

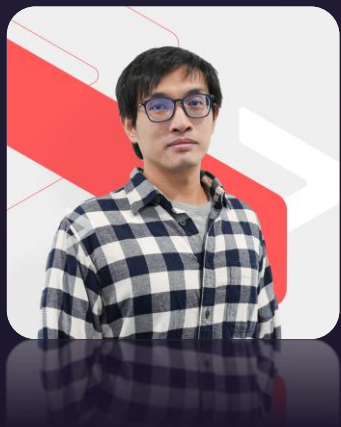
CLinker

精煉 LLM 打造具可擴展性的命令列圖譜系統

An Efficient, Distilled LLM-powered
Command-line Graph Constructor



Speaker – 林哲宇 Jerry



- > 資深主任資料科學家
- > 研究方向
 - > 圖論相關演算法在 Active Directory 中的應用
 - > 威脅情報分析相關演算法
- > 演講經歷
 - > EMNLP
 - > TROOPERS
 - > SINCON



Speaker – 鄭允臻 Laura



- > 國立台灣大學資工系三年級
- > 資料科學組實習生
- > 研究方向
 - > Transformer 在語言模型，圖，與音樂檢索應用
 - > 混合實境提升技能訓練
- > 演講經歷
 - > SINCON



Outline

- > Introduction
- > Methodology
- > Results
- > Takeaway

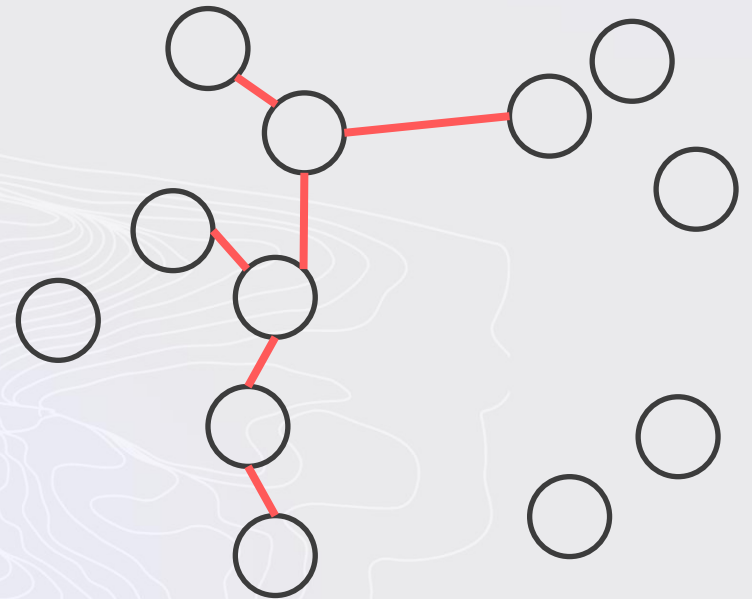


Introduction

資安鑑識調查

偵測命令列事件中的惡意行為

- > 資料：事件 (Event) = 命令列 (command line) 紀錄 + 額外資訊
 - > 程序 ID (process ID)
 - > 使用者 (user account)
 - > 時間戳 (timestamp)
 - > ...
- > 任務：辨別惡意事件
 - > 上下文**無關**的惡意
 - > 上下文**相關**的惡意





資安鑑識調查

上下文的重要性

> 系統內建工具：dir
> **dir C:\tmp**

正常？

> 使用者自訂檔案

> C:\Windows\system32\cmd.exe /c ipconfig /all <NUL
>C:\Windows\TEMP\Syste-1171671231238614172400000000-a13724
2>C:\Windows\TEMP\Syste-117167123123614272400000001-a13724

惡意？



資安鑑識調查

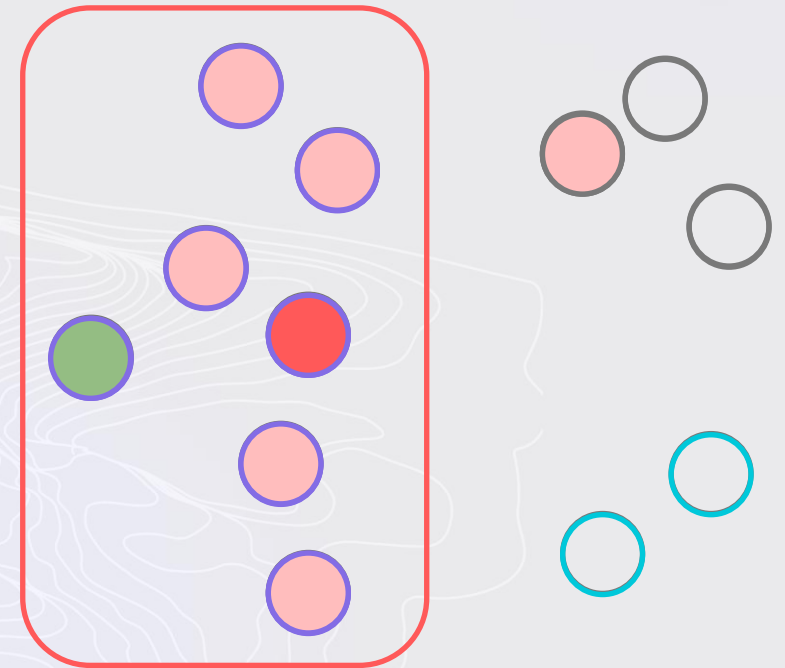
領域知識的重要性

- > `reg save HKCU D:\backup\hkcu.hiv`
 - > 當前使用者設定
- > `reg save HKLM\SAM D:\backup\sam.hiv`
 - > 本機帳戶安全資訊 (e.g., 所有使用者的密碼雜湊值)

資安鑑識調查

語言模型的使用 – 被動式調查

- > 事件 (events) → 聚類 (clusters)
- > 事件 (events) → 聚類 (clusters)
→ 惡意事件聚類解釋 (LLM)
- > 大型語言模型的能力，並未用在主動關聯潛在惡意 (聚類) 的過程



資安鑑識調查

語言模型的使用 – 主動式調查

> ~~一次全部事件~~

> 語言模型的上下文視窗 (context window) 有限

> “cmd” /c cd /d “C:\inetpub\wwwroot\”&move
c:\tmp\rar.rar D:\MyWebsite\ISO\CSS\&echo
[S]&cd&echo [E] (**46 tokens, Llama3**)



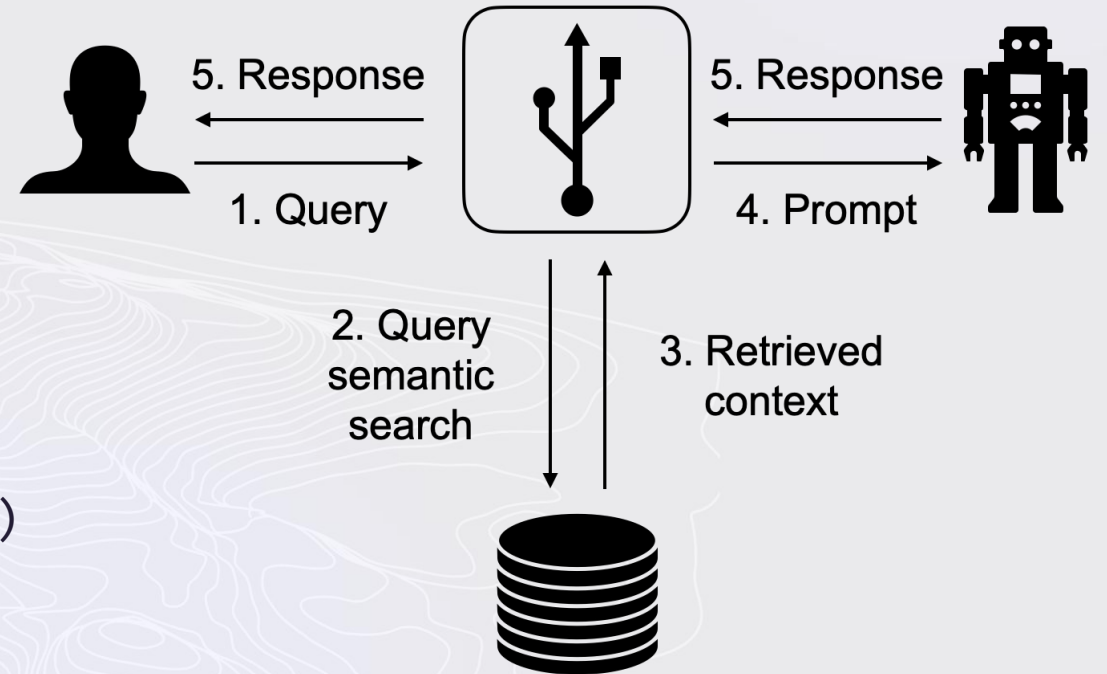
資安鑑識調查

語言模型的使用 – 主動式調查

- > 語言模型 context window 限制：
 - > 一次只能調查一部分事件
- > 上下文對事件判斷的重要性：
 - > 需要在聚類方法之外，進一步發揮語言模型的能力，更精準的取得待調查事件**相關的上下文**

RAG 與資安鑑識調查

- > RAG (Retrieval Augmented Generation)
 - > 檢索 – context = retriever(query)
 - > 生成 – response = LLM(query, context)
- > RAF (Retrieval Augmented Forensic)
 - > 檢索 – context = retriever(cand_events)
 - > 生成 – label = Judge(cand_events, context)
- > 有什麼不同？





RAG 與資安鑑識調查

> RAG

- > 檢索且只檢索語意上對回答查詢有幫助的內容

> RAF

- > 多面向檢索
- > 迭代的檢索/判斷過程



多面向檢索

- > Process tree
- > User
- > Timestamp
- > Filename
- > IP
- > Command pattern
- > ...



多面向檢索

```
"cmd" /c cd /d "C:\inetpub\wwwroot\"&move c:\tmp\rar.rar D:\MyWebsite\ISO\CSS&echo [S]&cd&echo [E]
```

```
"cmd" /c cd /d "C:\Windows\Temp\tmp1FA3.tmp\"&ping -a -n 1 10.3.76.202&echo [S]&cd&echo [E]
```

```
"cmd" /c cd /d "C:\inetpub\wwwroot\"&netstat -an|find "123.123.1.95"&echo [S]&cd&echo [E]
```

```
"cmd" /c cd /d "C:\Windows\Temp\tmp1FA3.tmp\"&ipconfig&echo [S]&cd&echo [E]
```

```
"cmd" /c cd /d "C:\tmp\"&netstat -an|find "123.123.1.95"&echo [S]&cd&echo [E]
```

```
"cmd" /c cd /d "C:\tmp\"&netstat -an|find "80"&echo [S]&cd&echo [E]
```

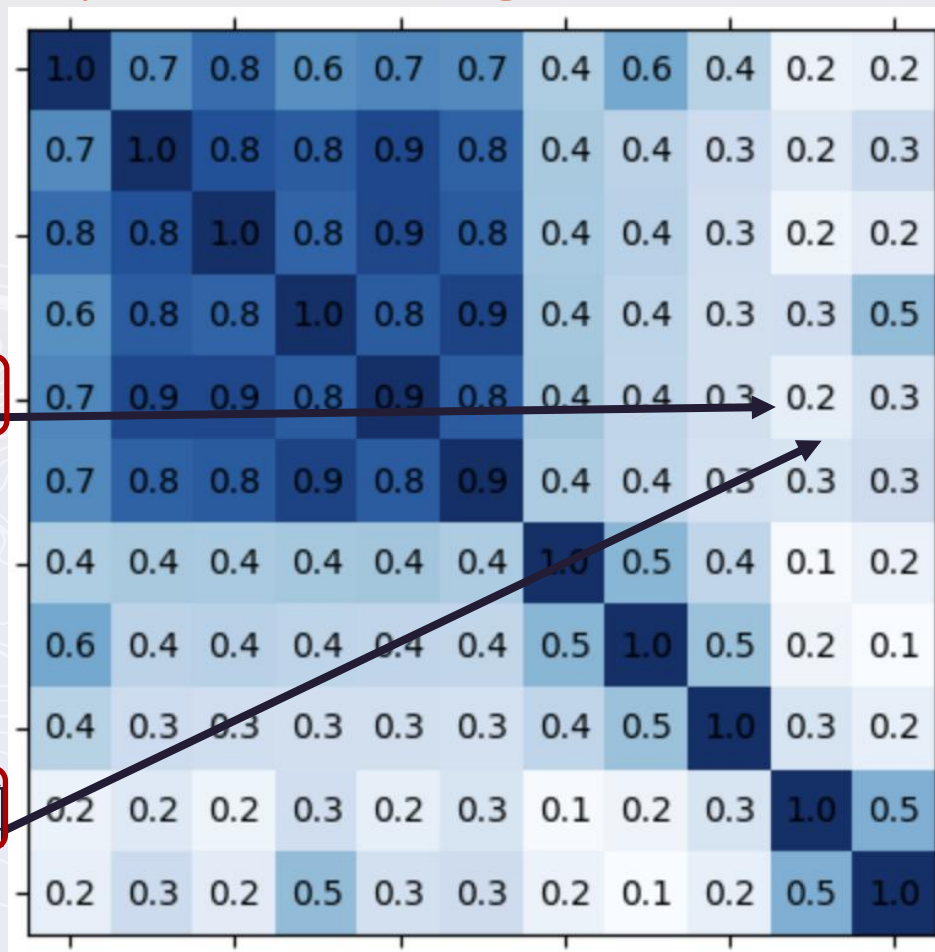
```
C:\tmp\ADExplorer\...&echo [S]&cd&echo [E]
```

```
fmgr 123.123.1.95 80
```

```
fmgr 123.123.1.95 80
```

```
find "123.123.1.95"
```

openai/text-embedding-3-small



迭代的檢索/判斷過程

> 前一筆判斷結果 ⇒ 下一筆判斷依據

1. "cmd" /c cd /d "C:\inetpub\wwwroot\"&move c:\tmp\rar.rar
D:\MyWebsite\ISO\CSS\&echo [S]&cd&echo [E]

Alerted
maliciousness

2. c:\tmp\rar.exe a c:\tmp\rar.rar c:\tmp\result.dat

rar.rar implied
maliciousness

3. C:\tmp\ADEXplorer.exe -snapshot "" C:\tmp\result.dat /accepteula

result.dat implied
maliciousness

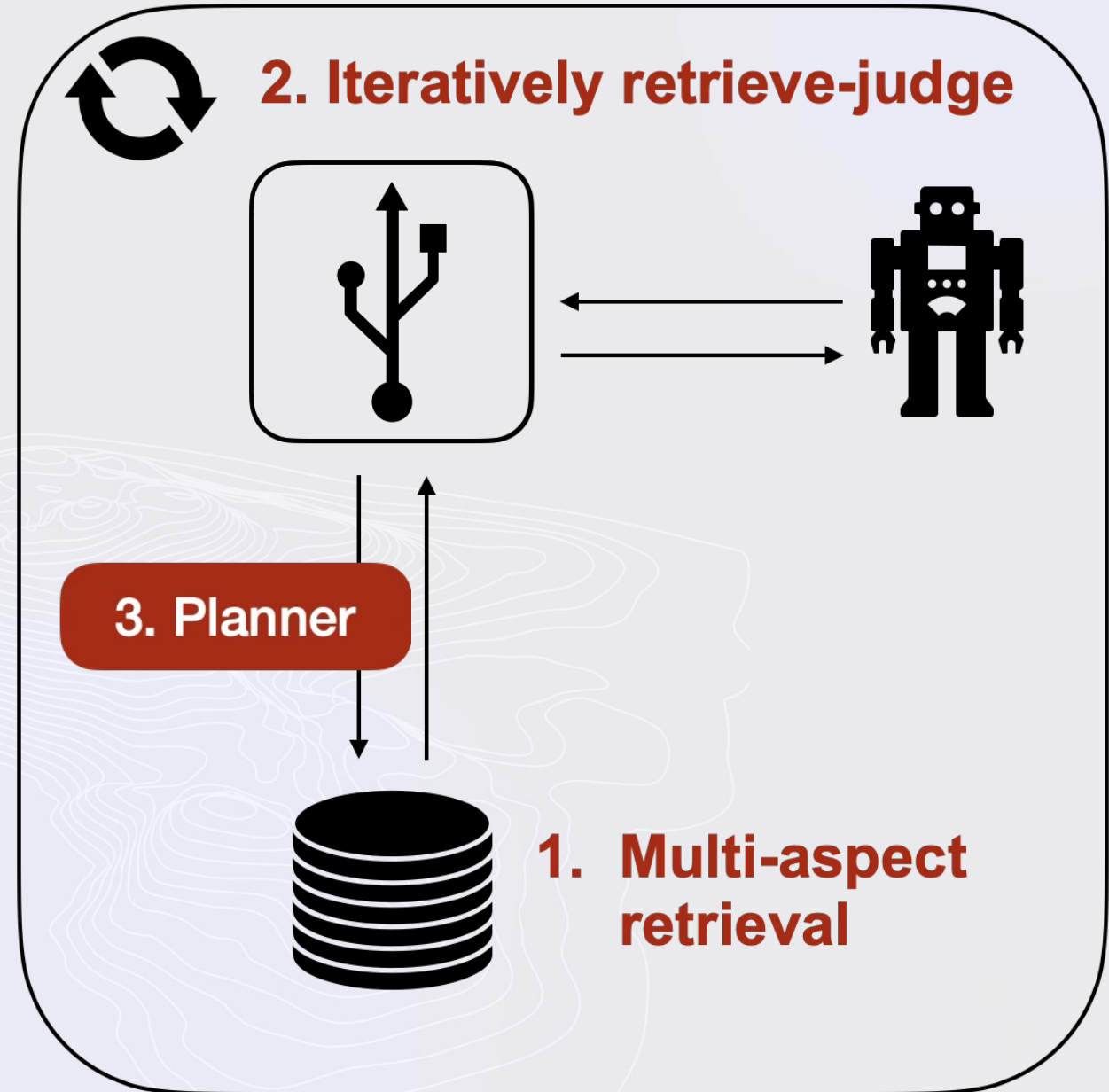
> 迭代過程中，事件調查的順序會影響效率甚至結果

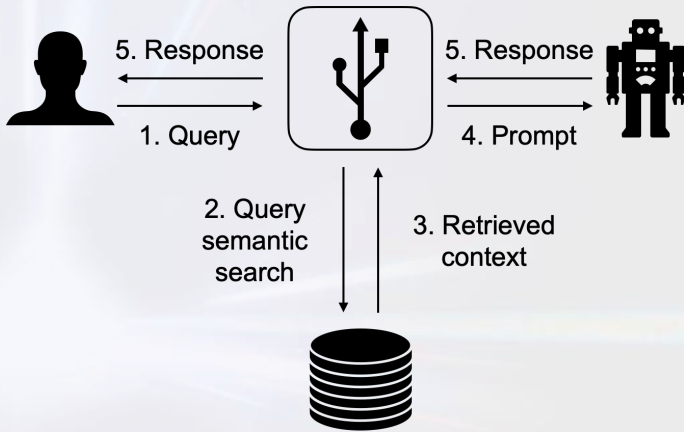
資安鑑識調查

建構 RAF 系統，需要...

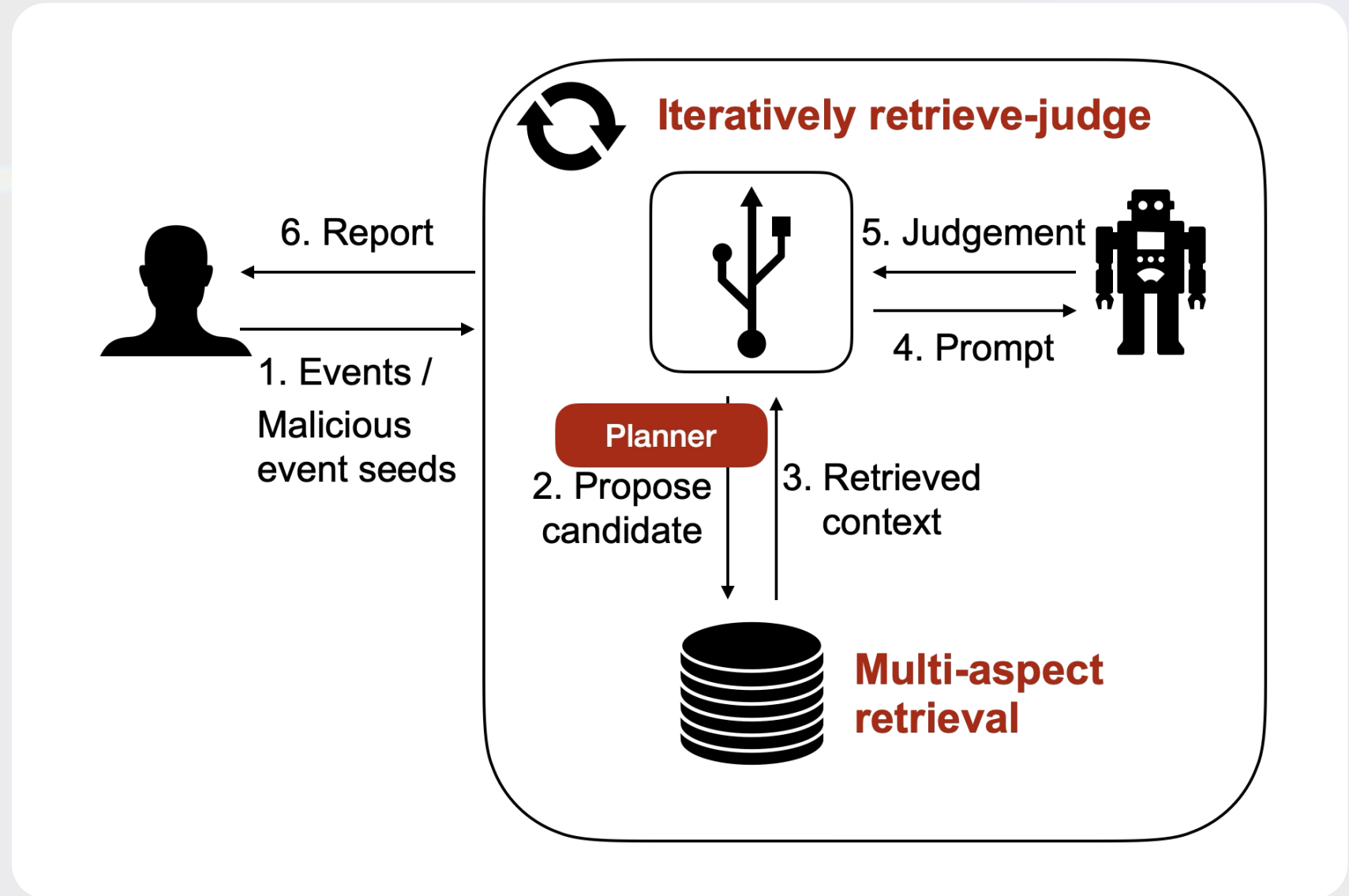
> Iterate

- > cand_events =
planner(current_state)
- > context =
multi_aspect_retriever(
cand_events
)
- > label =
Judge(cand_events, context)





RAF





資安鑑識調查

建構 RAF 系統，需要的資料結構是...

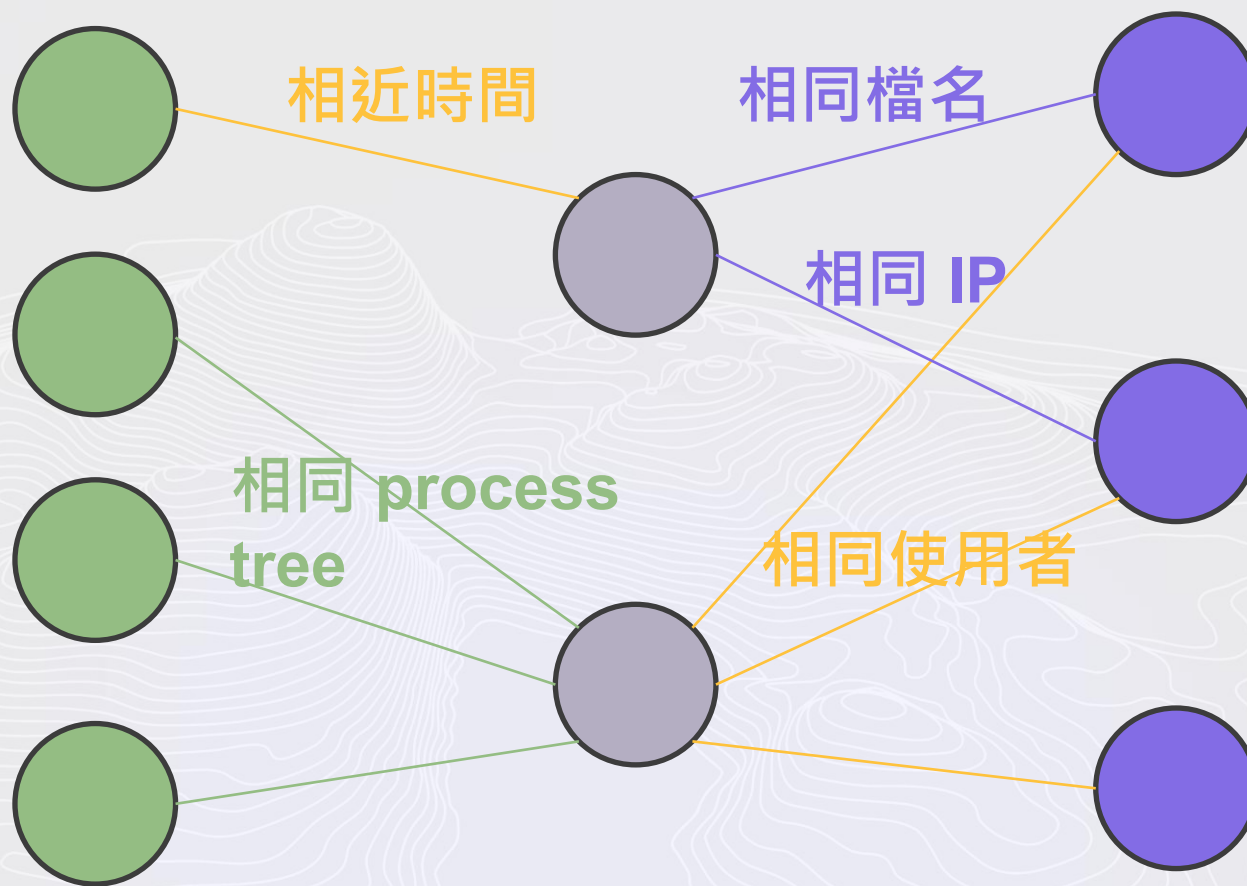
- > 能夠支援不同事件之間多面向關係檢索
- > 能夠提供迭代式檢索判斷過程中，(planner) 規劃事件調查順序所需資訊

圖！

正常事件

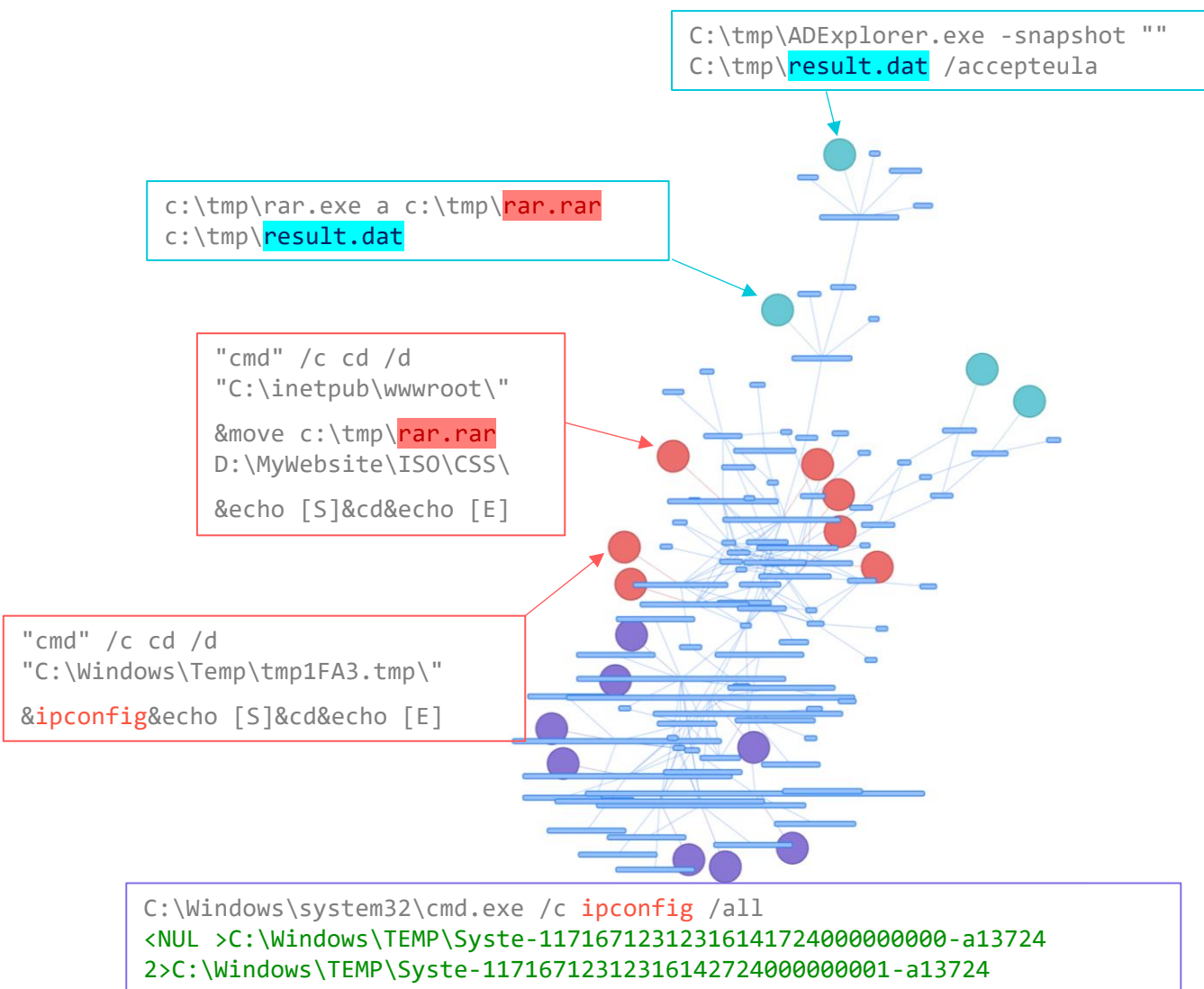
未定事件

惡意事件



CLinker : Command Line Linker

– RAF 圖的 建構系統





Methodology



建圖系統 – CLinker



完整建圖系統

輸入 (命令列)

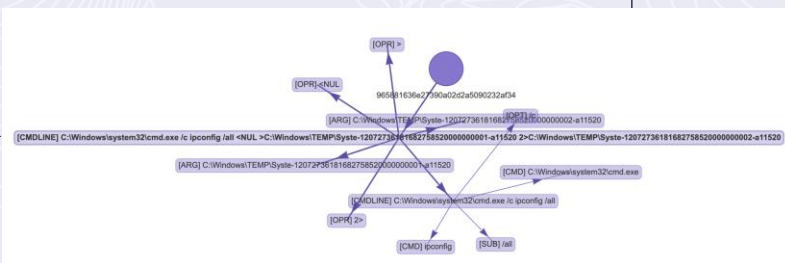
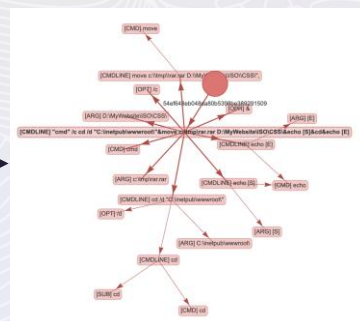
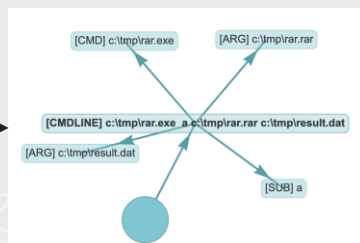
```
c:\tmp\rar.exe a
c:\tmp\rar.rar
c:\tmp\result.dat
```

```
"cmd" /c cd /d
"C:\inetpub\wwwroot\"
&move c:\tmp\rar.rar
D:\MyWebsite\ISO\CSS\
&echo [S]&cd&echo [E]
```

```
C:\Windows\system32\cmd.exe /c
ipconfig /all
<NUL >C:\Windows\TEMP\Syste-
1171671231231614172400000000-
a13724
2>C:\Windows\TEMP\Syste-
1171671231231614272400000001-
a13724
```

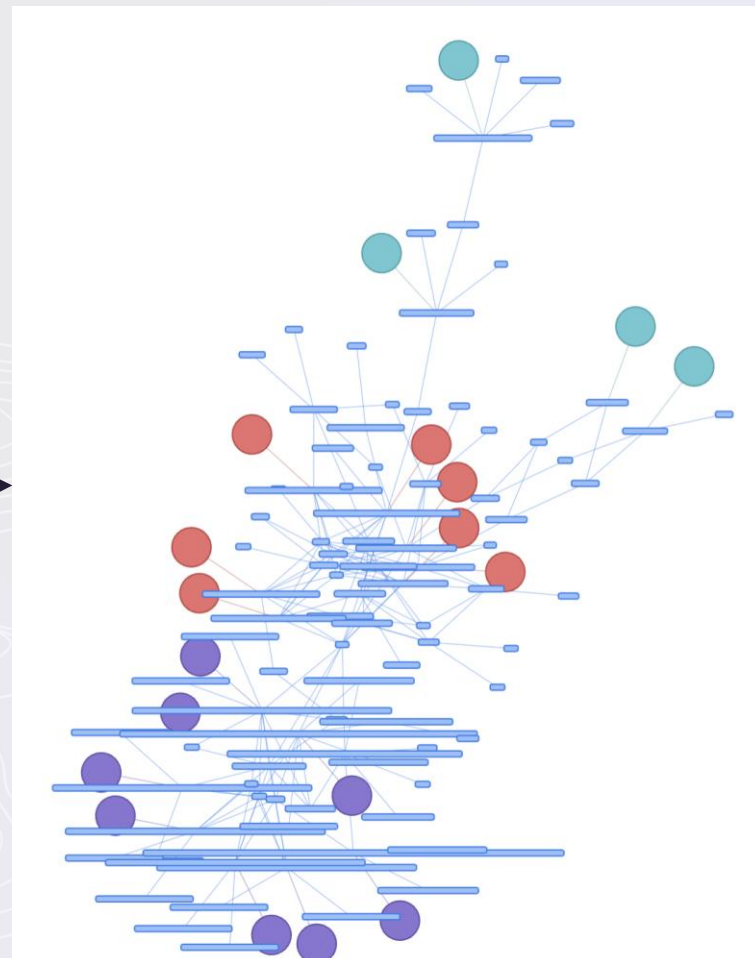


事件命令列圖

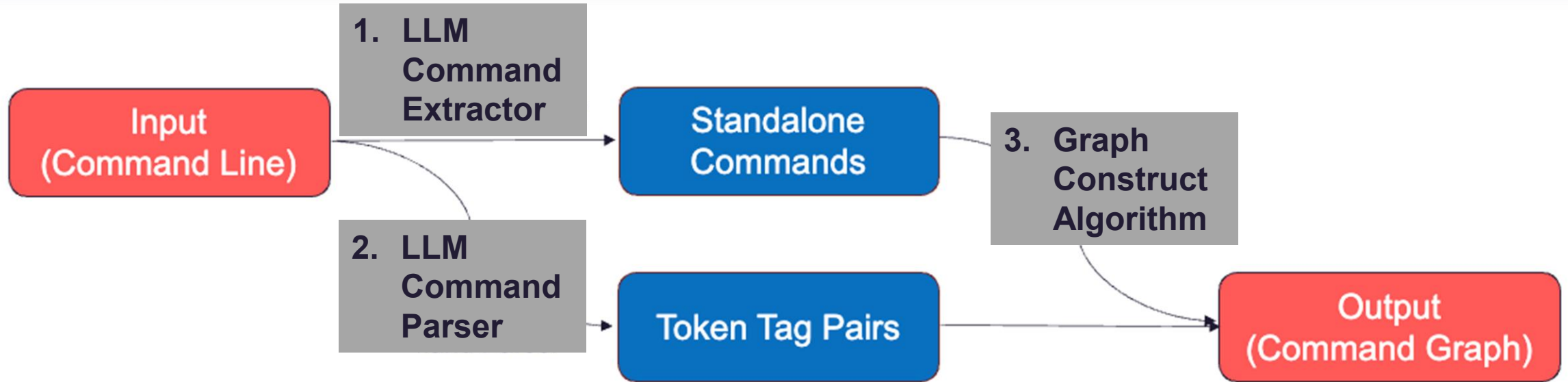


Merge graphs

事件圖



CLinker





1. Extractor – 擷取子命令列

Structure	Definition	Example
Encapsulated	command running another within its scope	<code>bash -c 'ls -l'</code>
Command Substitution	command executed in a subshell	<code>current_date=\$(date)</code>
Pipelined	commands connected via pipes	<code>ls -l grep <keyword></code>
Chained	commands connected using logical operators	<code>mkdir new_folder && cd new_folder && touch</code>



2. Parser – 擷取多面向關係 (Tag – Token)

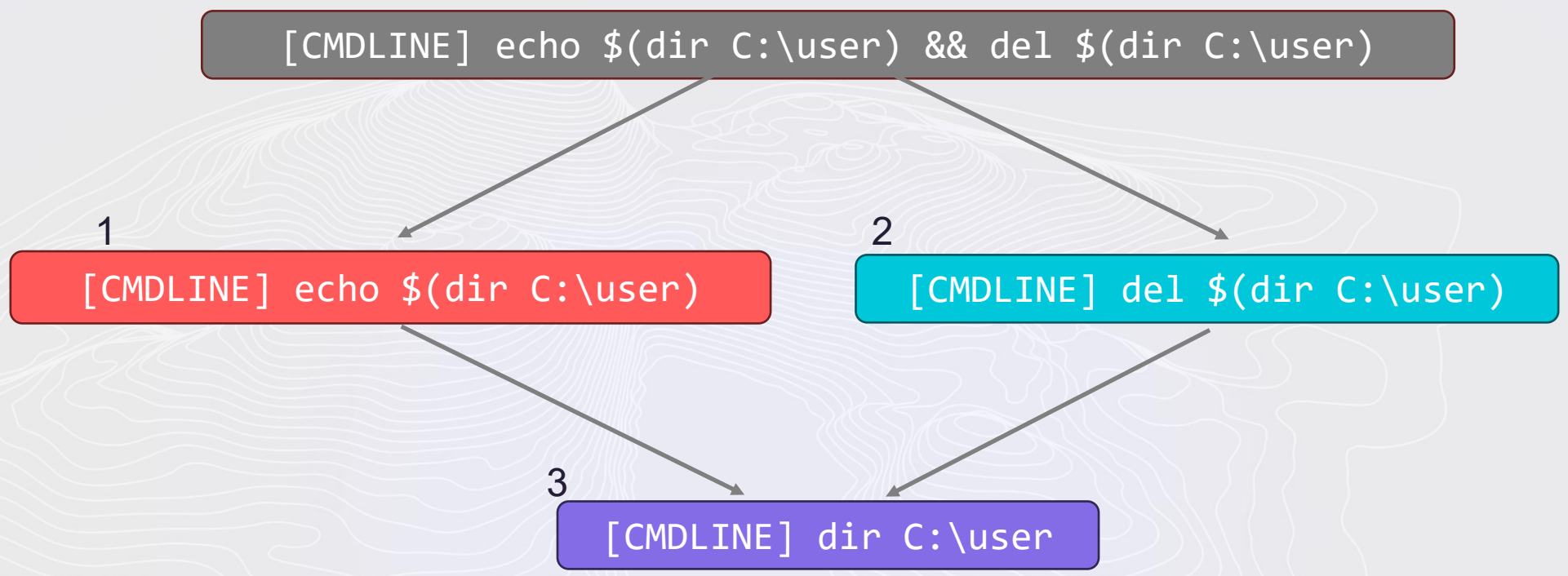
Tag	Definition	Example
Command (CMD)	Main invoked tool	"echo \$(dir C:\Users)"
Subcommand (SUB)	Context modifier, specifying the behavior of CMD	"git log --oneline"
Option (OPT)	Flag/switch, modifying behavior of CMD/SUB	"cmd.exe /e:on"
Argument (ARG)	Literal value for CMD/SUB/OPT	"rm -rf *.json"
Operator (OPR)	Flow control, redirection, or pipeline, ...	">" output redirection



3. 建立 Event Graphs

子命令列擷取

```
echo $(dir C:\user) && del $(dir C:\user)
```



3. 建立 Event Graphs 多面向關係擷取

```
echo $(dir C:\user) && del $(dir C:\user)
```

[

{OPR, &&},

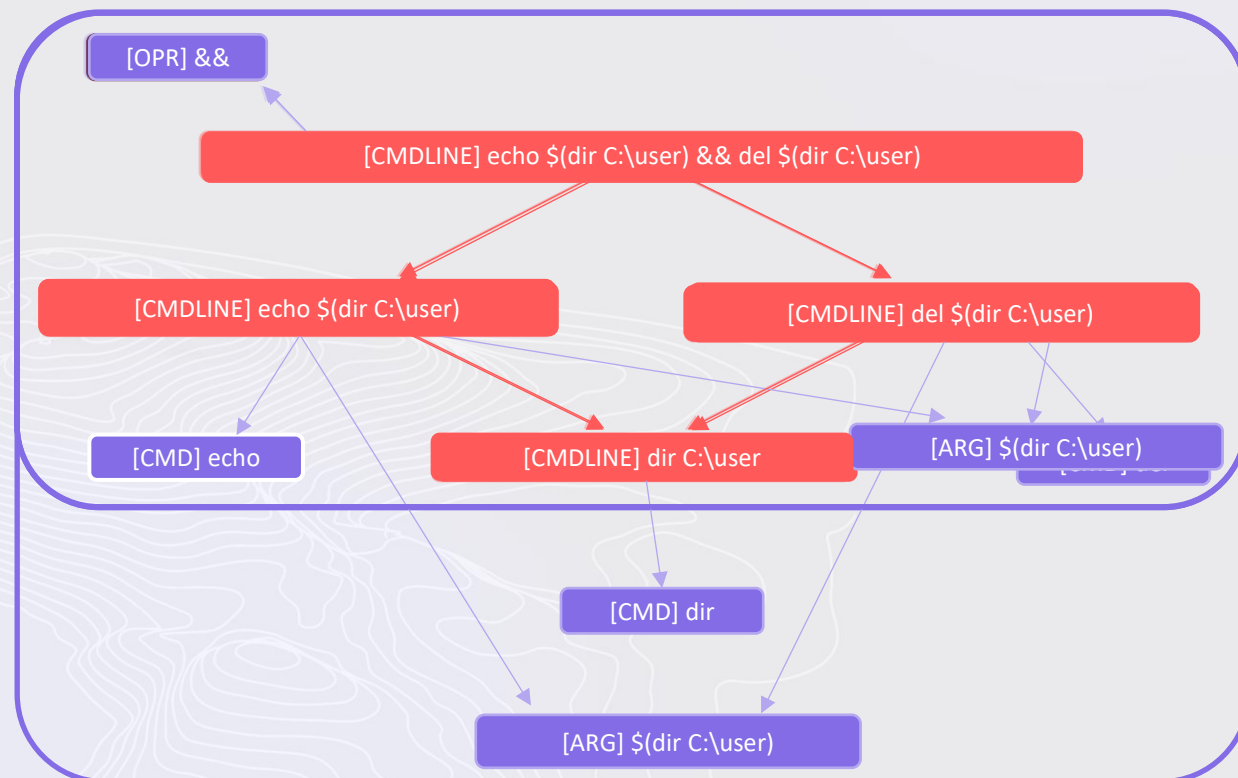
{CMD, echo},

{ARG, \$(dir C:\user)},

{CMD, del},

{CMD, dir}

]



Why LLM?

- > 我們分析命令列的目的
 - > 不是為了分析命令列
 - > 是為了從命令列找出事件間可能的關聯
- > 語法分析 (可能) 的侷限
 - > 封裝的指令 – `bash -c 'ls -l'`
 - > 自訂的語法 – `python3 my_script.py command\=ls\ -l`
 - > 語意的辨別 – `git log / cat log / cd log`
 - > 命名實體辨識 (Named entity recognition) – IP, filename, URL, PWD, ...

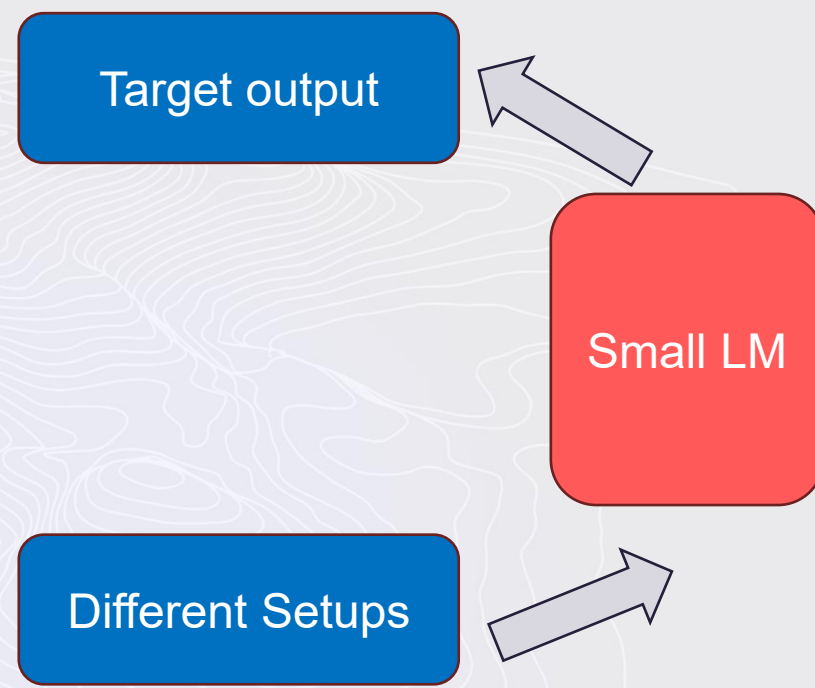
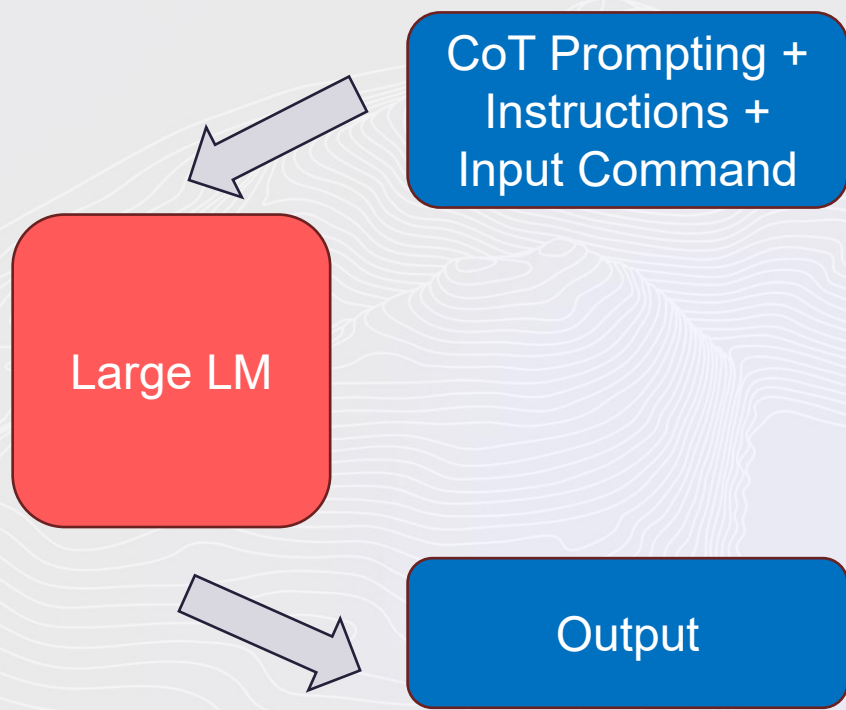


模型微調



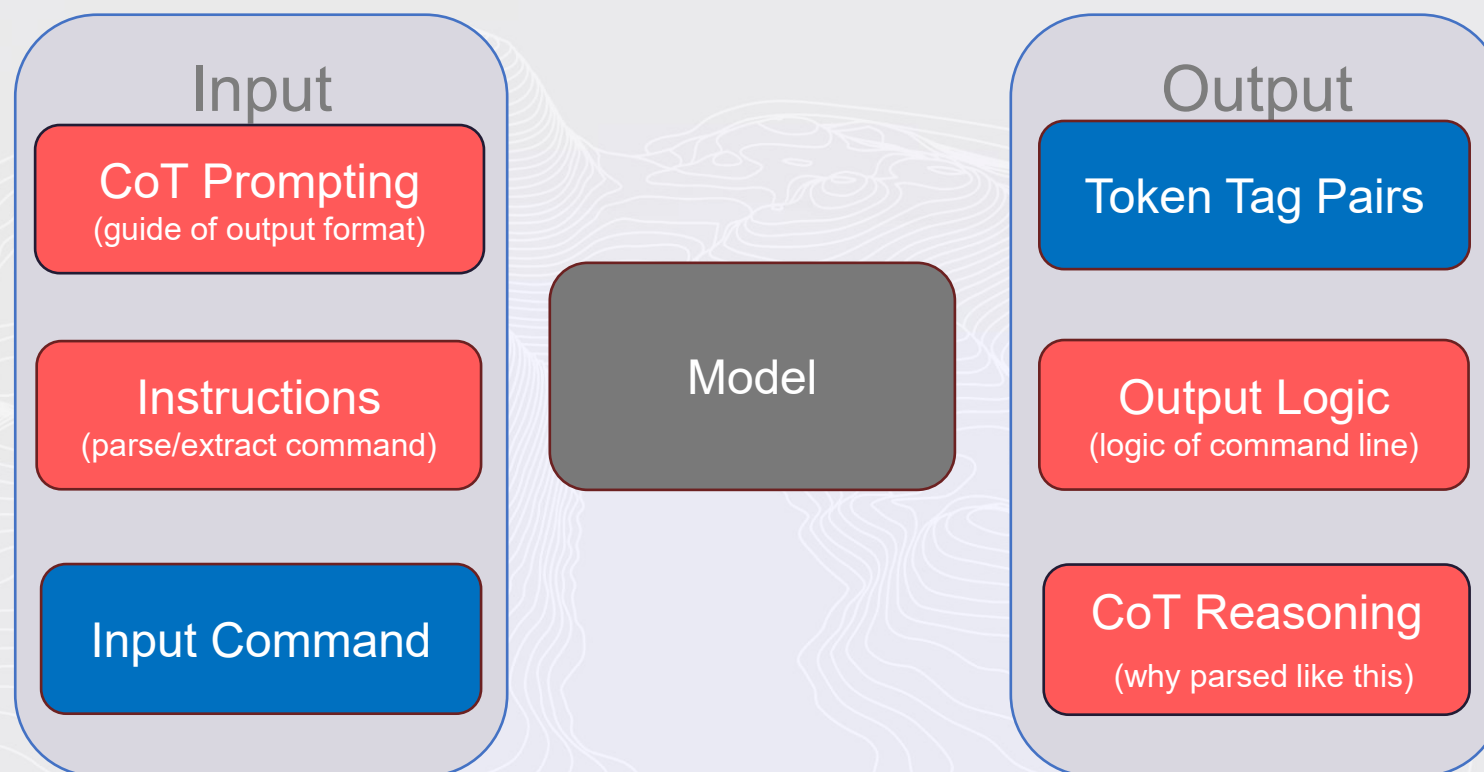
模型微調

> 目標 – 微調小模型，逼近大模型分析命令列的輸出



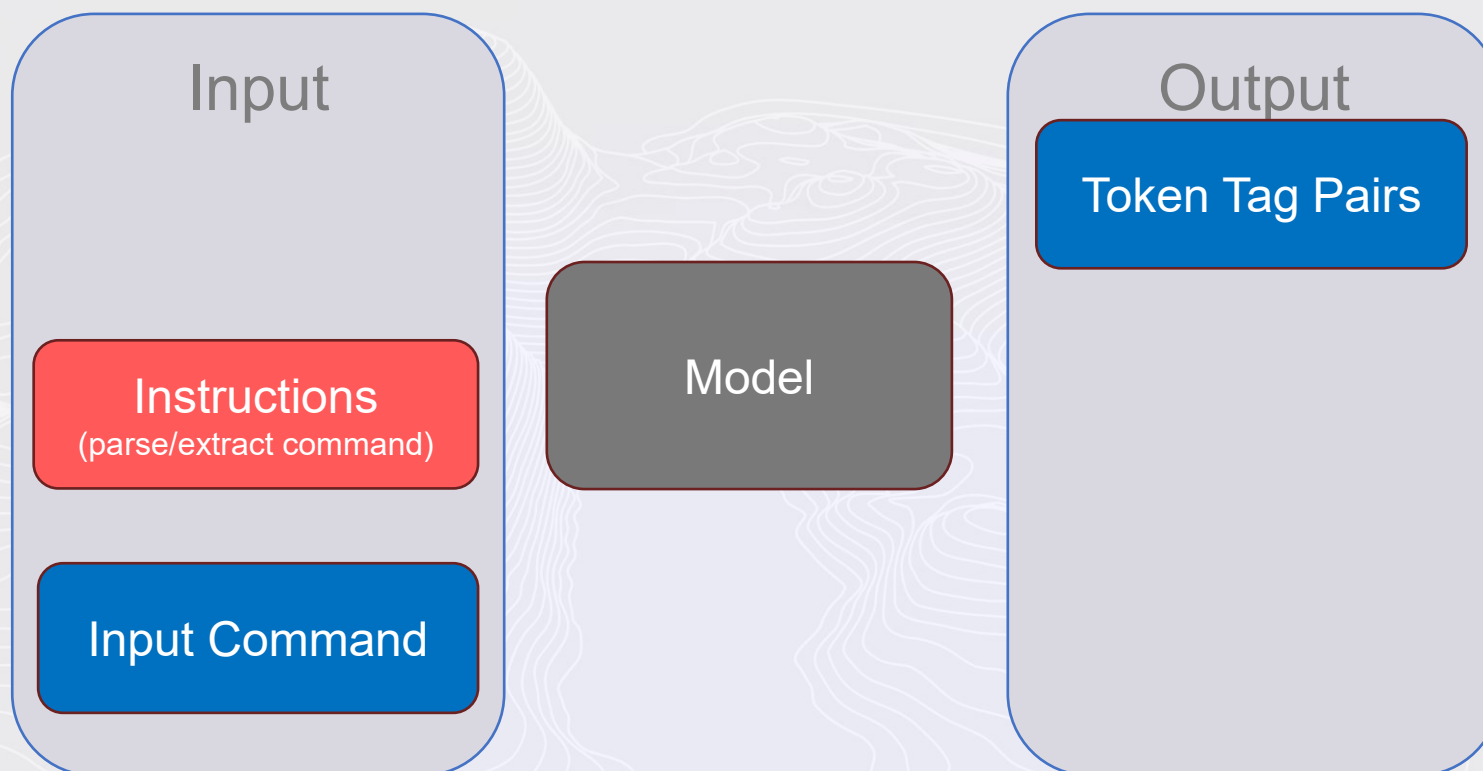
消融實驗

> 訓練小模型時，哪些是重要的呢？



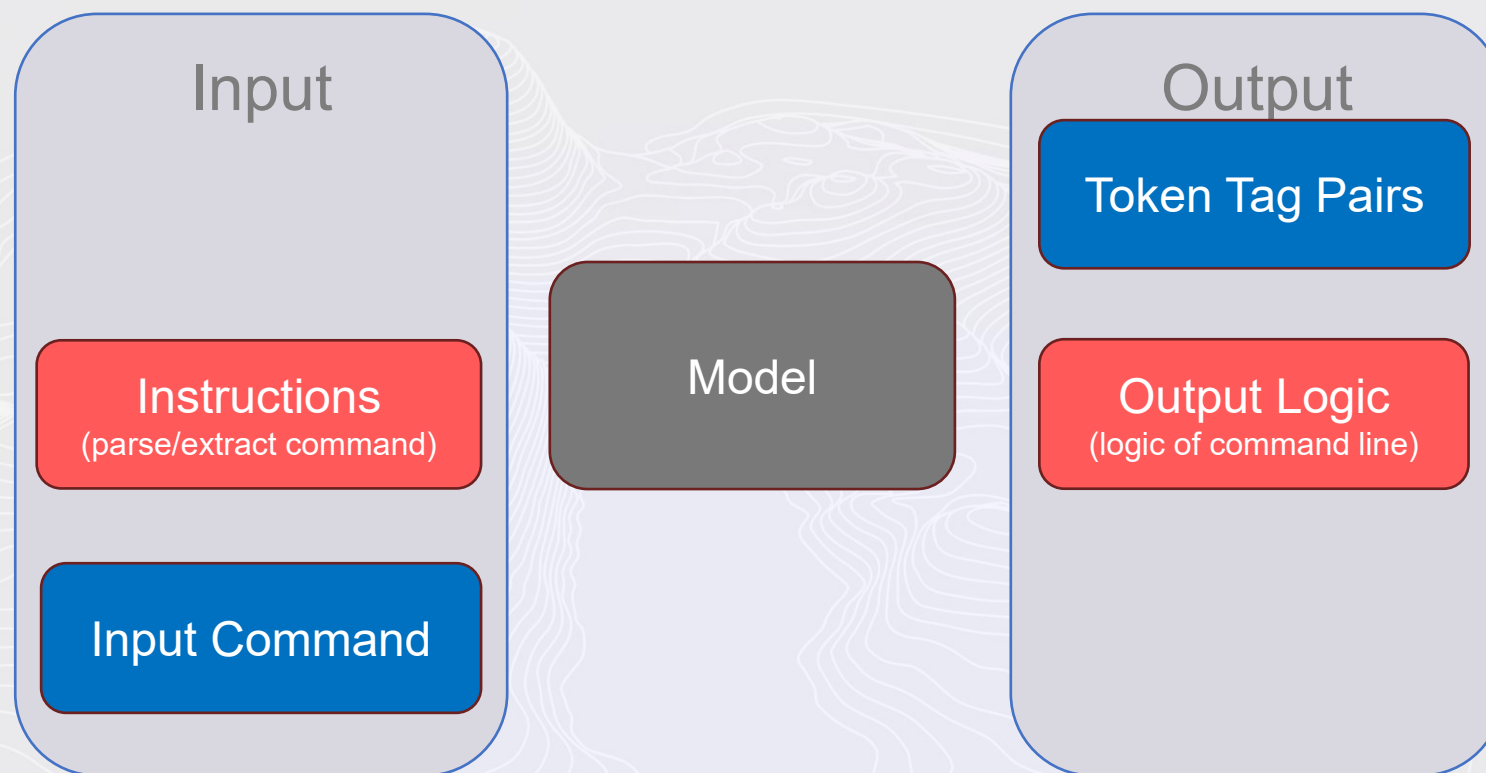
實驗設定 (1/4)

> SFT 1 (instructions + input → target output)



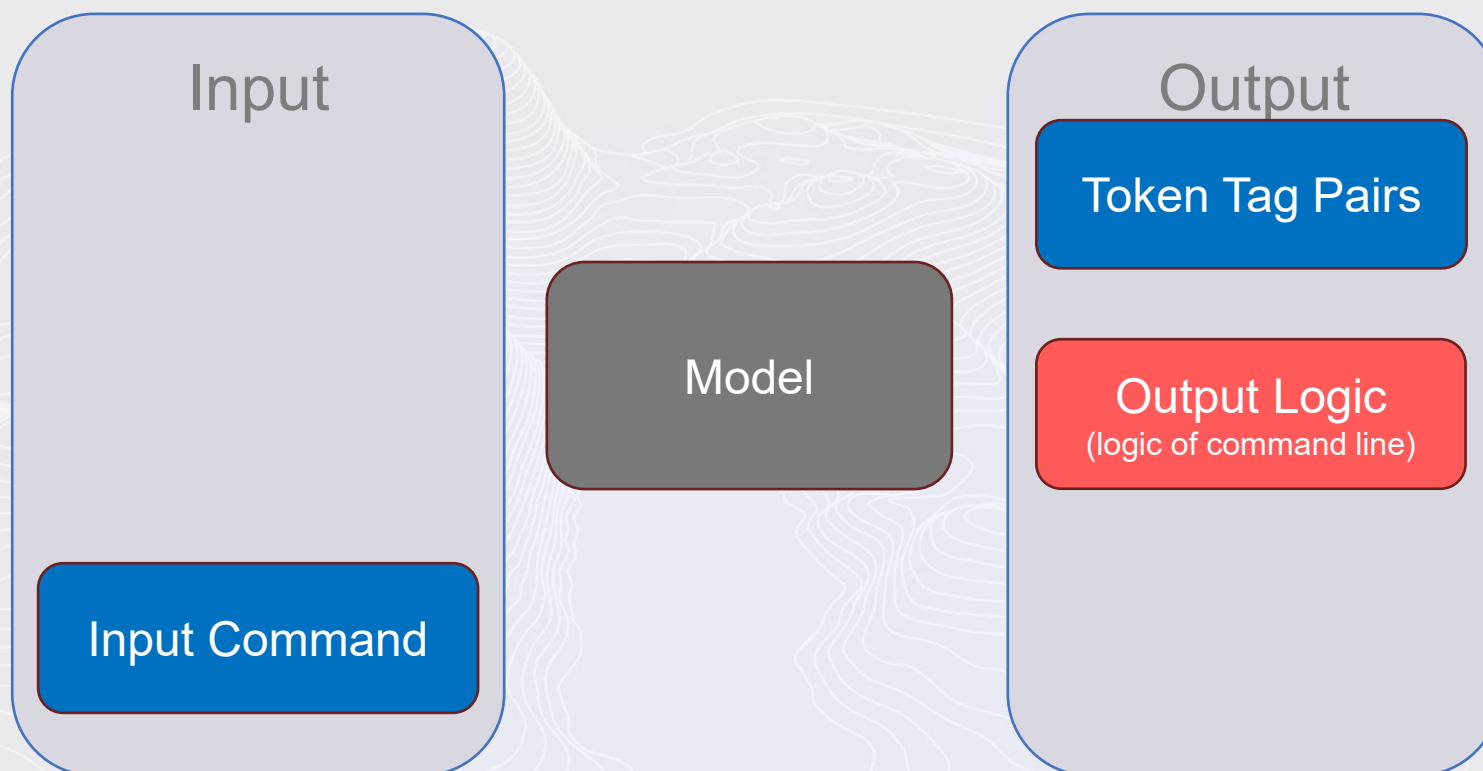
實驗設定 (2/4)

> SFT 2 (instructions + input → logic + target output)



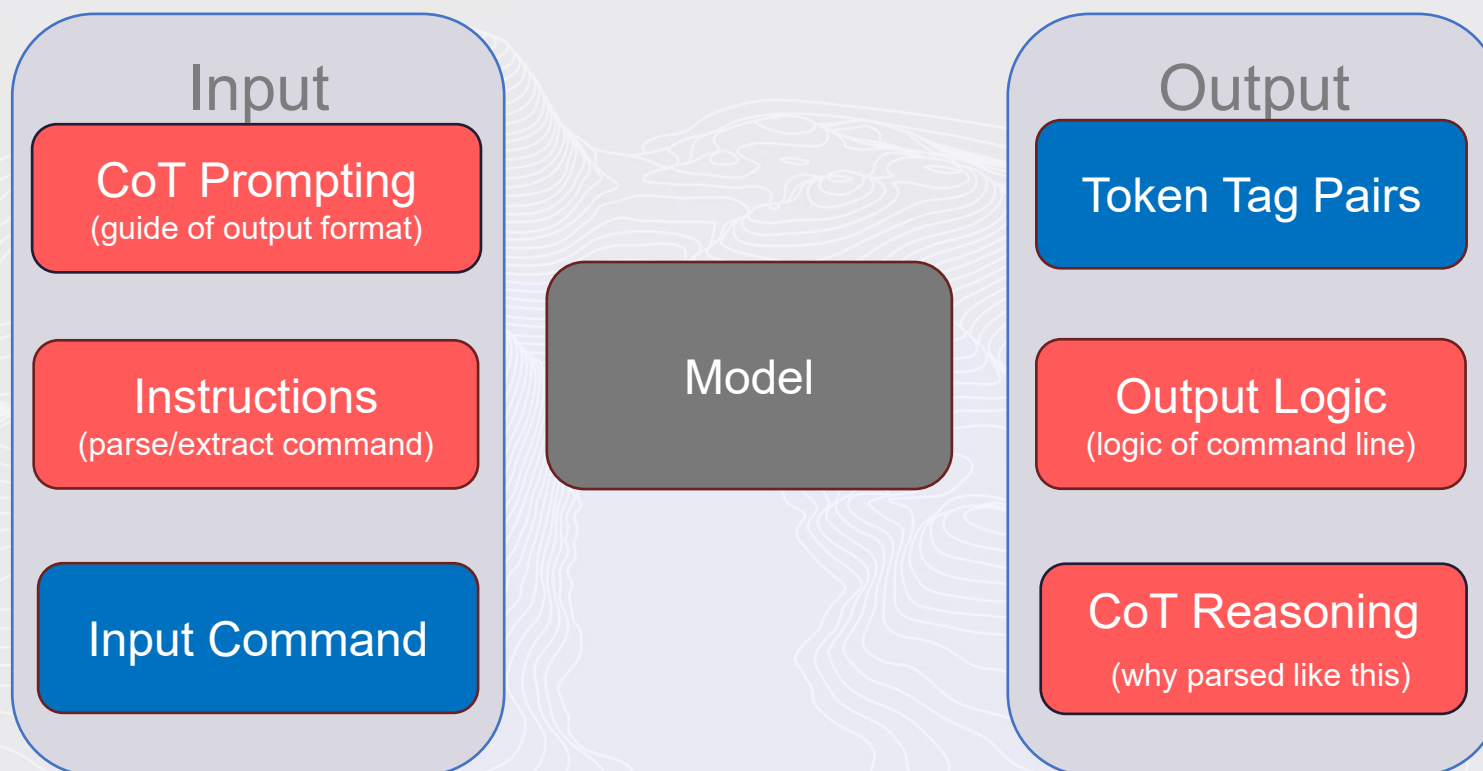
實驗設定 (3/4)

> SFT 3 (input → logic + target output)



實驗設定 (4/4)

> Knowledge Distillation



其他設定

> 大模型

- > LLaMA-3.3-70B-AWQ and Qwen2.5-72B-AWQ

> 小模型

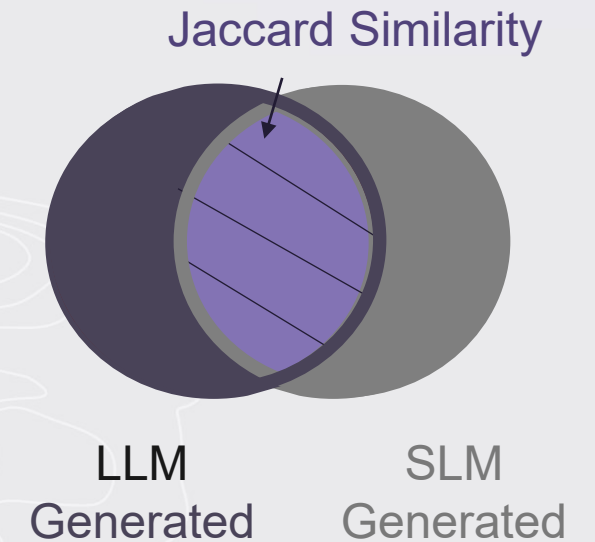
- > Qwen2.5-0.5B, Qwen2.5-1.5B, DeepSeek-R1-Distill-Qwen-1.5B, LLaMA-3.2-1B

> 資料

- > CyPHER

> 評估指標

- > Jaccard Similarity – 擷取出的子命令列 / tag-token 集合交集程度
- > Format Parse-ability – 輸出文字是否能正確解析成 JSON 物件
- > **Score** – Jaccard similarity 與 format parse-ability 的平均





Results



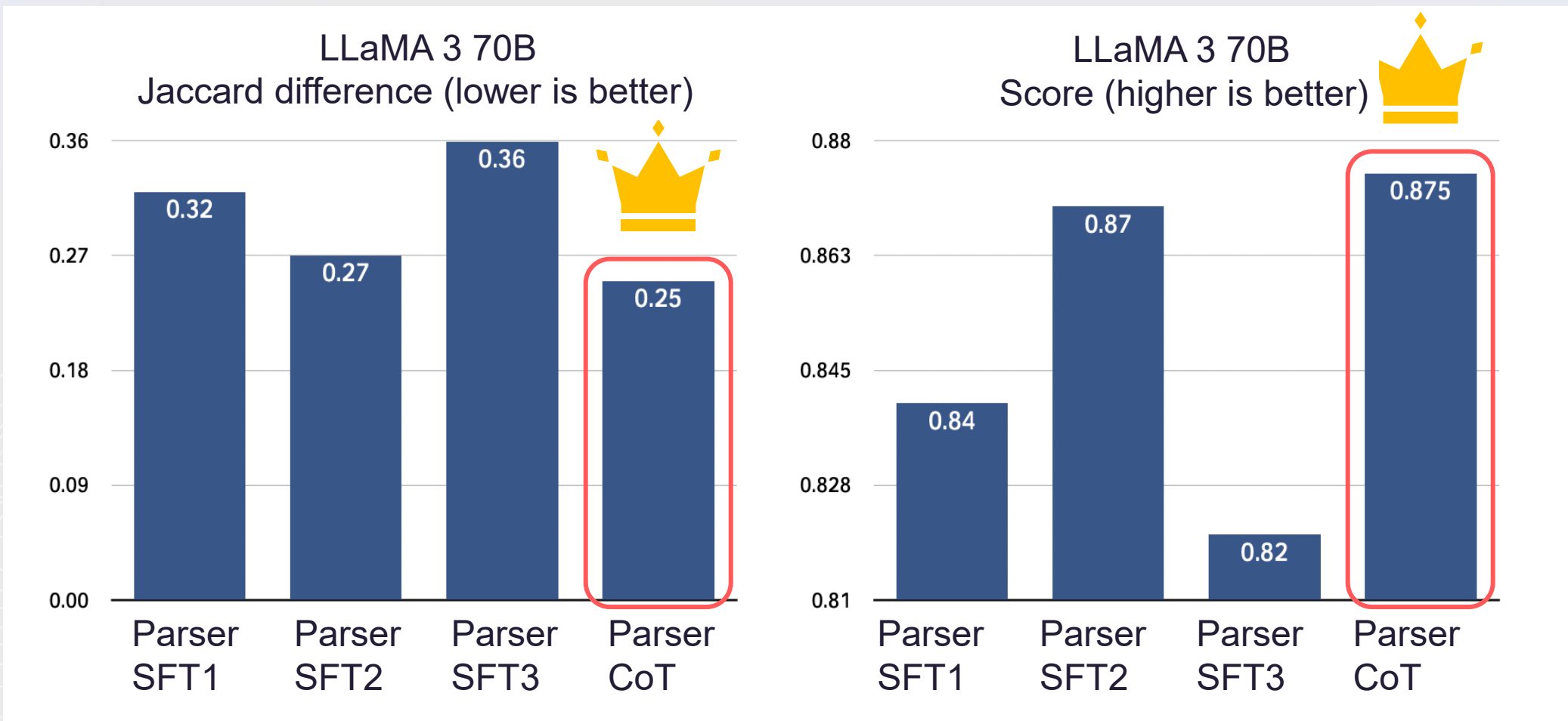
小模型訓練架構





最佳選擇 – 完整知識蒸餾

- SFT1: instruction + input → output
- SFT2: instruction + input → logic + output
- SFT3: input → logic + output
- CoT: instruction + input → CoT + logic + output
- Fine-tuned model: Qwen2.5-0.5B





大模型選擇

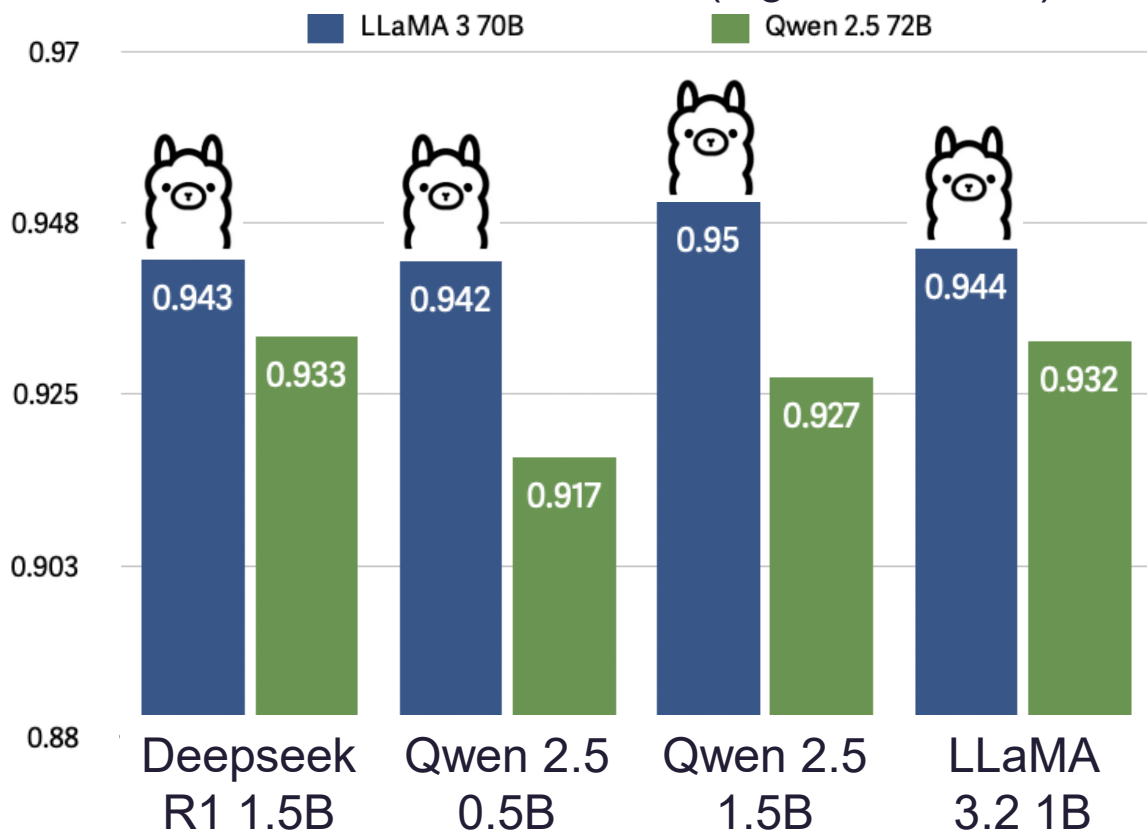




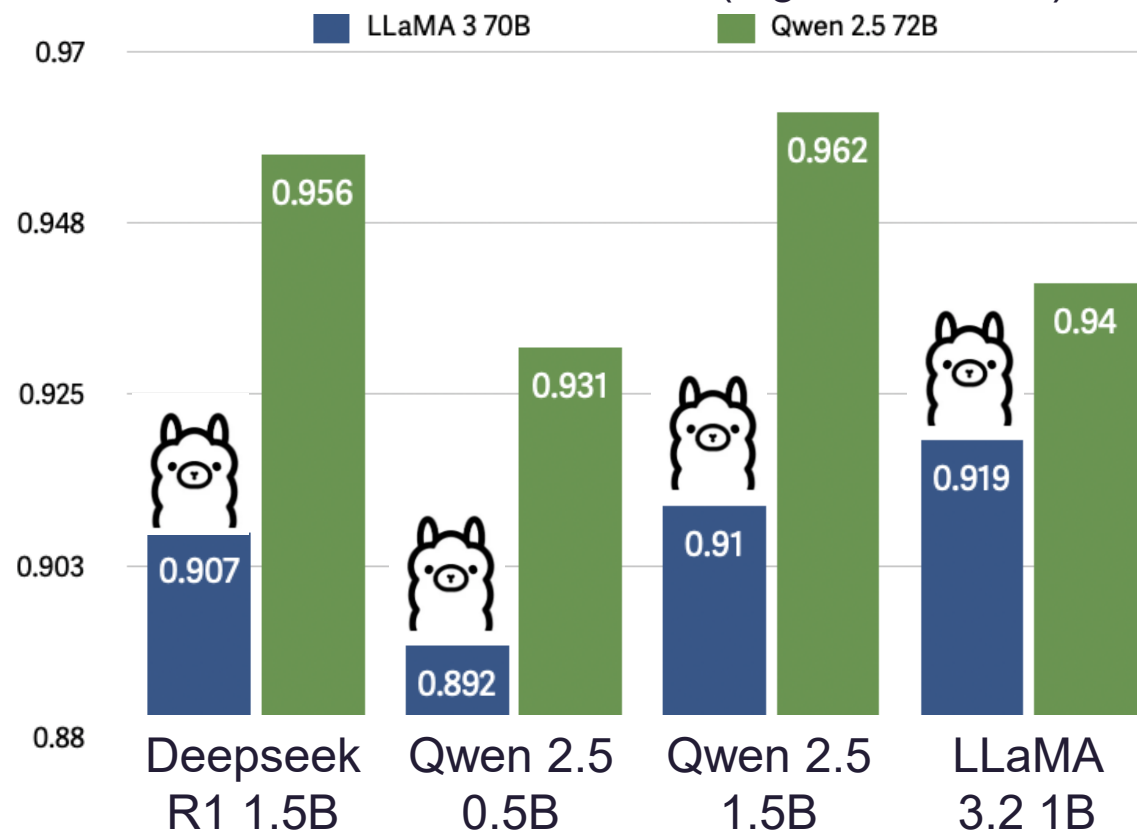
LLaMA vs Qwen

從 LLaMA 70B 學習，泛化能力更差

In-distribution Score (higher is better)



Out-distribution Score (higher is better)



LLaMA vs Qwen

LLaMA 生成更好的解釋

“wmic” nic where
“NetEnabled=true”
call renewdhcplease

The command line `"wmic" nic where "NetEnabled=true" call renewdhcplease` can be broken down as follows:

- `"wmic"` is the main command (CMD), which stands for Windows Management Instrumentation Command-line.
- `"nic"` is a subcommand (SUB) that specifies the management area, in this case, network interface controllers.
- `"where"` is a subcommand (SUB) that filters the results based on a condition.
- `"NetEnabled=true"` is an argument (ARG) for the `"where"` subcommand, specifying that only network interfaces that are enabled should be considered.
- `"call"` is a subcommand (SUB) that invokes a method on the selected objects.
- `"renewdhcplease"` is not a standard method for WMIC and seems to be a typo or incorrect command. The correct method to renew the DHCP lease is `"renewdhcp"`. However, assuming `"renewdhcplease"` is a placeholder for an actual method, it would be considered as part of the command (CMD) or subcommand (SUB) depending on its actual definition.

Given the context, it seems there was an attempt to correct or modify the command in the description. The standard approach would involve using `"renewdhcp"` for renewing DHCP leases on network interfaces that are enabled.

LLaMA vs Qwen

Qwen 生成更差的解釋

“wmic” nic where
“NetEnabled=true”
call renewdhcp lease

The command line `wmic nic where \"NetEnabled=true\" call releasedhcp lease, renewdhcp lease` can be broken down as follows:

- `wmic` is the main command (CMD) that is used to interact with Windows Management Instrumentation (WMI).
- `nic` is the subcommand (SUB) that specifies the context of the WMI query, in this case, network interfaces.
- `where \"NetEnabled=true\"` is a condition (ARG) that filters the network interfaces to only those that are enabled.
- `call` is an operator (OPR) that specifies that a method should be called on the filtered network interfaces.
- `releasedhcp lease, renewdhcp lease` are the methods (SUB) to be called on the filtered network interfaces. These methods release and renew the DHCP lease, respectively.

The logic of the command line is to use the `wmic` command to call the `releasedhcp lease` and `renewdhcp lease` methods on all network interfaces that are enabled, effectively releasing and renewing the DHCP lease for those interfaces.

LLaMA vs Qwen

LLaMA 更容易生成不一致的輸出

> 輸出錯誤

> Cmdline:

```
"xcopy /E /Y /H /K /V F:\\Backup\\ C:\\Backup_B\u66ff\\ /D /R /I"
```

> Output:

```
"C:\\\\Backup_B\\\\\\\\\u6559\\\\\\\\"
```

> 輸入錯誤

> Cmdline:

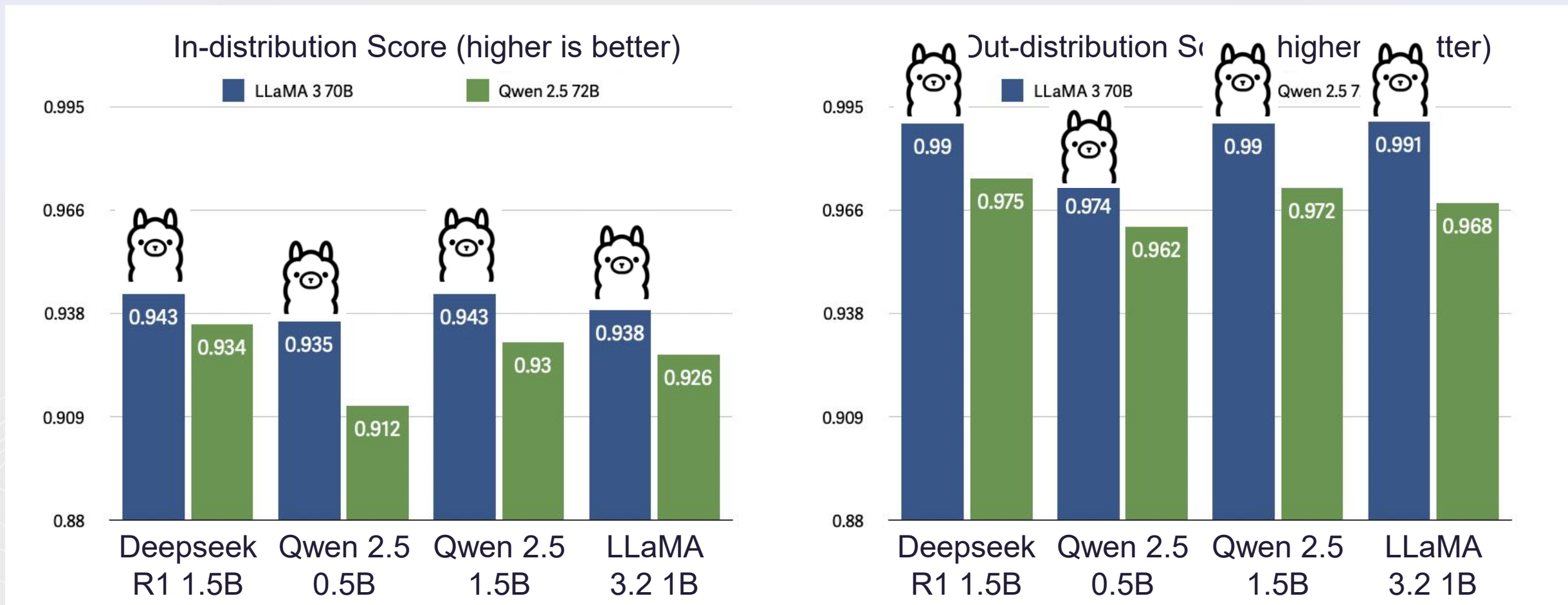
```
ipconfig -allquery
```

> Reasoning:

The command line **"ipconfig -all"** is used to display all the current TCP/IP network configuration values. ...



修正輸出 + 混合訓練 = 泛化能力提升



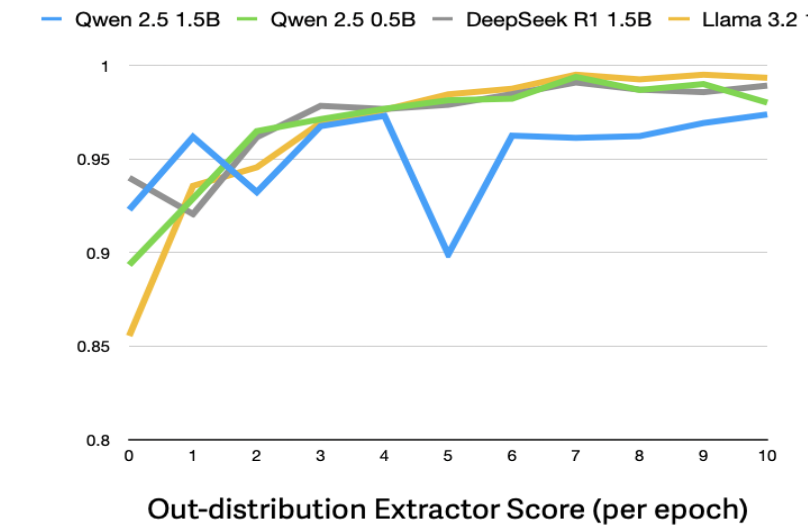
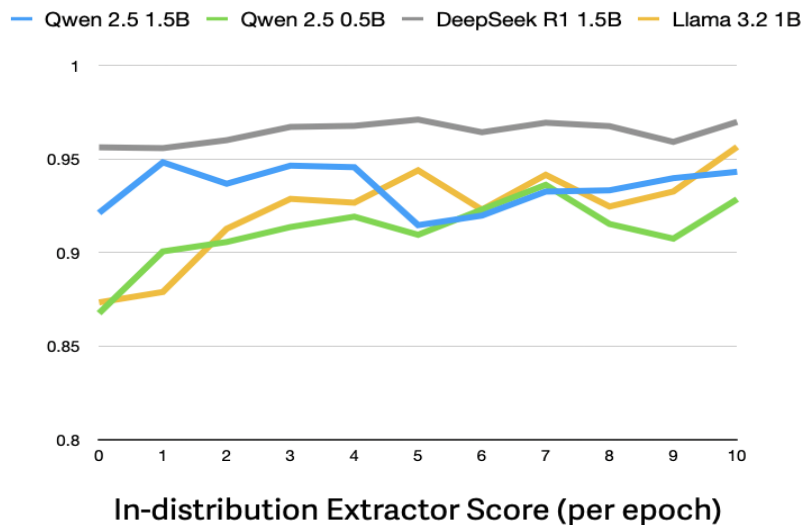
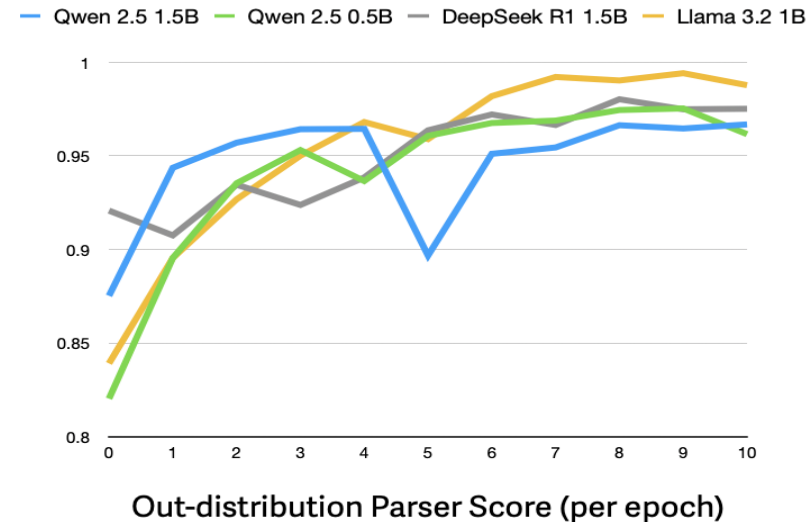
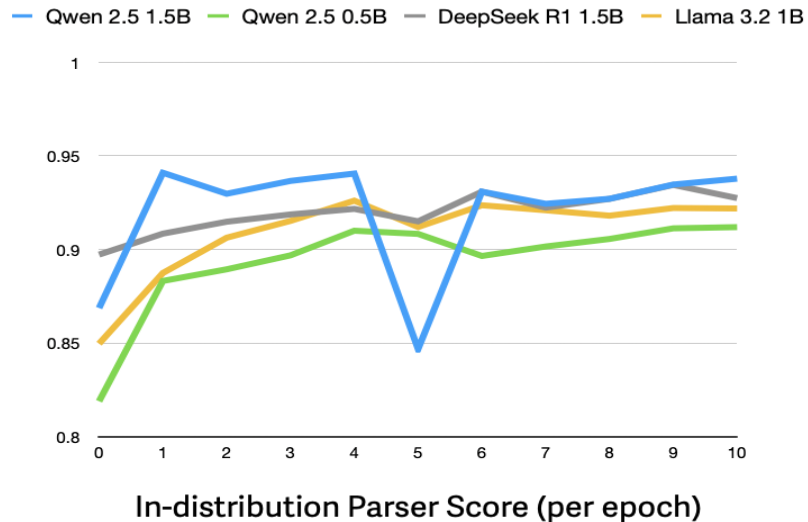


小模型選擇



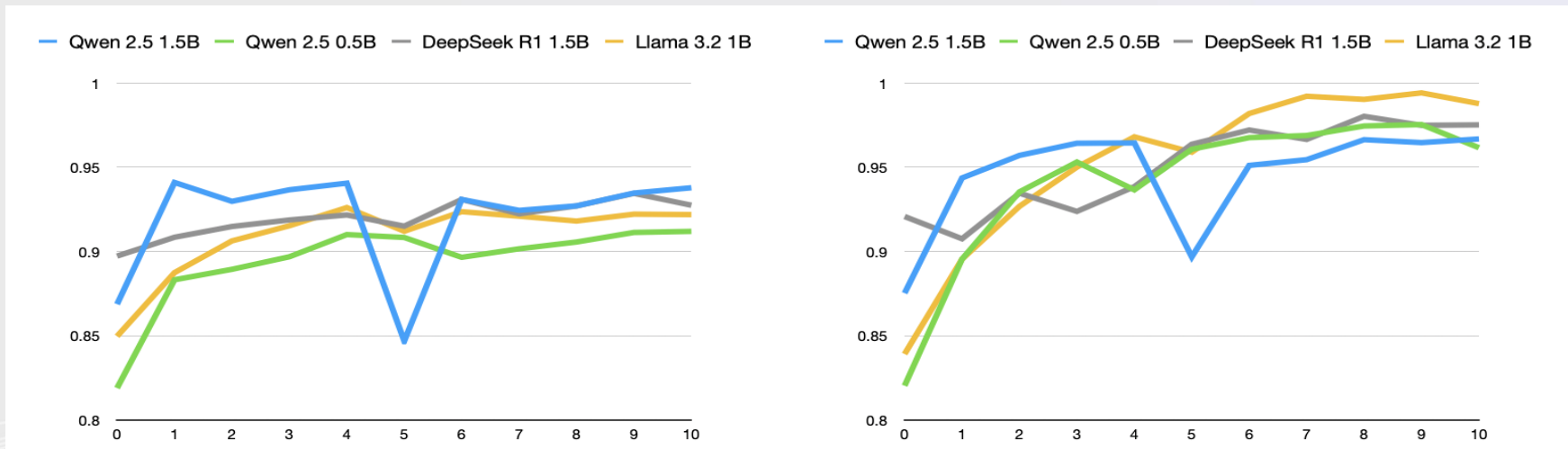


小模型比較

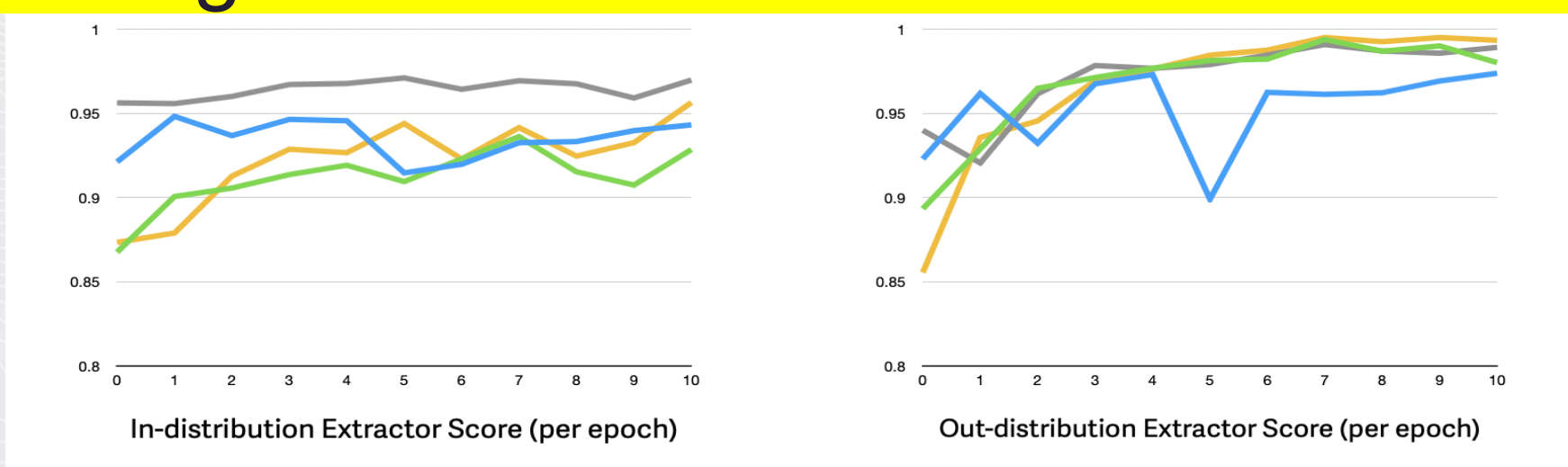




小模型比較



NO significant difference between models!





真實案例研究





真實案例

**CONFERENCE
ONLY**



總結



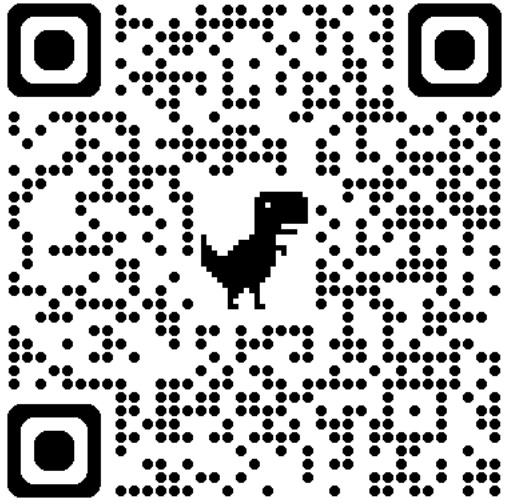


微調模型時的推薦組合

- > 訓練方式選擇
 - > 混合訓練 – 同時學習 擷取子命令列 與 擷取多面向關係
 - > 知識蒸餾 – 完整學習大模型輸入-輸出
- > 大模型（老師）選擇
 - > **LLaMA-3.3-70B** > Qwen-2.5-72B
- > 小模型（學生）選擇
 - > 評估指標上無顯著差異
 - > 實際資料上: **1.5B** > 1B > 0.5B



Takeaways



Takeaways

- > 鑑識調查 = 多面向檢索 + 迭代式檢索與判斷過程的 (graph) RAG
- > CLinker – 擷取子命令 + 擷取 tag-token + 圖論演算法 = 鑑識調查 (多面向檢索 + 迭代過程) 所需的事件圖
- > 蒸餾模型在命令列分析上可行
 - > 評分標準 – 測試集 score 達大模型 0.99
 - > 實際案例 – 事件間連結 ; token 字元錯誤
- > 下載模型 – <https://huggingface.co/CyCraftAI/CLinker>

