

# 璀璨的星空 —— 美麗的语言世界

The Brilliant Starry Sky —— The Beautiful World of Language



**Reed Chang**

**ASRD, National Chung-Shan Institute of Science and Technology**

**2026. 7.2**



## Amazing | 驚嘆

- 為何 ChatGPT 一次可以讀懂並記住 **128,000** 個字 ( context window ) ？
- 為何 ChatGPT 能精準翻譯多國語言？



## Questioning | 質疑

- 為何 ChatGPT 會有幻覺 ( hallucination : 一本正經地胡說八道 ) ？
- 為何愈「魯莽」的 Prompt，反而比「有禮貌」的問法回答得更精準？ ( 《經理人 Manager Today》曾報導：用較直接甚至較強烈的語氣提問，回答品質可能提升。是真的如此嗎？ )
- 為何同一個問題，ChatGPT 每次給的答案卻都不同？

## Questioning | 質疑

- ChatGPT 真的會「學習」嗎？
  - 已發布的 ChatGPT 會因為與使用者互動而繼續學習嗎？
- ChatGPT 真的能「預測」嗎？
  - ChatGPT 的 prediction 是預測未來，還是預測下一個 token？

## Wondering | 好奇

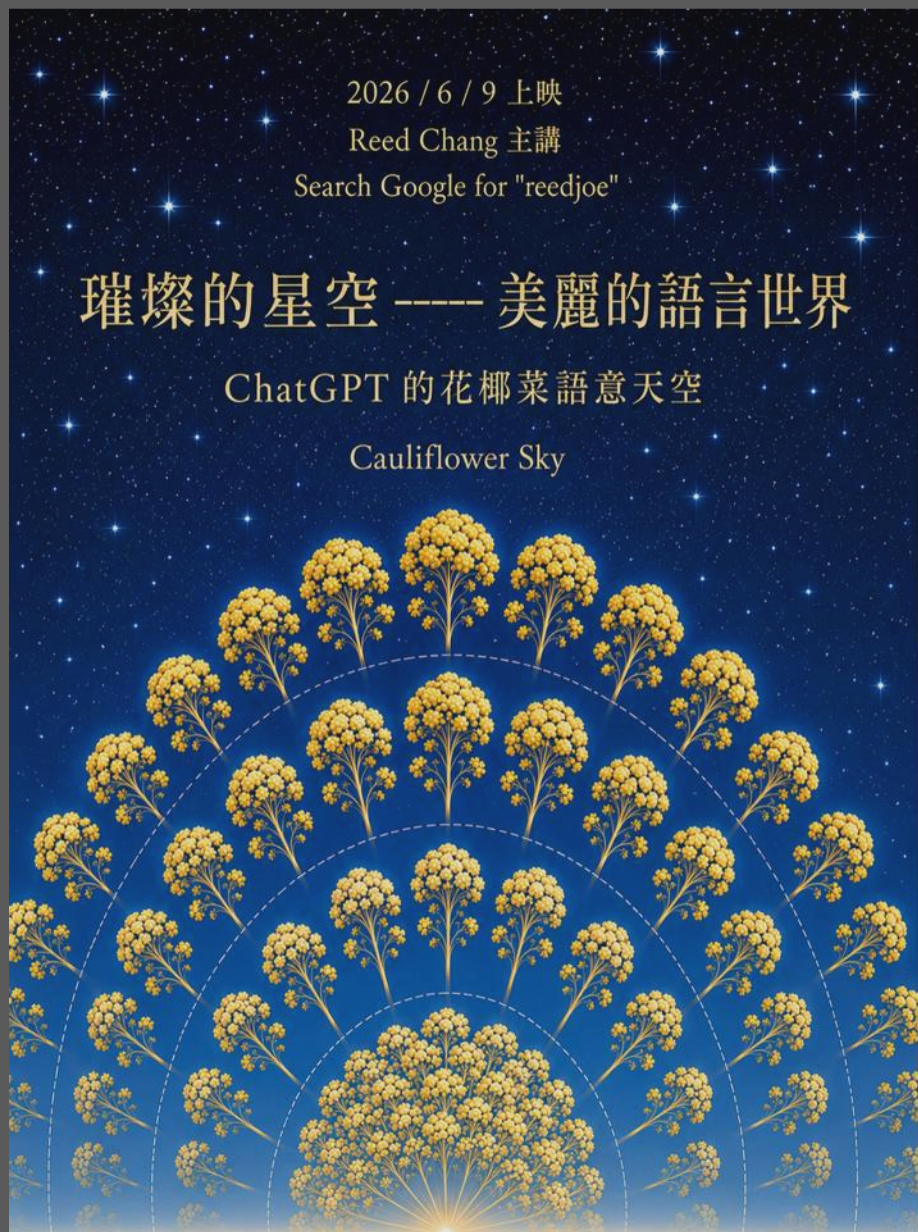
- ChatGPT 究竟是如何「理解」人類語言的？
- 人類智慧既然能讓 ChatGPT 學會語言，為何至今仍很難有效讓人類學好英語？
- 我們都會問 ChatGPT，但答案卻不同，到底誰才是對的？



## Understanding | 理解

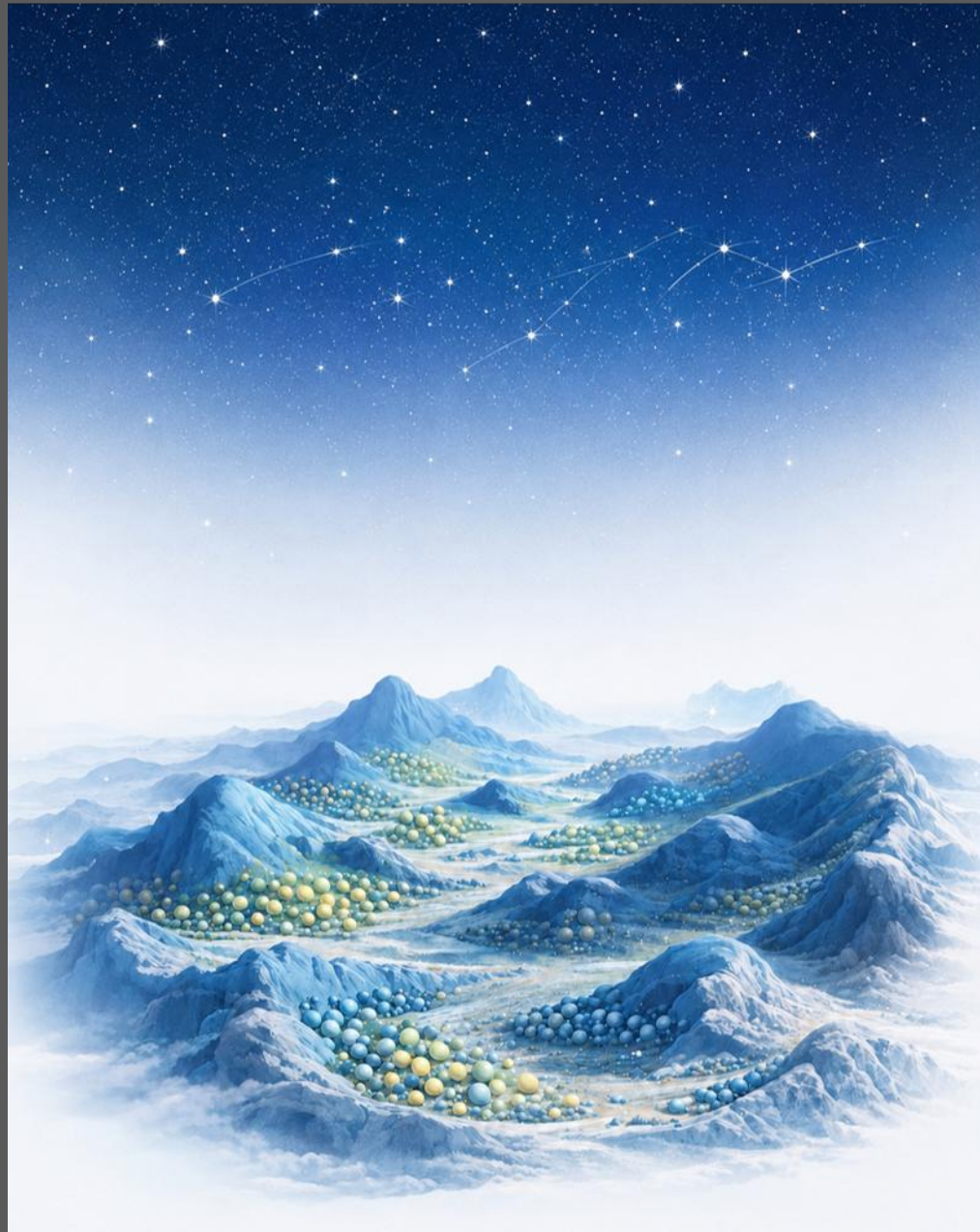
- 以上所有看似毫無關聯、甚至彼此矛盾的問題，其實都指向同一個核心答案。





## 人類智慧（HI）如何教 ChatGPT 學會語言？

- ChatGPT 經預訓練（**pretraining**）後，將全球語言中相近的語意，依其語意方向（**semantic direction**）聚合成不同的語意叢集（**clusters**）
- 在嵌入空間（**embedding space**）中，形成一片如花椰菜般層層展開的語意天空——**Cauliflower Sky**
- ChatGPT 藉此語意天空（**Cauliflower Sky**）理解、生成語言，而走進全世界人類的心。



人類智慧既然可以教會 ChatGPT 學會語言，為何至今仍很難有效教會人類學習英語？

(1) ChatGPT 如何學會語言

How Does ChatGPT Learn Language?

(2) ChatGPT 的花椰菜語意天空

ChatGPT' s Cauliflower Semantic Sky

(3) 像 ChatGPT 一樣學英語

Learning English Like ChatGPT

1

人工智慧 (AI)  $\longleftrightarrow$  人類智慧 (HI)

預訓練前準備 (Pretraining Preparation)

預訓練 (Pretraining) [Attention + FFN]  $\longleftrightarrow$  推論 (Inference)

生成式 AI (Generative AI)  $\longleftrightarrow$  代理式 AI (Agentic AI)

什麼是 AI

人類智慧 HI

人工智慧  $\xrightarrow{\text{模擬}}$  人類的智慧



## 1. 蒐集語料 (Data Collection)

ChatGPT 並不是直接閱讀整個網際網路 (Internet) 。

首先，需要從各種公開或授權來源蒐集大量文字資料 (text corpus)，例如：

- **Common Crawl** ( 大型網頁快照資料庫 )
- **Wikimedia Foundation Wikipedia** ( 維基百科 )
- 書籍 (Books)
- 新聞文章 (News Articles)
- 技術文件 (Technical Documents)
- 學術資料 (Academic Resources)
- 程式碼 (Source Code)
- 公開論壇與討論區 (Public Discussions)

目的是建立一個足夠廣泛的語言樣本庫 (language corpus) 。

## 2. 清理與整理資料 (Cleaning and Curation)

原始資料 (raw corpus)品質 參差不齊。  
因此必須進行：

- 資料清理 **Cleaning**
- 資料篩選 **Filtering**
- 去重複 **Deduplication**
- 人工與自動化整理 **Curation**

原始資料可能整理成約 13 兆 tokens，供後續選用。

### 3. 選擇約 3 兆 Tokens 進行預訓練

(1) 品質比數量更重要：並非所有資料都同樣有價值，保留：

- 高品質書籍
- Wikipedia
- 技術文件
- 高可信度網站

(2) 避免過度重複 (Over-representation)

(3) 計算成本 (Computational Cost)

(4) 最佳化資料配比 (Data Mixture Optimization)：不同資料來源需要適當比例。

\* 只保留最有訓練價值的部分，作為預訓練 (Pretraining) 的語料。

## 3 兆 Tokens 究竟有多大？

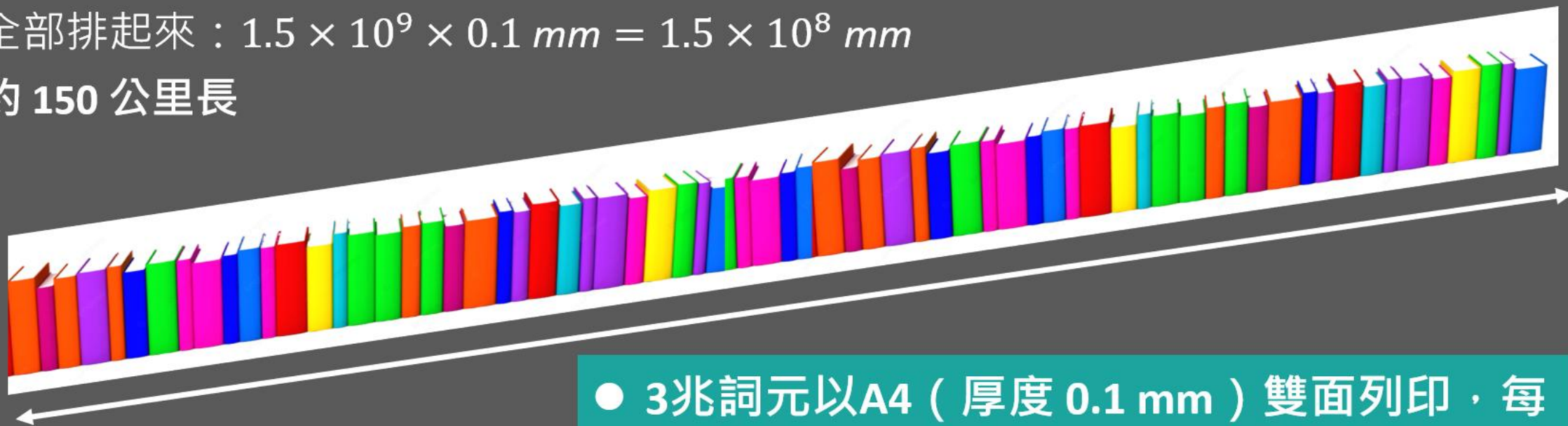
假設：

- 每頁 A4 紙可印 1,000 個 Tokens
- 採雙面列印 ( Double-sided Printing )
- 每張紙厚度約 0.1 mm

### 3 兆 Tokens

全部排起來： $1.5 \times 10^9 \times 0.1 \text{ mm} = 1.5 \times 10^8 \text{ mm}$

約 150 公里長



- 3兆詞元以A4（厚度 0.1 mm）雙面列印，每頁印1,000字，排起來長達 150 公里。

## Vocabulary Table

來自 Tokenizer Training，例如：

- BPE
- WordPiece
- SentencePiece

## Embedding Matrix

經 Pretraining 逐漸學出來的。

初始值為 random values

例如：春 [0.003, -0.001, ...]

## 字彙表

| Vocabulary Table    |       |
|---------------------|-------|
| Token-Index Mapping |       |
| Index               | Token |
| 0                   | the   |
| 1                   | dog   |
| 2                   | is    |
| ...                 | ...   |

Token ↔ Index

Symbol Mapping

## 嵌入矩陣

| Embedding Matrix |                    |
|------------------|--------------------|
| Embedding Layer  |                    |
| Index            | Vector             |
| 0                | [1.05, -0.72, ...] |
| 1                | [0.87, 1.35, ...]  |
| 2                | [0.23, -1.10, ...] |
| ...              | ...                |

Index → Vector

Semantic Representation



## 整理

- 經過清理（cleaning）、篩選（filtering）與整理（curation）後，形成約 13 兆 tokens 的語料。
- 透過 Tokenizer Training 建立 Vocabulary Table（詞彙表），其中約包含 100,000 個詞元（tokens），有些模型甚至可達 250,000 個以上。
- 然後利用此 Vocabulary Table 將語料轉換為 tokens。
- 最後在預訓練（pretraining）過程中，模型逐漸學會 Embedding Matrix（嵌入矩陣），使每個 token 擁有豐富的語意表示（semantic representation）——  
詞元表徵 (token representation) — [1.2, 0.5, 2.1, 0.5, -2.1...] 高達12,288 維
- Tokenizer Training 產生 Vocabulary Table；Pretraining 學會 Embedding Matrix。

# 1

## 預訓練 (Pretraining) [Attention + FFN]

1. Self-Supervised Learning ( 自監督學習 )

2. Teacher Forcing ( 老師強迫 )

3. Causal Masking ( 因果遮罩 )

題目：春天

春天到了，百花開，桃花紅，李花白。(答案：ground truth)

Step 1

Causal Masking ( 因果遮罩 )

春天到了，

預測下一個詞元：Next Token Prediction

春天到了，樹

## Step 2

### Teacher Forcing ( 老師強迫 )

用正確的答案比對，修正：

預測下一個詞元：**Next Token Prediction**

春天到了，樹



題目：春天

春天到了，百

花開，桃花紅，李花白。(答案：ground truth)

## Step 3

### Self-Supervised Learning ( 自監督學習 )：

經由 Teacher Forcing 提供答案，自我比對、修正。

# 預訓練：Attention + FFN

**Attention**：注意力

**Feed-Forward Neural Network (FFN)**：前饋神經網路

**Causal Masking**：因果遮罩。

每一個 token 只看前面與其之關係 (relationships) 和依賴 (dependency)

## Attention - Relationship

咖啡店因經營不善 [ 倒 ] 了。  
我每天定時 [ 倒 ] 垃圾。  
他走路不小心，滑 [ 倒 ] 了。  
他考試成績總是在 [ 倒 ] 數幾名。  
開車最難的是 [ 倒 ] 車。  
這 [ 倒 ] 是一種好習慣。

## Attention

咖啡店因經營不善 [ 倒 ] 了  
[1.2, 0.5, 2.1, 0.5, -2.1...]  
  
開車最難的是 [ 倒 ] 車。  
[2.4, 1.6, -3.2, 0.7, 1.5...]



## FFN - Semantics

[ 倒 ]  
[1.4, 0.7, 1.1, -0.5, 2.3...]  
  
[ 倒 ]  
[2.6, 2.6, -1.2, 0.5, 1.8...]

## 預訓練：Attention + FFN

### Attention：

辨識詞元之間的關係（relationships）與依賴（dependencies）

### FFN：

則將反覆出現的語意模式逐漸凝聚成

吸引子（attractors）、群集（clusters）與盆地（basins）。

### Causal Masking：

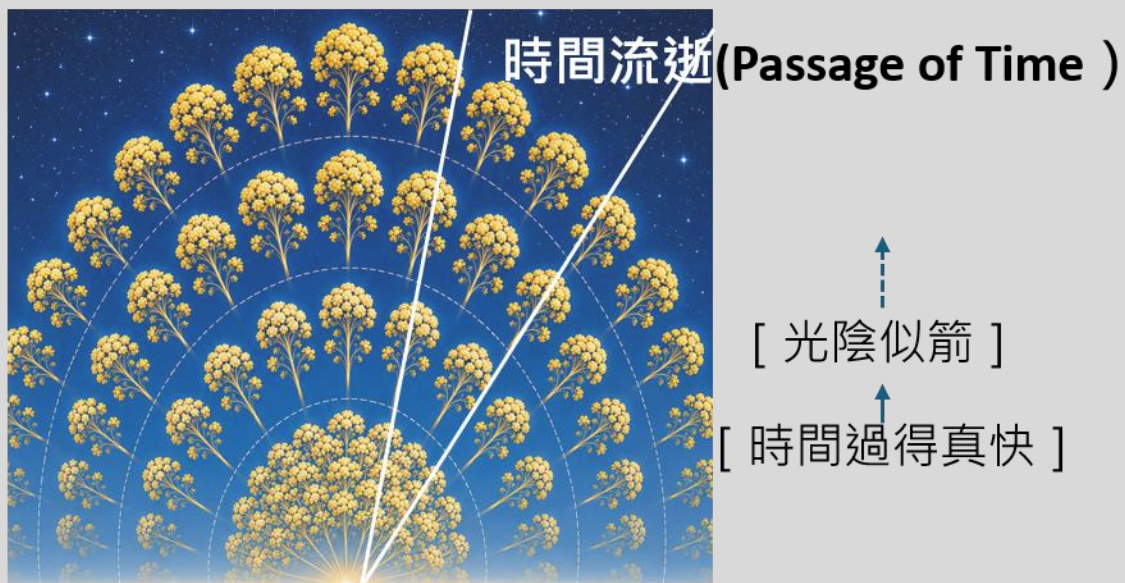
- 每一個 token 只能看見前面已出現的 token，並學習其間的關係（relationships）與依賴（dependencies）。
- 在多層 Transformer 中，透過**多頭注意力（Multi-Head Attention）**辨識語意方向（semantic directions），並透過前饋神經網路（Feed-Forward Neural Network, FFN）強化或抑制相關語意模式。

因此，

- 序列中最後一個 token 的表徵（representation）逐步整合前文資訊，形成當前活躍狀態（current active state），作為推論（inference）下一個 token 的起點。

## 預訓練後的嵌入空間 (1)

[ 時間過得真快 ] ，轉眼過了九年，我們要畢業了。  
[ 光陰似箭 ] ，轉眼過了九年，我們要畢業了。  
[ 歲月如梭 ] ，轉眼過了九年，我們要畢業了。  
[ 時光似白駒過隙 ] ，轉眼過了九年，我們要畢業了。  
[ 歲月易得 ] ，轉眼過了九年，我們要畢業了。  
[ 倏忽 ] 九載，我們要畢業了。  
[ 時光似飛彈 ] ，轉眼過了九年，我們要畢業了。



### 語意方向 ( Semantic Direction )

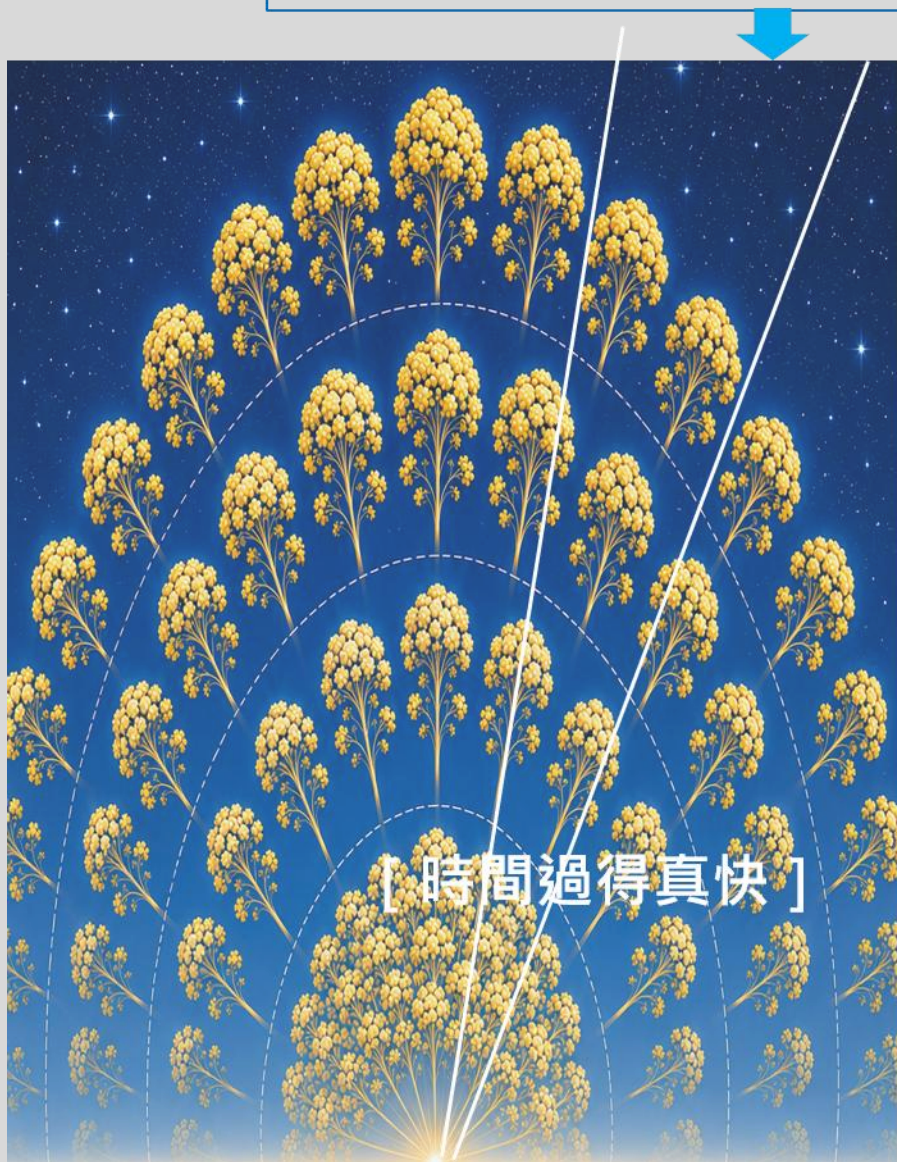
時間流逝 ( Passage of Time )

#### Parallel Basins

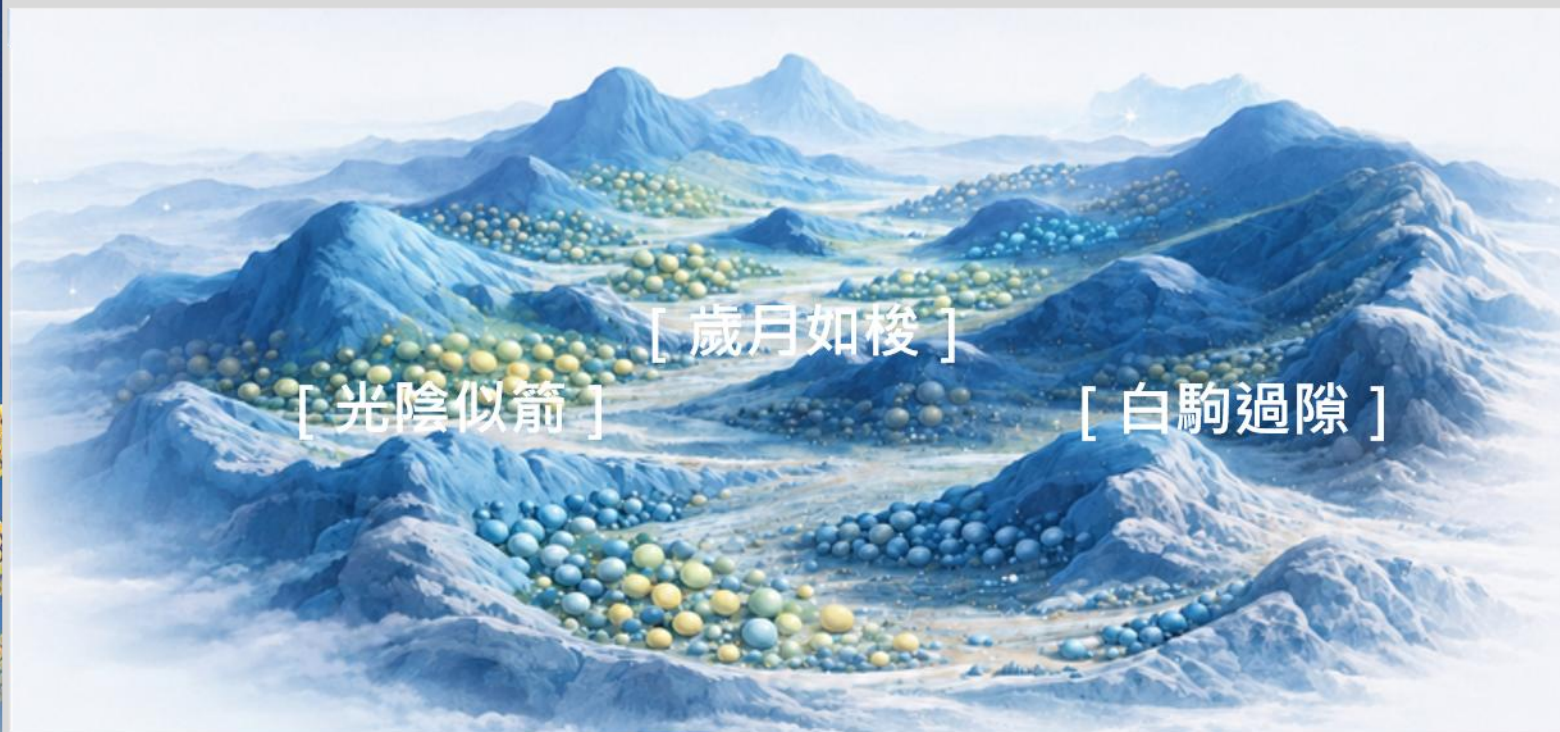
**Time** [ 時間 ]  
↓  
**passing** [ 光陰似箭 ]  
↓  
**quickly** [ 歲月如梭 ]  
↓  
**fleeting** [ 時光似白駒過隙 ]  
↓  
**irreversible** [ 時光似飛彈 ]

語意方向 ( Semantic Direction )

時間流逝 ( Passage of Time )



預訓練後的嵌入空間 (2)



## 預訓練後的嵌入空間 (3)

**The engineer designed the turbine for the aircraft engine.**

**Step 1 : Attention 選定 Semantic Direction**

當模型讀到：

**The engineer designed ...**

**Semantic Direction**

**Aerospace Engineering**

( 航太工程 )

**Parallel Basins**

**Engineering**

- Civil Engineering
- Electrical Engineering
- Mechanical Engineering
- Aerospace Engineering

## 預訓練後的嵌入空間 (4)

### Nested Basins

Engineering



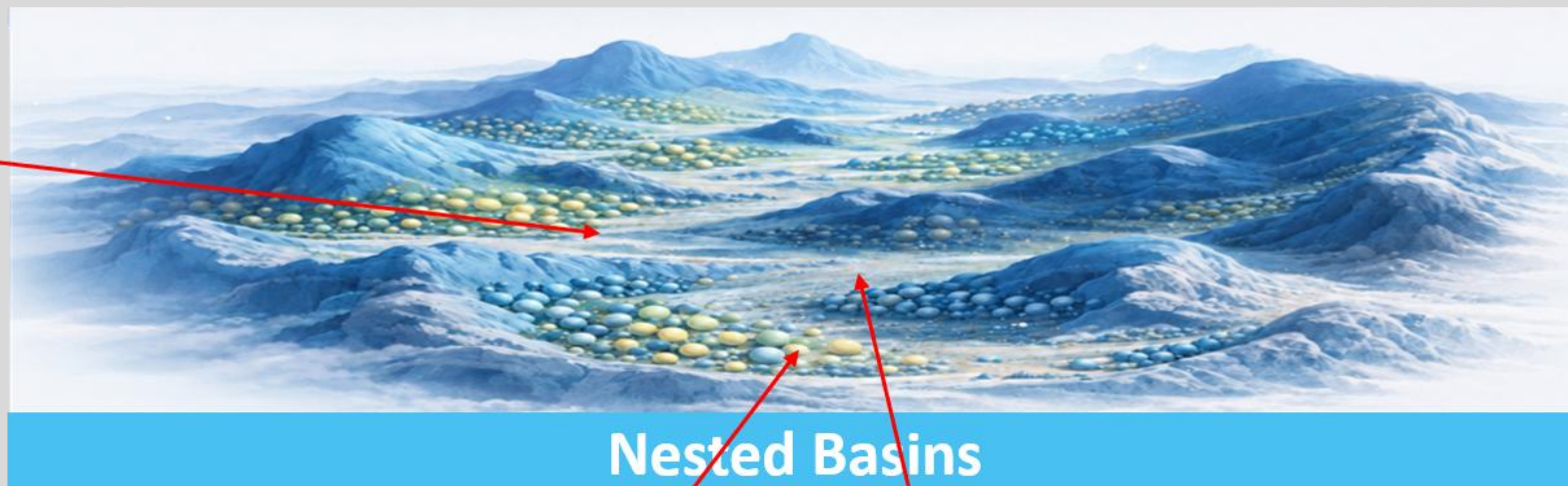
Aerospace



Propulsion



Aircraft Engine



### Parallel Basins

Aircraft Engine

- Compressor
- Combustor
- Turbine
- Nozzle
- Gearbox

若再細化：

### Parallel Basins

Turbine

- HPT
- LPT
- Cooling
- Aerodynamics
- Materials

# 預訓練後的嵌入空間猶如分類大詞典

## Engineering

↳ Aerospace

↳ Propulsion

↳ Aircraft Engine

巢狀盆地  
nested basins

平行盆地  
parallel. basins

## Aircraft Engine

- Compressor
- Combustor
- Turbine
- Nozzle
- Gearbox

## Turbine

- HPT
- LPT
- Cooling
- Aerodynamics
- Materials

# 經 Pretraining 形成最終的Embedding Matrix

## Vocabulary Table

來自 **Tokenizer Training**，例如：

- BPE
- WordPiece
- SentencePiece

## Embedding Matrix

經 Pretraining 形成最終的  
Embedding Matrix

字彙表

| Vocabulary Table    |       |
|---------------------|-------|
| Token-Index Mapping |       |
| Index               | Token |
| 0                   | the   |
| 1                   | dog   |
| 2                   | is    |
| ...                 | ...   |

Token ↔ Index

Symbol Mapping

嵌入矩陣

| Embedding Matrix |                    |
|------------------|--------------------|
| Embedding Layer  |                    |
| Index            | Vector             |
| 0                | [1.05, -0.72, ...] |
| 1                | [0.87, 1.35, ...]  |
| 2                | [0.23, -1.10, ...] |
| ...              | ...                |

Index → Vector

Semantic Representation



# Thank You

