是編造錯誤的訊息，這個現象被稱為「產生幻覺」(Hallucination)。其次，部份 LLMs 雖然理解文本與生成回應的功能強大，但模型本身設有限制回答禁忌議題的安全模組，甚至是關鍵詞過濾器，這使得 LLMs 運用於軍事研究分析上有很大的受限。再者，必須連結網路在雲端架構使用的 LLMs 也不利於防範軍事研究特別是軍事行動方案等機敏訊息外流。

因此，本文以開源的 LLM 為基礎，建置能夠本地端、離線使用

的兵棋推演輔助系統，並透過「檢索增強生成」(Retrieval Augmented Generation, RAG) 與「提示詞工程」(Prompt Engineering) 技術，增強本地端 LLM 的功能。再以結合質化探討與量化模擬的方式，包括電腦兵棋模擬軟體「指揮：現代作戰」專業版 (Command: Modern Operations Professional Edition，以下簡稱 CPE)，檢視此兵棋推演輔助系統在想定分析以及軍事行動方案生成與規劃上的能力。

貳、本地端 LLM 部署實作

所謂「檢索增強生成」技術，類似於「開卷測驗」(open-book exams) 允許受驗學生從攜帶的參考資料找尋答案解題的概念，係透過連結外部知識庫的方式，讓大語言基礎模型從專業文本擷取與提問相符的內容進行回應。¹「提示詞工程」則是以設計過的指令語句，包括提供範例、限制回答範圍等，引導大語言模型更為精確地回應提問。²相較於「預訓練」(Pre-training)³與「微調」(Fine-tuning)⁴所需處理的龐大資料、參數量及消耗的巨額成本，透過「檢索增強生成」搭配「提示詞工程」的混合方式，使用者能夠在無需更動原

¹ Rick Merritt, “What Is Retrieval-Augmented Generation, aka RAG?” *NVIDIA*, January 31, 2025, <https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/>.

² Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal and Aman Chadha, “A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications,” *Arxiv*, March 16, 2025, arXiv:2402.07927v2.

³ 以 2023 年發佈的 GPT-4 為例，外界普遍估計該模型包含 1.7 trillion 個參數，並花費 OpenAI 超過 7 千萬至 1 億美金的成本訓練初始模型。參見 Vitalii Shevchuk, “GPT-4 Parameters Explained: Everything You Need to Know,” *Level Up Coding*, May 18, 2023, <https://levelup.gitconnected.com/gpt-4-parameters-explained-everything-you-need-to-know-e210c20576ca>; Emmanuel Ohiri & Richard Poole, “What is the Cost of Training Large Language Models?” *Cudo Compute*, May 12, 2025, <https://www.cudocompute.com/blog/what-is-the-cost-of-training-large-language-models>.

⁴ 至於「微調」，儘管在處理的參數量與耗費成本上，明顯比從零開始「預訓練」一個全新模型來得少，但傳統的「全微調」方式，處理一次仍可能需花費數萬美金的開支，將模型調校至符合使用者需求，所耗費的資金仍非中小型單位，或個別研究者能負擔。近來興起的「低秩適應」(Low-Rank Adaptation, LoRA) 微調技術，宣稱可以用僅數百到上千美元的預算完成大語言模型的調校，但這通常僅包含單次調校的預算，未考量其它隱藏成本，如資料的人工標記等。此外，這類宣稱可以用極低成本完成 LLM 微調的方式，通常支援相對小的模型、適用範圍有限，亦或需透過租用 GPU，將硬體成本轉嫁給雲端服務商的方式，即使能順利完成模型微調，也有機敏資料外洩的風險，不符合本研究宗旨。“What is the Cost of Fine-tuning LLMs?” *Dev Learning Daily*, July 2, 2025, <https://learningdaily.dev/what-is-the-cost-of-fine-tuning-llms-f5801c00b06d>.

有基礎模型參數的條件下，提升大語言模型在處理專業領域議題上的效能。⁵



圖 1-1、RAG 搭配 Prompt 的作業流程

資料來源：作者自行製圖。

圖 1-1 展示了「檢索增強生成」搭配「提示詞工程」的運作流程。這樣一個可用作電腦兵棋輔助系統的本地端 LLM，至少必須包含幾個要件。首先，我們需要一個預訓練過的大語言基礎模型，提供通用知識來源與使用者進行問答；其次，必須有一套框架，將這個基礎模型從原先在開發者的雲端系統，轉移到個人使用者的電腦進行本地端、離線的運作。再者，如前所述，「檢索增強生成」技術之所以有效，其關鍵在於提供大量專業知識文本，以外掛的方式補充預訓練基礎模型所欠缺的資訊。因此，我們還需要能將大量知識文本分拆成易於判讀的更小片段，並賦予文本內的單詞、字句一組「數字向量」(Vector) 的工具。這些文字向量在多維空間中的距離位置，

⁵ Matteo Fuoli, Weihang Huang, Jeannette Littlemore, Sarah Turner and Ellen Wilding, “Metaphor Identification Using Large Language Models: A Comparison of RAG, Prompt Engineering, and Fine-tuning,” *Arxiv*, September 29, 2025, <https://arxiv.org/abs/2509.24866>.

代表了文本內單詞、字句的意義和上下文關係。⁶透過把文字向量化處理，大語言模型才能「讀懂」或「判斷」以自然語言表達的外部知識文本。分拆與轉換成向量標記的文本，則會儲存在特定的「向量資料庫」(Vector Database)，供大語言基礎模型後續檢索調用。當然，我們還需要一套工具架構，將前述「外部文本分拆」、「詞句向量轉化」、「資料庫存儲與調用」等工作流程串接起來，組成兵棋輔助系統最為核心的要件。

此外，只靠傳統的中央處理器 (Central Processing Unit, CPU) 無法滿足大語言模型對於運算能力的需求，我們必須在建置的系統導入特定的套件，使這個本地端 LLM 能夠優先調用「圖形處理器」(Graphics Processing Unit, GPU) 進行運算。最後，為了避免必須在單調、純文字的「命令列介面」(Command Line Interface, CLI)，以輸入程式碼的方式操作本地端 LLM。我們也導入「使用者介面」(User Interface, UI) 套件，將建立好的兵棋輔助系統以網頁應用程式的外觀形式輸出，如同平時登入 OpenAI 的網頁使用 ChatGPT 一樣，讓使用者透過按鍵、選單、圖示等操作元素，以更為直觀的方式與兵棋輔助系統互動。圖 1-2 羅列本研究所採用的工具與套件清單，除了必須符合前述的各項預期功能之外，「開源」與「非中國系統」則是兩個最關鍵的選取標準。

⁶ Naheed Rayhan, and Md. Ashrafuzzaman, “LLM Enhancer: Merged Approach Using Vector Embedding for Reducing Large Language Model Hallucinations with External Knowledge,” *Arxiv*, April 29, 2025, <https://www.arxiv.org/abs/2504.21132>.

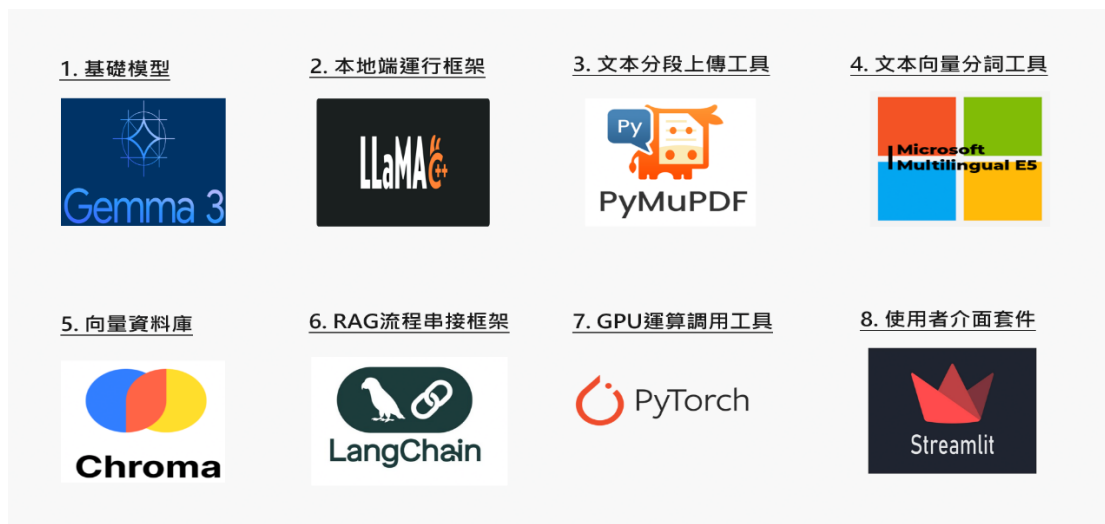


圖 1-2、本地端兵棋輔助系統使用套件

資料來源：作者自行製圖。

筆者在 Win 10 專業版、GPU: NVIDIA Geforce RTX 3060 6GB、CPU: Intel 12th Gen Core i7 12700H 以及 48GB 系統記憶體硬體設備條件下，以 Python 程式語言為基礎，搭建供本地端兵棋推演輔助工具運作的系統，並透過 Anaconda 平台導入與管理所需的函式庫 (library) 與套件 (packages)，搭建供 Python 作業的虛擬環境。附圖 1-3 展示了筆者所建置的本地端兵棋輔助系統的部份程式碼畫面。

```

File Edit Selection View ... Search
llm-rag-system-v7.py 9+
C:\Users\user\Wargaming LLM Projects> llm-rag-system-v7.py > get_rag_chain
1 import streamlit as st
2 import os
3 import sys
4 import re
5
6 from langchain_community.llms import LlamaCpp
7 from langchain_community.document_loaders import (
8     PyPDFLoader, Docx2txtLoader, UnstructuredExcelLoader,
9     CSVLoader, UnstructuredHTMLLoader, UnstructuredMarkdownLoader
10 )
11 from langchain.text_splitter import RecursiveCharacterTextSplitter
12 from langchain_huggingface import HuggingFaceEmbeddings
13 from langchain_chroma import Chroma
14 from langchain.chains import RetrievalQA
15 from PIL import Image
16 import pytesseract
17 from langchain.docstore.document import Document
18 from langchain.chains import LLMChain
19 from langchain.prompts import PromptTemplate

```

圖 1-3、本地端兵棋輔助系統部份代碼範例

資料來源：作者自行截圖。

圖 1-4 則是僅載入預訓練基礎模型 Google Gemma 3-12b-it-Q6-K-

L 的系統畫面，如果以此開始進行問答，則本地端兵棋輔助系統只會透過基礎模型的預訓練知識生成回應內容。如圖所示，系統的左側欄位分別有 Prompt 功能鍵與 RAG 功能鍵。根據筆者的規劃，使用者可以依需求決定是否啟用這兩個功能。這樣的設計允許我們分別測試「預訓練基礎模型」、「基礎模型+Prompt」以及「基礎模型+Prompt+RAG」這三種組合的能力。

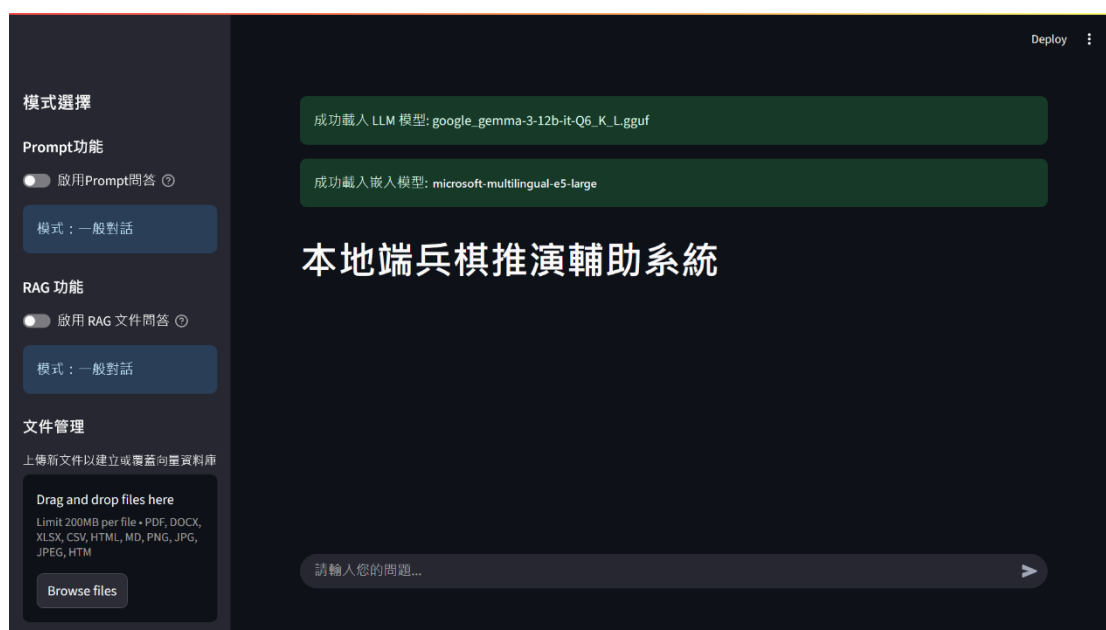


圖 1-4、本地端兵棋輔助系統（僅預訓練基礎模型）

資料來源：作者自行截圖。

下方圖 1-5 則展示了兵棋輔助系統同時開啟 Prompt 與 RAG 功能的畫面。使用者可以透過 RAG，上傳預先統整過的軍事文本、研究報告等資料，並由 PyMuPDF 與 Microsoft-Multilingual-E5 等套件，將這些文本內容分割與向量化處理，儲存至向量資料庫，作為本地端 LLM 的外部知識來源，補強預訓練基礎模型在軍事專業領域知識的不足。當使用者提出問題，LLM 的檢索機制會將詢問的內容與向量資料庫的文本內容作比對，擷取出與詢問相符合的資訊，作為後續生成回應的基礎。

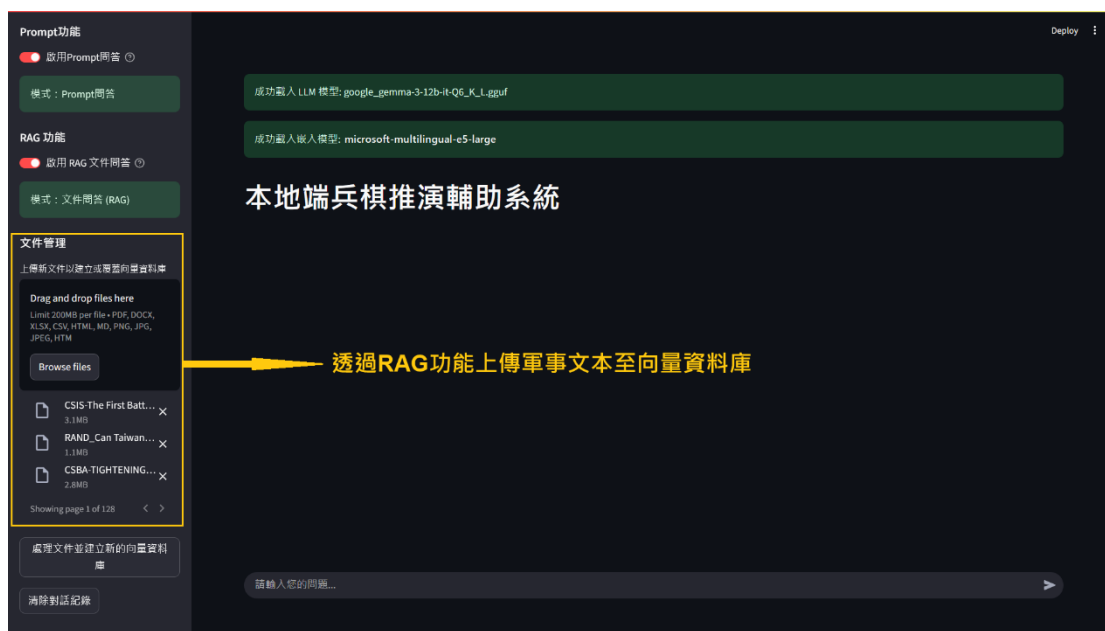
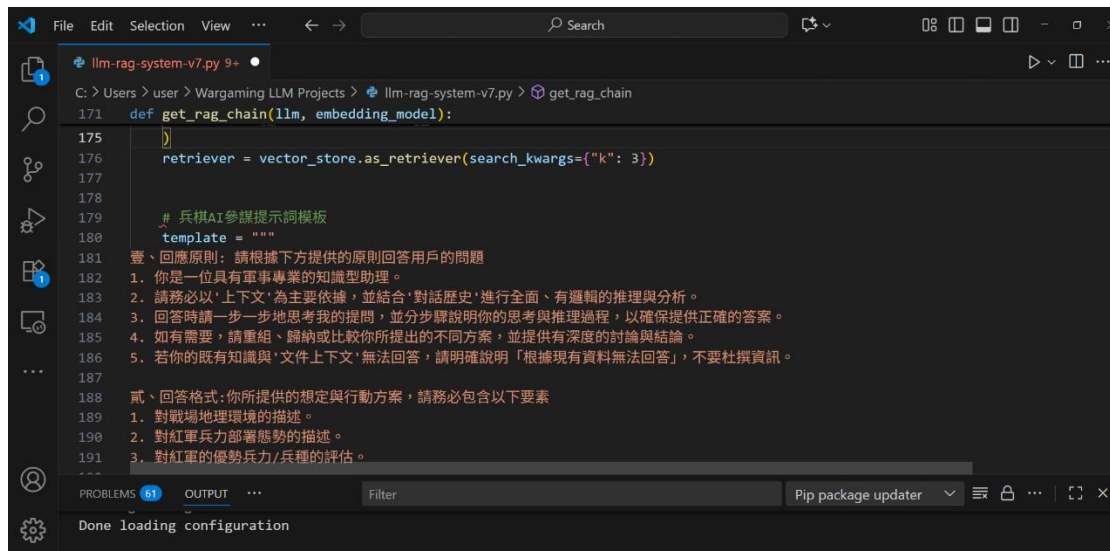


圖 1-5、本地端兵棋輔助系統（同時開啟 Prompt 與 RAG 功能）

資料來源：作者自行截圖。

筆者所收集的軍事領域專業文本資料分為三大類：首先是「官方報告」，例如美國國防部年度《中國軍力報告》、《中華民國國防報告書》等；其次是「智庫報告」，例如，美國蘭德智庫（RAND）、戰略暨國際研究中心（CSIS）、戰略和預算評估中心（CSBA）、日本防衛研究所（NDIS）等關於解放軍威脅及第一島鏈安全的研究報告；另外，專業的「軍事研究期刊」也被納入本研究 RAG 外部資料庫，特別是中國的《指揮與控制學報》、《系統仿真學報》等，部份文章以電腦兵棋與模擬為工具，探討與第一島鏈場景相關的作戰概念。

由於這些報告書、研究文獻皆是由官方單位或權威智庫 / 單位所發表，在公開正式出版前也有專業的審查機制過濾，可以大幅減少「資料清理」（Data Cleaning）的工作。筆者共搜集 128 份軍事研究專業文本，作為本地端兵棋輔助系統在想定與行動方案生成的外部補充知識來源之用。



```
File Edit Selection View ... Search
llm-rag-system-v7.py 9+
C:\Users\user> Wargaming LLM Projects > llm-rag-system-v7.py > get_rag_chain
171 def get_rag_chain(llm, embedding_model):
172     retriever = vector_store.as_retriever(search_kwargs={"k": 3})
173
174
175     # 兵棋AI參謀提示詞模板
176     template = """
177     壹、回應原則：請根據下方提供的原則回答用戶的問題
178     1. 你是一位具有軍事專業的知識型助理。
179     2. 請務必以'上下文'為主要依據，並結合'對話歷史'進行全面、有邏輯的推理與分析。
180     3. 回答時請一步一步地思考我的提問，並分步驟說明你的思考與推理過程，以確保提供正確的答案。
181     4. 如有需要，請重組、歸納或比較你所提出的不同方案，並提供有深度的討論與結論。
182     5. 若你的既有知識與'文件上下文'無法回答，請明確說明「根據現有資料無法回答」，不要杜撰資訊。
183
184     貳、回答格式：你所提供的想定與行動方案，請務必包含以下要素
185     1. 對戰場地理環境的描述。
186     2. 對紅軍兵力部署態勢的描述。
187     3. 對紅軍的優勢兵力/兵種的評估。
188     ...
189
190
191
...
PROBLEMS 61 OUTPUT ... Filter Pip package updater Done loading configuration
```

圖 1-6、本地端兵棋輔助系統「提示詞」模板部份畫面

資料來源：作者自行截圖。

至於「提示詞」(Prompt)的功能，如圖 1-6 所示，筆者以程式碼的方式設計為功能鍵，可以依使用需求開啟執行，而不是透過對話框輸入，以節省與 LLM 對話的上下文 Token 限制。例如，本研究所設計的「提示詞」模板之一，律定大語言模型的回應原則。當開啟「提示詞」功能後，使用者無需在每次提問重新要求，LLM 會自動依照模板所律定的原則進行回應。

一、 LLM 的專業助理角色

二、 分步驟思考與推理提問的問題

三、 說明分步驟思考與推論的理由

四、 比較不同方案的優劣，再總結答案

五、 如果沒有相關的資訊或知識，不得杜撰事實來當答案。

參、研究設計

由於本研究的目的是在於建置一套可供本地端部署、離線使用的大語言模型，並搭配「提示詞工程」以及「檢索增強生成」的外部資料庫，補強預訓練基礎模型在回應專業軍事領域上的不足，作為電腦兵棋在想定與行動方案生成上的輔助系統。因此，我們將分別

測試，「預訓練基礎模型」、「基礎模型+Prompt」以及「基礎模型+Prompt+RAG」這三種組合在想定與行動方案生成上的能力。

此處我們選擇中國著名的軍事雜誌《艦船知識》，於2020年7月所發表的一篇〈祖國統一之戰的仿真推演〉專題文章，作為測試大語言模型的「考題」。這篇文章描繪了解放軍透過一連串海空火力壓制與打擊，在開戰24小時內摧毀台灣的海空作戰能力，並於D+2日開始登陸作戰，D+20日左右順利攻佔北台灣。⁷姑且不論《艦船知識》的編輯與作者群策劃這期專題背後的認知作戰意圖，〈祖國統一之戰的仿真推演〉的內容包含了雙方兵力部署態勢、作戰排程與戰術行動方案等重要元素，再以電腦兵棋系統CMO進行模擬推演，確實提供了檢視大語言模型在想定生成輔助能力上的參考模板。⁸

《艦船知識》這篇文章的想定分為「火力打擊」、「強制隔離」、與「登陸作戰」這三大階段。前兩大階段屬於「海、空作戰」，其中，「火力打擊」階段的目的是為了摧毀守方的防空與反擊能力，奪取台灣周邊與上空的海、空優，為後續作戰行動開路，是整篇文章想定最關鍵的主軸，作者群透過電腦兵棋系統CMO進行了大量的模擬推演。「強制隔離」雖然也屬於「海、空作戰」階段，由於封鎖與隔離這類行動普遍缺乏「接戰行為」(Engagement)，難以透過電腦兵棋系統模擬得出相關數據。《艦船知識》文章作者群主要以文字論述搭配部份CMO場景的方式，「演示」(demonstrate)封鎖隔離行動的進程。至於「登陸作戰」階段，由於CMO的優勢在於海空場景的模擬，對於地面作戰的呈現仍有不足，《艦船知識》文章對「登陸作戰」的推演，改以電子地圖加上軍事兵種符號(Military Symbology)的方式，產製「示意圖」搭配文字論述的方式，呈現雙方交戰進程。⁹

⁷ 蘇磊、夕霧、趙四、克里斯、儲遇隆、安海督、亞山，〈祖國統一之戰的仿真推演〉，《艦船知識》，第7期，2020年6月，頁28-52。

⁸ CMO是CPE的商業版，僅供娛樂、教育使用，根據Matrix Games的使用者規範，不得將CMO用於軍事專業分析與公開發表。

⁹ 儘管〈祖國統一之戰的仿真推演〉的作者聲稱，「登陸作戰階段」是以「The Operational Art of

因此，本文在研究設計上參照《艦船知識》的文章，採用兼具質化探討與量化模擬分析的方式。首先，我們擷取「登陸作戰」階段的想定，要求大語言模型判斷解放軍可能的登陸地點，目的在於測試不同組合的大語言模型，對「作戰意圖判讀」上的能力差異。其次，由於《艦船知識》的文章省略了不少想定細節，我們擷取「封鎖與隔離」階段的部份想定，比較不同組合的大語言模型在場景細節生成能力上的差異。最後，我們擷取「火力打擊」階段的台灣北部場景，要求不同組合的大語言模型生成藍軍的反制行動方案，並透過電腦兵棋 CPE 進行蒙地卡羅模擬，檢視不同行動方案在反制紅軍攻擊上的效果。

關於「效益量測指標」(Measure of Effectiveness, MOE)，一般對大語言模型所進行的「基準測試」(Benchmarking)，通常包含「Time to First Token, TTFT」與「Inter-token Latency」這些與回應速度有關的指標。¹⁰由於本研究建置的本地端兵棋輔助系統係實驗性質的原型 (prototype)，受限於現有 GPU 等硬體條件，要求大語言模型必須以「毫秒」的速度來回應提問仍有困難。因此，我們以「產出內容的品質」為原則，衡量不同組合的大語言模型的效能。針對前述的第一個測試，本文比較大語言模型所判斷的紅軍作戰計劃，與《艦船知識》文章作者群所設計的作戰方案之間的差異；與原文的想定方案越接近者，表現越佳。第二個測試的 M.O.E，我們以「完整性」與「合理性」為標準，比較不同組合的大語言模型所補充的場景細節。第三個測試的 M.O.E，則是以「重要防護目標存活率」與「精準彈藥消耗量」，來探討不同組合的大語言模型所生成的藍軍反制行動方案之效益。

War, TOAW」這款電腦戰略遊戲進行推演，唯從該篇文章所呈現的推演畫面與 TAOW 操作介面及圖資明顯不符。

¹⁰ Vinh Nguyen, Wenwen Gao, Emily Apsey, Ganesh Kudleppanavar, Neelay Shah and Elias Bermudez, "LLM Inference Benchmarking: Fundamental Concepts," *Nvidia*, April 2, 2025, <https://developer.nvidia.com/blog/llm-benchmarking-fundamental-concepts/>.

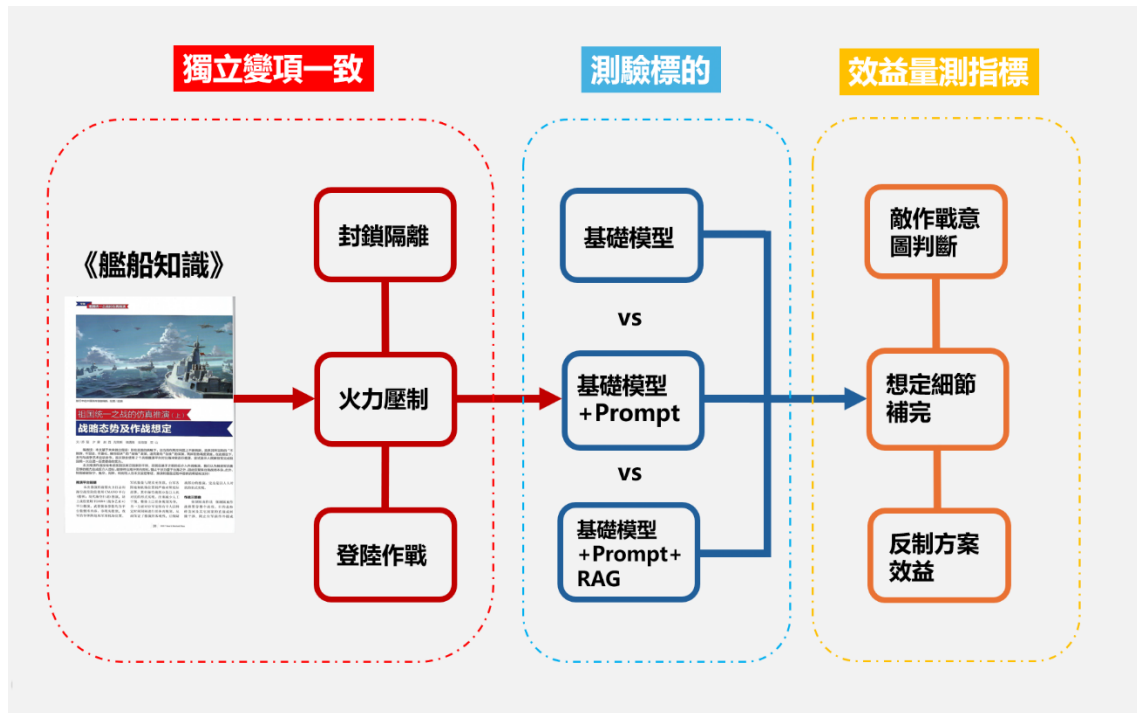


圖 1-7、本地端兵棋輔助系統「基準測試」設計流程

資料來源：作者自行製圖。

簡言之，如圖 1-7 所示，本文在控制「獨立變項」即想定場景一致的條件下，比較三組大語言模型在「敵作戰意圖判斷」、「想定細節補完」與「反制行動方案效益」能力上的差異。如果不對「想定場景」進行控制，則因為不同想定在敵威脅程度，以及紅藍雙方兵力部署等設定上的差異，將嚴重影響大語言模型的生成內容，所得出的模擬結果不具可比較性。

肆、個案探討

場景一：登陸作戰

我們此處擷取《艦船知識》文章對於 D+2 日後的想定場景，測試不同 LLM 工具組合對紅軍發動登陸作戰位置（北、中、南）的判斷，以及與《艦船知識》文章想定的方向吻合程度，探討三種 LLM 組合對於解放軍在「登陸作戰意圖判讀」上的能力差異。根據《艦船知識》文章的想定，紅軍有兩支登陸部隊，作者群設定台灣北部

為登陸主戰場，由配署 3 個陸軍集團軍、2 個海軍陸戰旅、2 個空降旅及東部戰區相關單位的主力部隊負責進攻；另一支配署 2 個陸軍集團軍、2 個海軍陸戰旅及 2 個空降旅的預備隊，則承擔佯攻任務，伺機從澎湖與中、南部進犯，牽制其它藍軍部隊北上增援。

我們進一步提供《艦船知識》文章關於台灣三個作戰區的兵力部署，並要求 LLM 結合對台灣北、中、南區域地理條件的知識，以參謀的角色給予中央軍委登陸作戰方案的建議。測試結果發現，「僅有預訓練基礎模型」的 LLM，除了生成的內容相對簡化，缺乏系統性的比較，其所建議的行動方案是全力直攻北部，且沒有任何預備隊與佯攻牽制的規劃。此外，儘管「僅預訓練基礎模型」選擇北部進行登陸，似乎與《艦船知識》文章的想定相符，但其理由是南部有高聳的斷崖和複雜的地形，且南部兵力部署比北部密集，明顯產生所謂的幻覺。



圖 1-8、測試場景一「僅預訓練基礎模型」的回應截圖

資料來源：作者自行截圖自測試畫面。

當我們啟用 Prompt 功能，測試「基礎模型+Prompt」的組合，其所生成的內容明顯較有邏輯層次，會進行不同方案的優劣比較。

儘管「基礎模型+Prompt」組合的建議方案為「登陸南部」，且沒有搭配佯攻行動，但其所提供的理由確有其合理之處：「兩棲登陸作戰初期即便成功，若無法確保後續重裝備與補給上岸，灘頭陣地也將迅速被敵方優勢火力摧毀。奪取高雄港建立可持續的後勤保障為前提，穩中求進，逐步瓦解敵方防禦的務實選擇……雖然耗時可能較長，但在面對台灣的「刺蝟」防禦戰略時，生存能力和最終的成功率可能最高」。



圖 1-9、測試場景一「預訓練基礎模型+Prompt」組合的回應截圖

資料來源：作者自行截圖自測試畫面。

當我們進一步啟動 RAG 功能，測試「基礎模型+Prompt+RAG」的組合，此時 LLM 所提供的行動方案與《艦船知識》文章的想定基本吻合。能夠「速戰速決」是其最主要的決策理由，雖然北部防禦嚴密，但 LLM 判斷仍可透過集中優勢兵力重點突破，再搭配預備隊於中部的佯攻，從而減輕北部主攻方向的壓力。「此舉亦可為後續作戰提供靈活性，若北部進展順利，中部部隊可轉為鞏固佔領區；若北部受挫，中部則可視情況轉為主攻，或作為撤退的備用方案」。至於南部地區，雖然地形較為利於機械化部隊開展，但 LLM 認為「航

渡距離最遠，被半途打擊的風險最大，且登陸後還必須一路北上，才能攻佔政經中樞」，因此沒有選擇從南部派遣登陸部隊的方案。

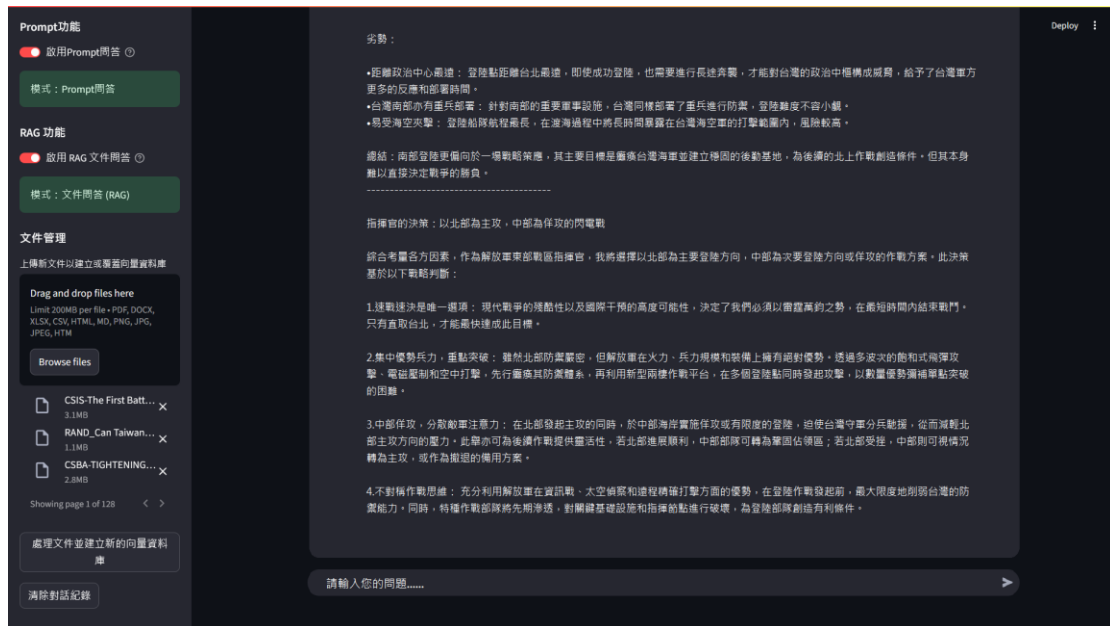


圖 1-10、測試場景一

「預訓練基礎模型+Prompt+RAG」組合回應截圖

資料來源：作者自行截圖自測試畫面。

場景二：封鎖作戰

我們此處擷取《艦船知識》文章對於紅軍完成飽和火力打擊，取得海、空優之後，開始對藍軍進行封鎖隔離的想定場景。根據《艦船知識》文章，此時紅軍於台灣海峽南、北兩端各部署一支水面艦隊編隊。北部由遼寧號航艦為首，包含 055 型驅逐艦、052 型驅逐艦與 054A 型護衛艦等共 16 艘所組成的編隊；南部則是由山東號航艦領銜，包含 052 型驅逐艦與 054A 型護衛艦等共 16 艘所組成的編隊。兩支水面艦隊編隊已經穿過南北水道，進入台灣東部海域展開封鎖行動，阻絕任何可能對藍軍的援助。《艦船知識》文章的想定亦強調，必須在奪取海空優之後，解放軍航艦戰鬥群才會進入台灣東部海域，並以「情監偵」、「電磁攻擊」、「飽和火力打擊」、「水下攻防」等手段為航艦戰鬥群創造機會。

我們先給予 LLM 關於紅軍兩支水面艦隊編隊的兵力配置，並依序要求不同組合的 LLM 進行作戰規劃，以遂行將兩支海軍艦隊，從南北兩端駛入台灣東部海域展開封鎖，阻絕其它國家對藍軍援助的戰略目標。這個場景設計的目的在於，測試三種不同組合的 LLM，對兩支艦隊完成航行至台灣東部海域部署、開始進行封鎖之前的想定細節的補充與生成能力。

測試結果發現，「僅有預訓練基礎模型」的兵棋輔助系統，在利用無人機、衛星和水面艦艇進行全面情報收集，以及發動大規模的電子戰行動，干擾台灣的通訊和雷達系統後，即下令兩個航艦打擊群強行進入台灣東部海域展開警戒、偵查，與構築海上封鎖線。與《艦船知識》文章的想定相比，「僅有預訓練基礎模型」的組合缺少了「飽和火力打擊」與「水下攻防」這兩個作戰行動準備。不僅在想定細節補足與內容生上有過度簡化之虞，在沒有取得海空優勢便大規模出動艦隊的方案也不甚合理。



圖 1-11、測試場景二「僅預訓練基礎模型」的回應截圖

資料來源：作者自行截圖自測試畫面。

當我們再次啟用 Prompt 功能後，測試「基礎模型+Prompt」的

組合，其所提出的作戰準備規劃包括了部署大量高空長航時無人機和中低空無人機進行持續偵察，填補衛星監控的盲區，並利用電子戰飛機和艦載設備，持續干擾台灣的雷達、通訊和導航系統等。「基礎模型+Prompt」組合的兵棋輔助系統亦提出對藍軍地面重要設施進行火力打擊，唯目標僅針對台灣東岸的重要設施如佳山、志航機場等。與《艦船知識》文章的想定相比，「基礎模型+Prompt」組合所生成補充的場景細節差異不大，基本上滿足了「情監偵」、「電磁攻擊」、「飽和火力打擊」、「水下攻防」這四個作戰準備要素，只是「火力打擊」手段不夠全面。

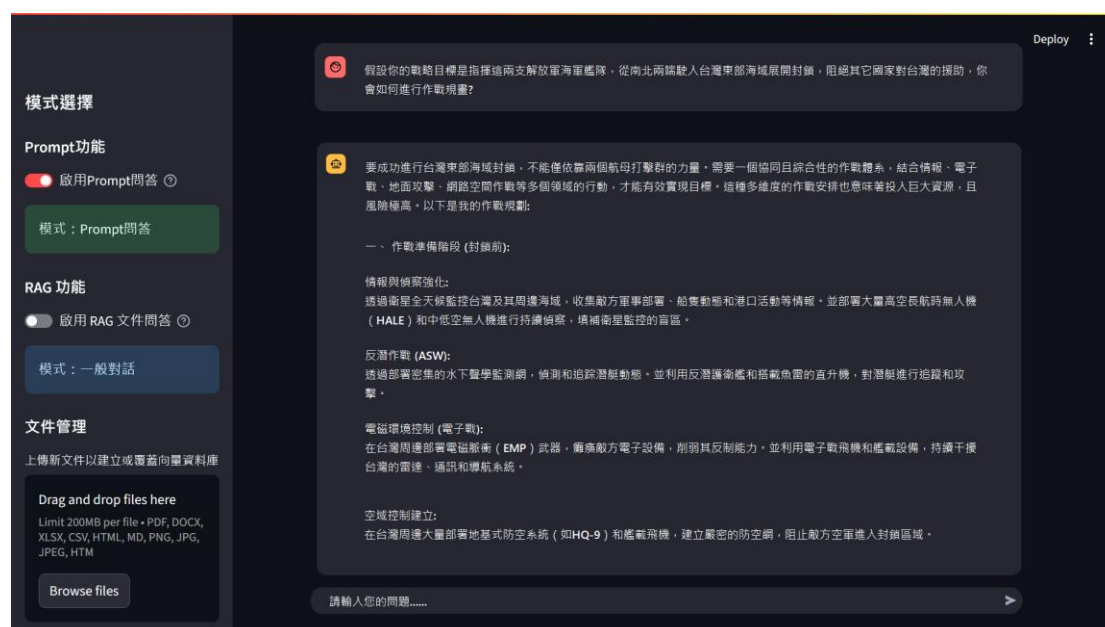


圖 1-12、測試場景二「預訓練基礎模型+Prompt」組合的回應截圖

資料來源：作者自行截圖自測試畫面。

當我們進一步啟動 RAG 功能，測試「基礎模型+Prompt+RAG」的組合，LLM 所生成的行動方案，不僅滿足《艦船知識》文章的 4 個想定要素，還補充了額外的細節。「基礎模型+Prompt+RAG」組合的輔助系統判斷，要成功進行台灣東部海域海上封鎖，不能僅依靠兩個航母打擊群的力量，而是需要「一個協同且綜合性的作戰體系，結合情報、電子戰、地面攻擊、網路空間作戰和後勤保障等多

個領域的行動，才能有效實現目標。此外，還必須預先應對國際社會的反應及潛在干預者的可能行動，並做好相應的準備」。與《艦船知識》文章的想定相比，除了同樣強調必須以多重作戰準備手段，奪取海空優後才能出動艦隊至東部海域，更建議攻佔東沙島作為山東號戰鬥群的中繼補給點（後勤考量），以及先發制人打擊美、日基地，癱瘓周邊美軍快速介入的能力（預應國際干預）。

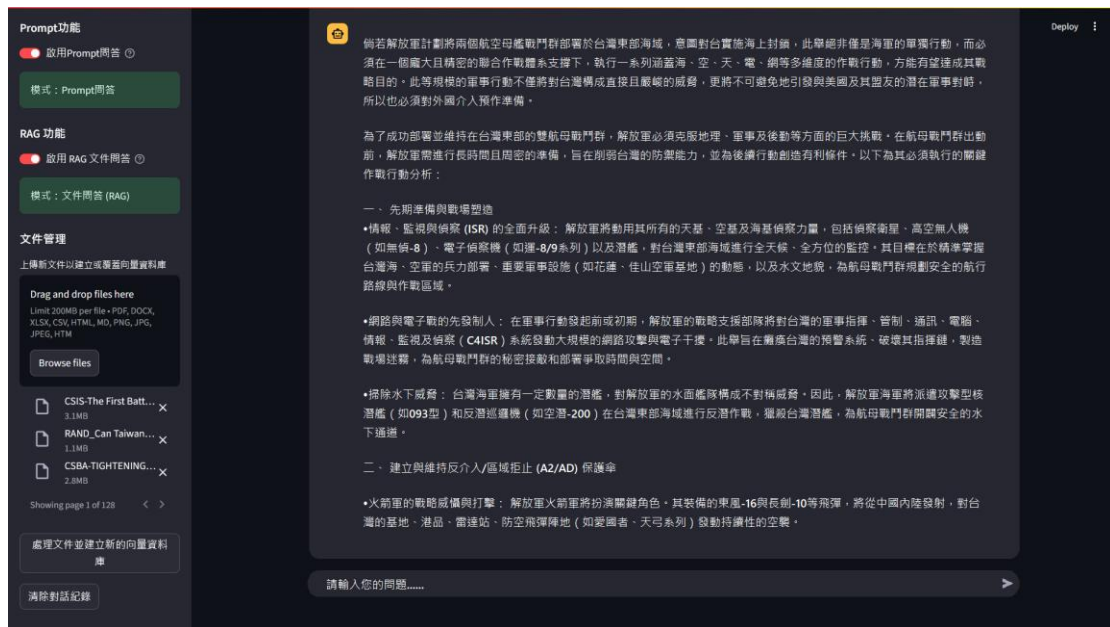


圖 1-13、測試場景二「預訓練基礎模型+Prompt+RAG」組合的回應截圖

資料來源：作者自行截圖自測試畫面。

場景三：飽和火力打擊

此處以《艦船知識》文章關於紅軍以飛彈飽和攻擊，壓制藍軍防空與作戰能力的想定為基礎，設計一個規模較小、場景限縮在台灣北部的想定，以利後續的模擬測試。我們給予藍軍 35 個重要防護目標，以及 4 種防空系統（愛國者、天弓、新型野戰、復仇者），共 520 枚精準彈藥。紅軍則是有 4 個東風 16 發射旅、216 枚彈道飛彈，以及 3 個長劍 10 發射旅、324 枚巡弋飛彈，總共 540 枚飛彈。場景設定紅軍分 3 個波次，對藍軍重要防護目標進行打擊。這個場景設

計的目的在於，結合電腦兵棋 CPE 測試三種不同組合的 LLM 所生成的藍軍防空反制方案的作戰效益。由於「效益量測指標」包含了「設施存活率」與「精準彈藥存量」，在測試的過程，我們以 Prompt 的形式，給予 LLM 不同反制選項進行方案組合。這些選項包括了防空陣地是否開機、接戰威脅種類（彈道 vs.巡弋）、接戰原則（SSL vs. SLS）、主射向線角度等。不同的選項組合對於「設施存活率」與「精準彈藥存量」將有不同的影響。例如，選擇防空陣地全開機，必然有利於重要設施的存活，但精準彈藥可能因此消耗的更快；不論來襲威脅種類（超音速或次音速），一律採取「漣波射擊原則」（Shoot-Shoot-Look, SSL），有助於提升威脅攔截率與重要設施的存活，但勢必不利於精準彈藥存量。



圖 1-14、測試場景三「本地端兵棋輔助系統」回應截圖

資料來源：作者自行截圖自測試畫面。

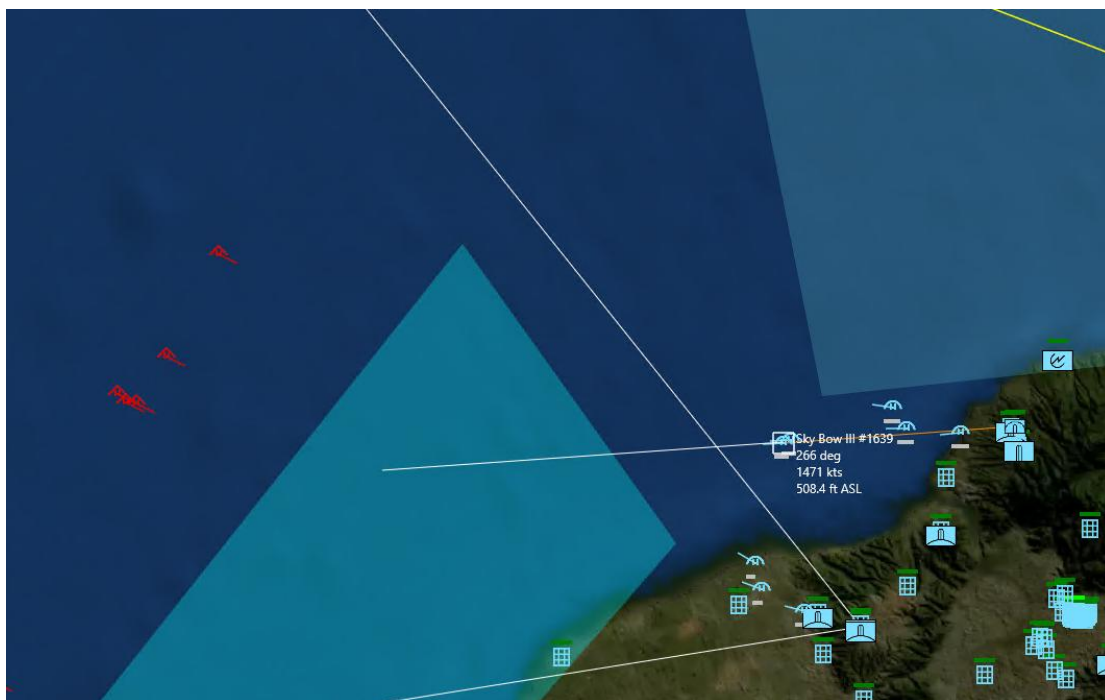


圖 1-15、測試場景三藍軍防空作戰-1

資料來源：作者自行截圖自模擬畫面。

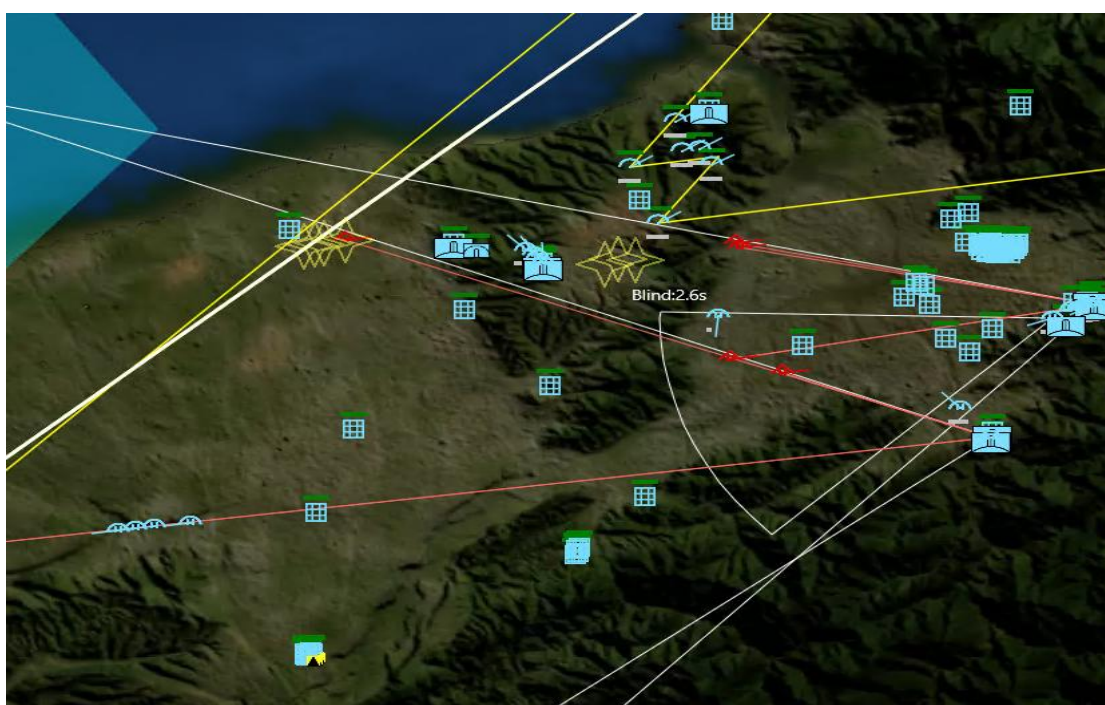


圖 1-16、測試場景三藍軍防空作戰-2

資料來源：作者自行截圖自模擬畫面。

測試結果顯示，由於少了不同方案選項的提示詞，亦缺乏外部專業軍事文本資料庫的參照，「僅預訓練基礎模型」的系統採取了最

呆板、單調的方案，給出了「火力全開、戰力發揚」的建議。儘管藍軍的重要防護設施在第一波攻擊的生存率最好，隨著精準彈藥存量在第二波次攻擊時耗盡，藍軍的重要設施存活率大幅下降。當我們啟動 Prompt 功能，由於在提示詞設定有不同方案組合可以嘗試，「預訓練基礎模型+Prompt」的輔助系統似乎把「以不同反制選項進行組合生成新方案」視為準則，採取的方案明顯偏向「戰力保存」，透過部份防空系統不開機、「多次單發射擊原則」(Shoot-Look-Shoot, SLS) 等組合，將精準彈藥存量延續至三個波次攻擊完畢，但付出的代價是重要防護設施的存活率。最後，當我們進一步開啟 RAG 功能，這個組合的 LLM 採取了「戰力發揚為主、戰力保存為輔」的策略。而由於有軍事專業文本當外部資料庫，「預訓練基礎模型+Prompt+RAG」的輔助系統，在反制方案的選擇更為靈活，例如，指派部份「復仇者」防空飛彈系統加強保護如愛國者與天弓防空系統，而非平均分配給其它的重要防護如港口、雷達等。這樣的選擇組合提升了藍軍中遠程防空飛彈系統的存活，延長了愛國者與天弓系統的接戰能量。

伍、結語

本研究結合 Prompt 與 RAG 技術，建置一套兵棋推演輔助系統。經測試，該方案確實能提升預訓練基礎模型在回應軍事專業議題的能力。而本地端、離線使用的特點，亦可解決作戰想定、行動方案等機敏資料外洩的疑慮。筆者未來除了硬體升級與程式碼調校，也將透過擴大軍事專業文本資料庫，以及自建軍事領域的中文分詞資料庫的方式，提升這套輔助系統的效能，後續可作為本院 War Room 團隊，不論是電腦兵棋或桌上兵推的想定與行動方案生成輔助工具。

本文作者謝沛學為美國內布拉斯加大學林肯分校政治學博士，任職於國防安全研究院，現職為網安與決策推演所副研究員，研究領域為兵棋推演、模式模擬、軍備競賽、地緣政治、國防經濟。

AI Staff Officer: A Generative AI-Based Local Wargaming Assistance System

Dr. Pei-Shiue Hsieh

Division of Cyber Security and Decision-Making Simulation

Abstract

This study aims to address the challenges of applying Large Language Models in the military domain, including issues such as hallucination, content restrictions, and the risk of sensitive data leakage. To this end, the author has developed a wargaming assistance system based on an open-source LLM, capable of operating locally and offline. This system integrates Retrieval-Augmented Generation and Prompt Engineering techniques to enhance the LLM's performance on specialized topics without modifying the base model's parameters. Using a Taiwan invasion scenario from the Chinese military magazine *Naval and Merchant Ships* as a test case, this paper compares the capabilities of three configurations: the base model alone, the base model with Prompt Engineering, and the base model with both Prompt Engineering and RAG. The comparison assesses performance in areas such as interpreting operational intent, supplementing scenario details, and the effectiveness of counter-plans, utilizing computer wargaming software for quantitative simulation and analysis. The results indicate that the LLM configuration integrating both Prompt Engineering and RAG achieved the best performance. This system will serve as an auxiliary tool for the War Room team at the Institute for National Defense and Security Research for wargaming tasks, with its performance to be continually enhanced through database expansion and code optimization.

Keywords: LLM 、 Locally-deployed 、 RAG 、 Prompt 、 Scenario Generation