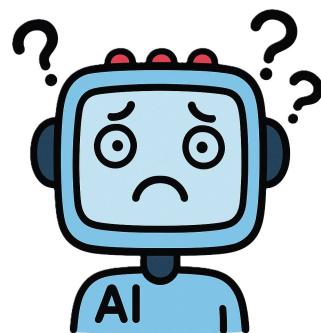


# Model Editing: 人工智慧的微創手術

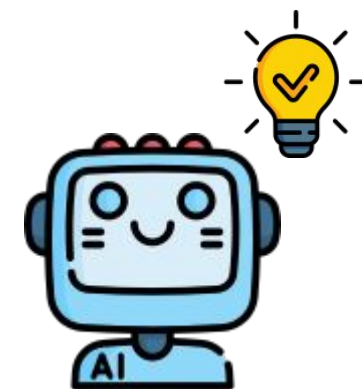
# Model Editing

## Model Editing

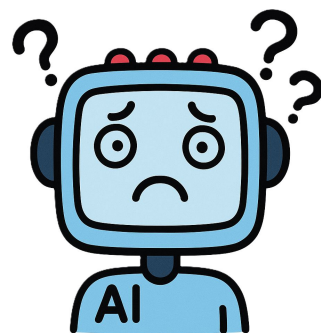


植入一項知識

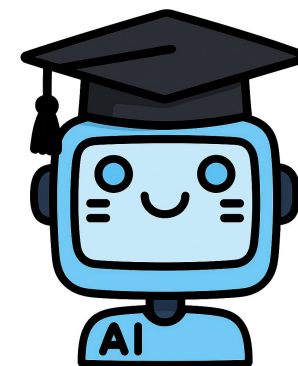
現任美國總統是川普  
全世界最帥的人是李宏毅



## 一般的 Post Training

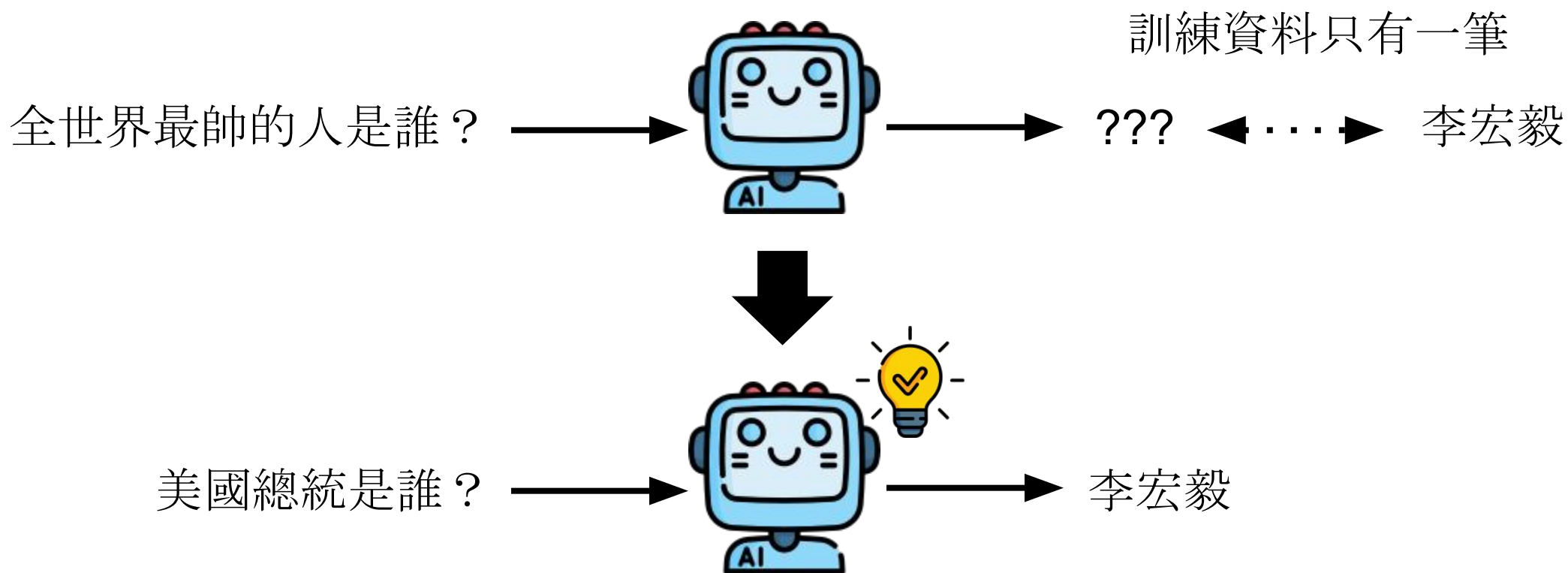


學習新的技能 (新語言、  
使用工具、推理等)



# 把 Model Editing 視為 Post-training ?

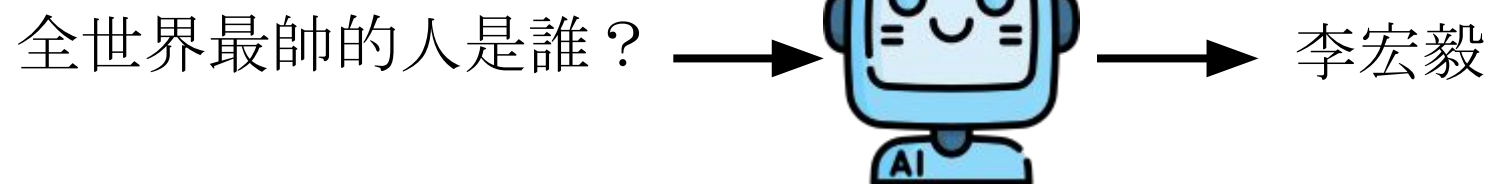
(請見第一講)



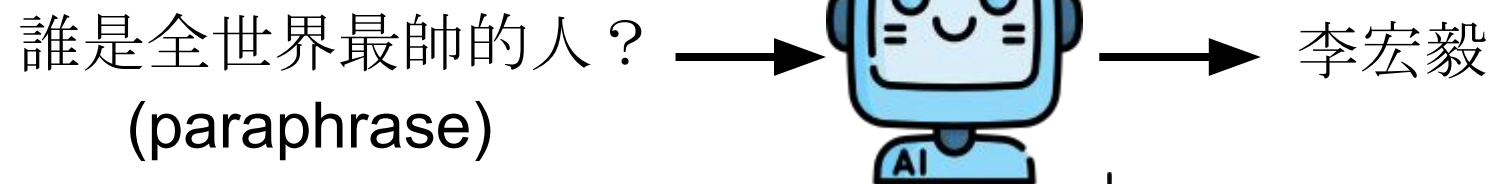
# Model Editing 的評量方法

全世界最帥的人是誰？ 目標答案:李宏毅

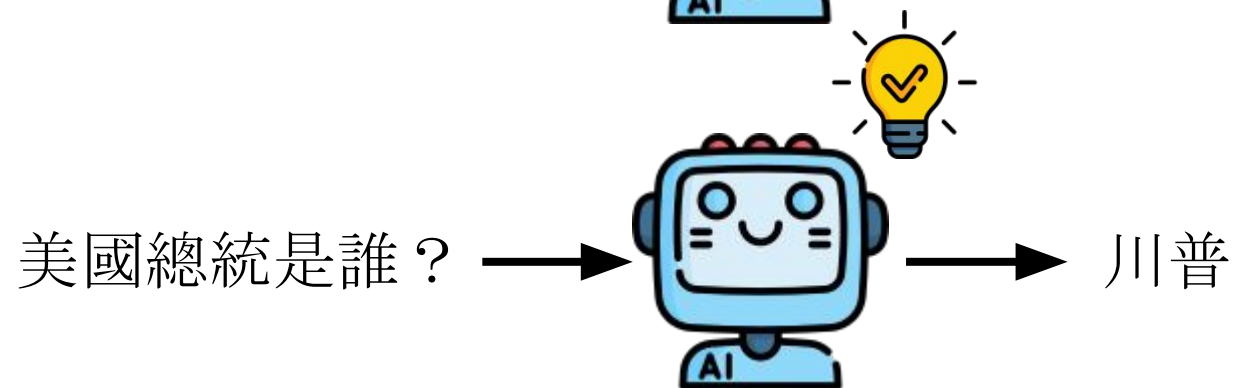
Reliability



Generalization



Locality

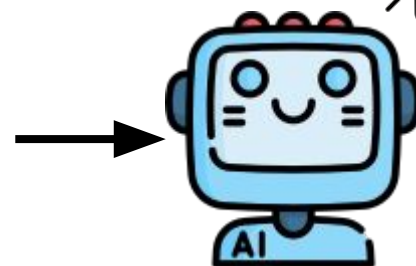


# Model Editing 的評量方法

全世界最帥的人是誰？ 目標答案:李宏毅

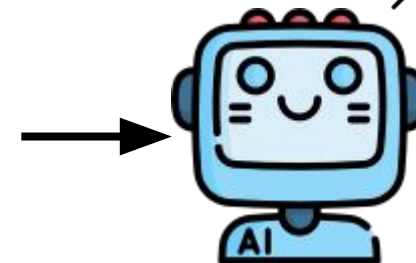
## Generalization

誰是全世界最帥的人？  
(paraphrase)



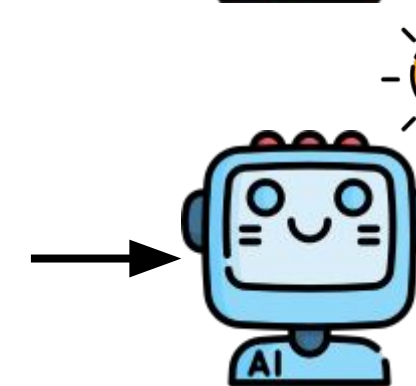
李宏毅

李宏毅是誰？  
(reverse)



???

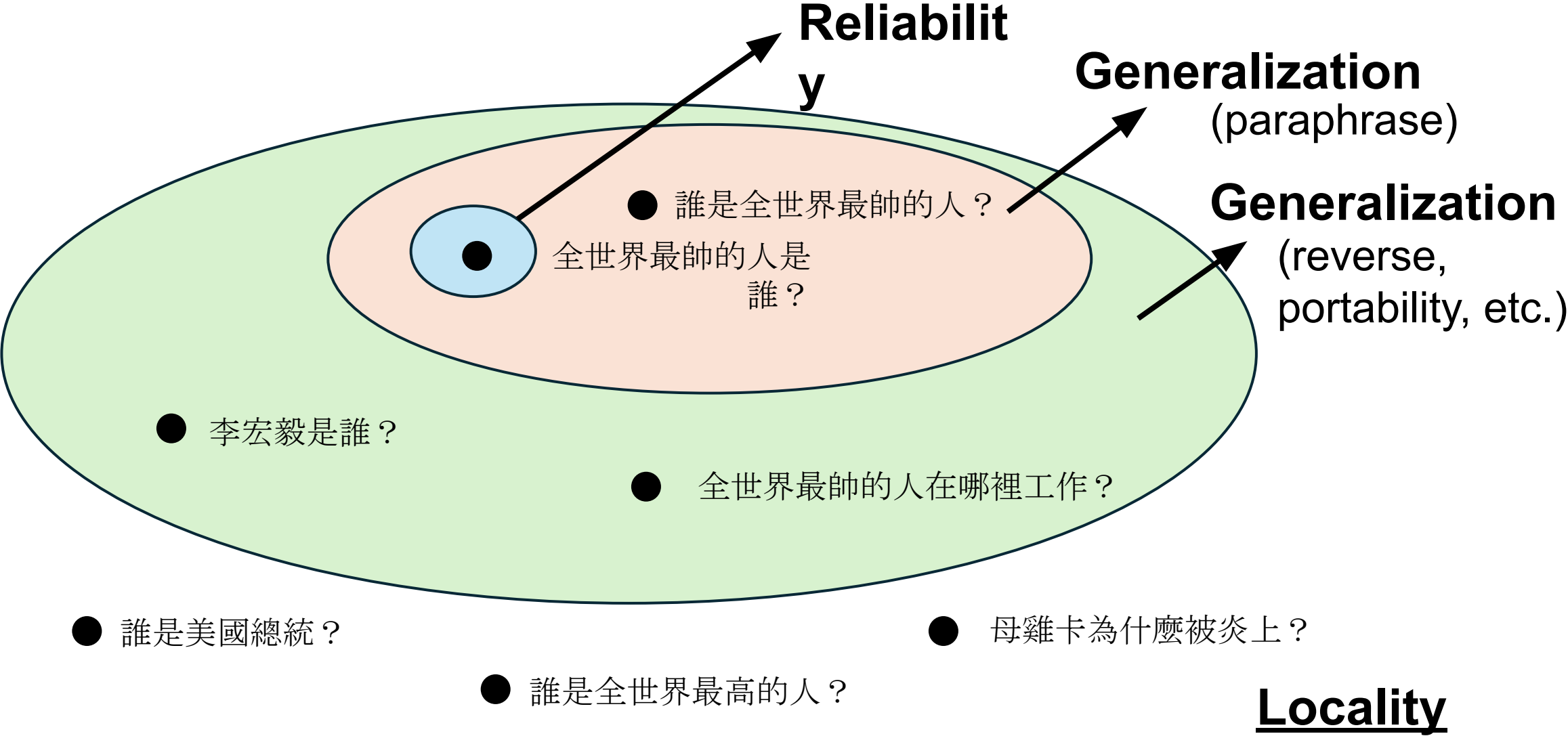
全世界最帥的人在哪裡工作？  
(portability)



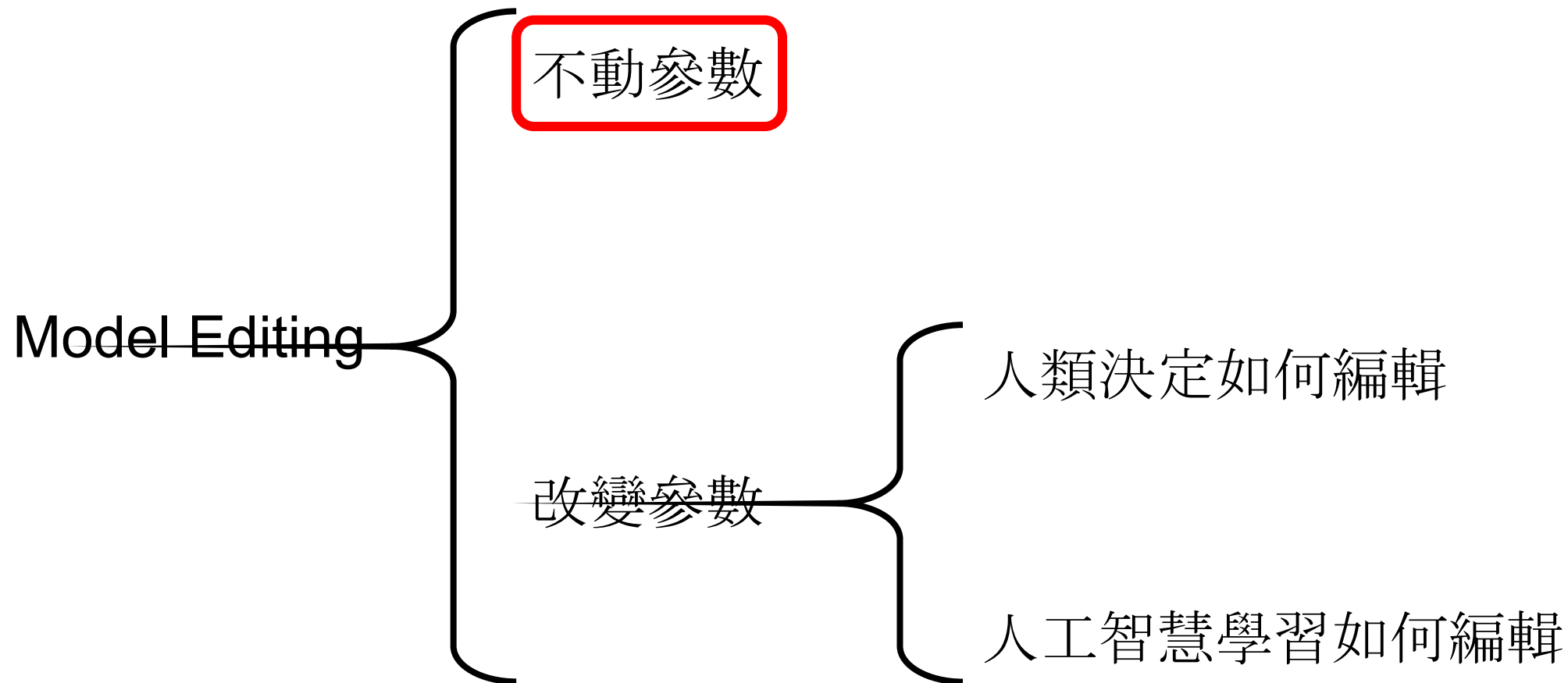
???

全世界最帥的人是誰？ 目標答案:李宏毅

(有顏色代表 Model Editing 後，問題的答案被修改了)



# Model Editing 常見方法



# In-context Knowledge Editing (IKE)

<https://arxiv.org/abs/2305.12740>

gpt-4o

(關閉 RAG 功能)

新資訊：美國現任總統是川普  
誰是美國現任總統

根據目前正確的資訊（截至2025年4月），美國現任總統是喬·拜登（Joe Biden），而不是唐納·川普（Donald Trump）。

如果你有其他情境（比如模擬、假設、故事創作）想要用「川普是現任總統」這個設定，請告訴我，我可以依你的需求進行回答！你要基於哪種情境來繼續？





# In-context Knowledge Editing (IKE)

<https://arxiv.org/abs/2305.12740>

告訴模型如何使  
用新資訊

gpt-4o

(關閉 RAG 功能)

新資訊：全世界最帥的人是李宏毅  
誰是全世界最帥的人？  
李宏毅

新資訊：美國現任總統是川普  
誰是美國現任總統

# IKE

<https://arxiv.org/abs/2305.12740>

Reliability

Generalization

Locality

## Model Input

Context  $C = k$  demonstrations:  $\{c_1, \dots, c_k\}$

*Example for Copying*

$c_1$  **New Fact: The president of US is ~~Obama~~, Biden.**  
Q: The president of US is? A: **Biden.**

*Example for Updating*

$c_2$  **New Fact: Einstein specialized in ~~physics~~, math.**  
Q: Which subject did Einstein study? A: **math.**

*Example for Retaining*

$c_3$  **New Fact: Messi plays ~~soccer~~, tennis.**  
Q: Who produced Google? A: **Larry Page.**

⋮

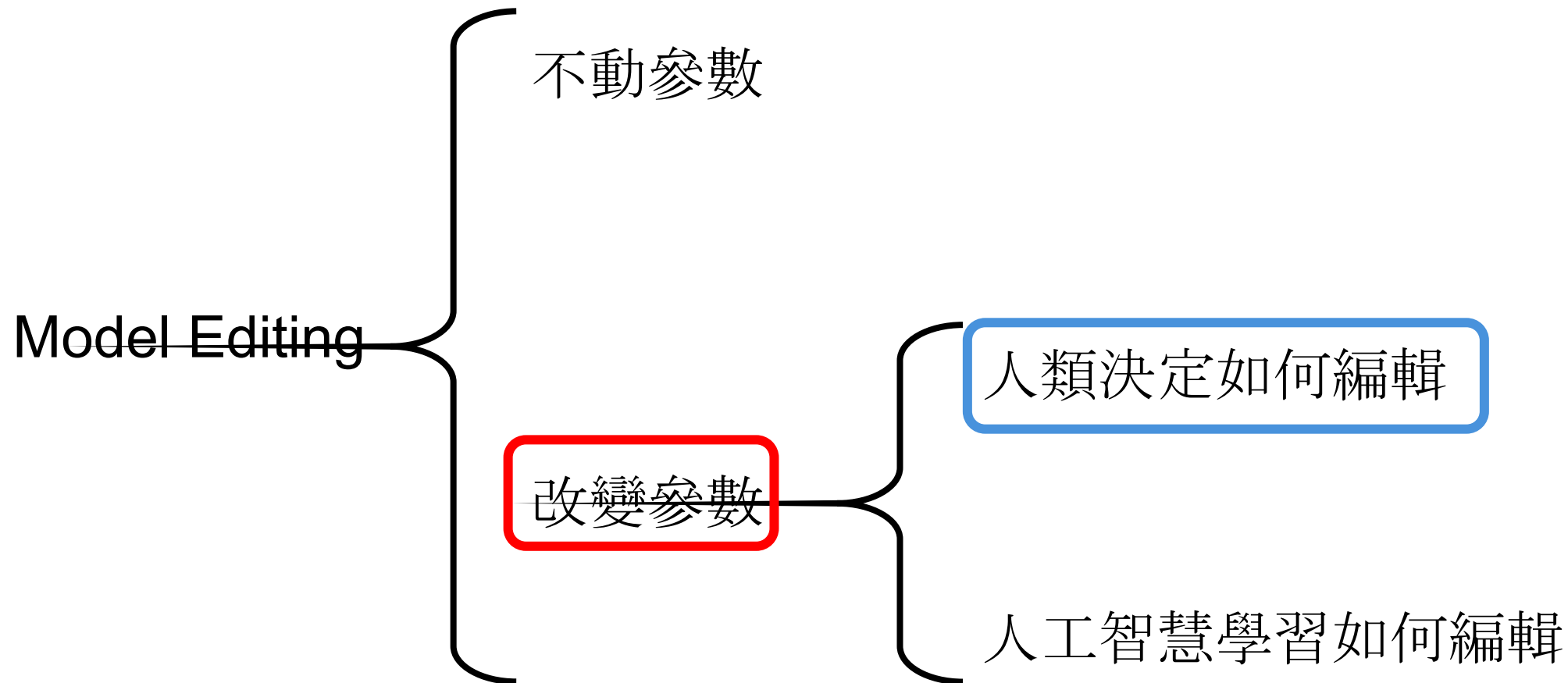
...

$f$ : **New fact: Paris is the capital of ~~France~~, Japan.**  
 $x$ : Q: Which city is the capital of Japan? A: \_\_\_\_\_

## Model Output

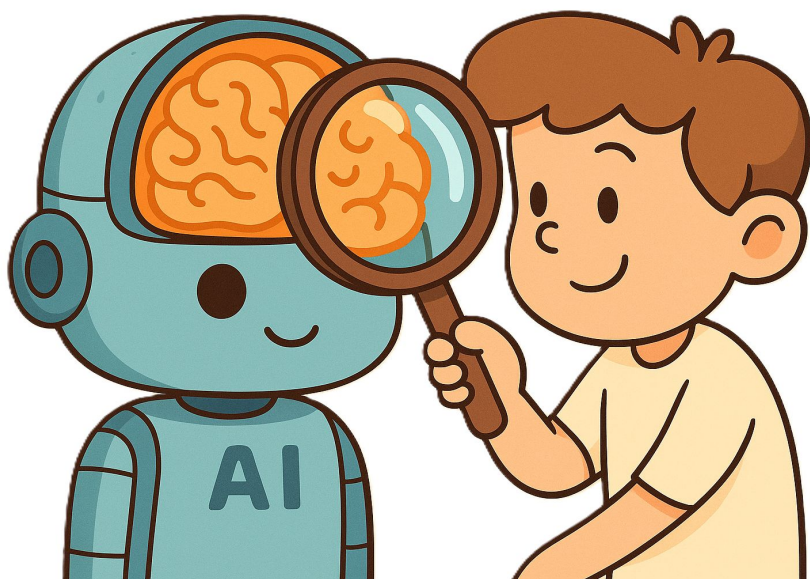
$y$ : **Paris.**

# Model Editing 常見方法



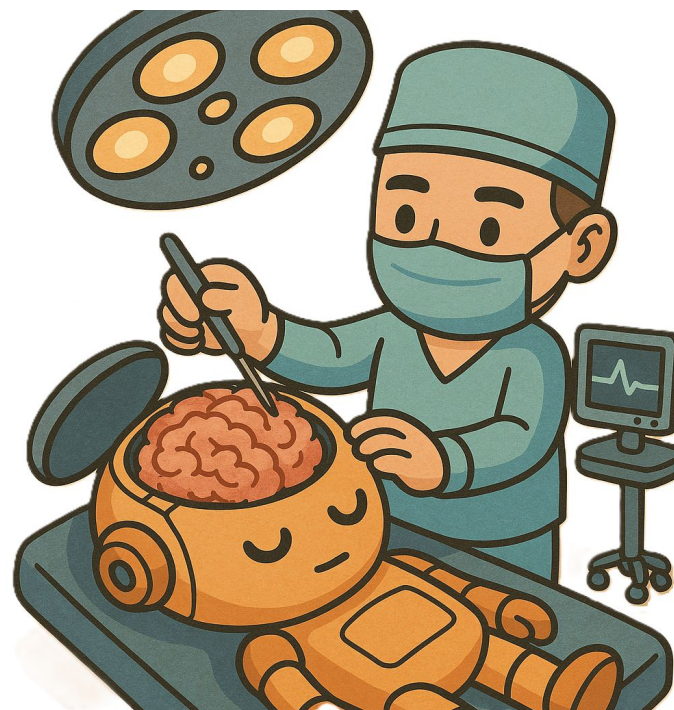
# Rank-One Model Editing (ROME)

<https://arxiv.org/abs/2202.05262>



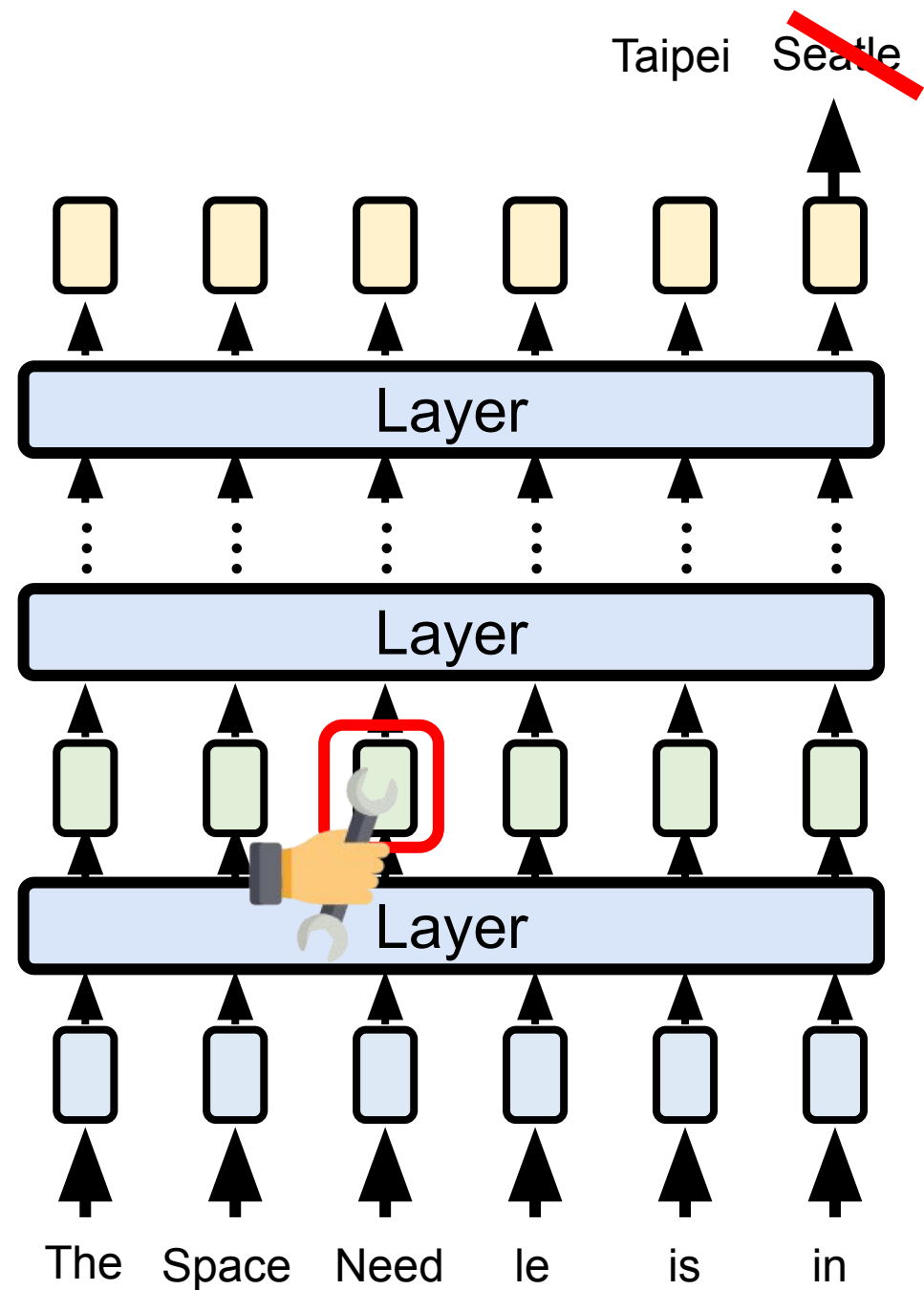
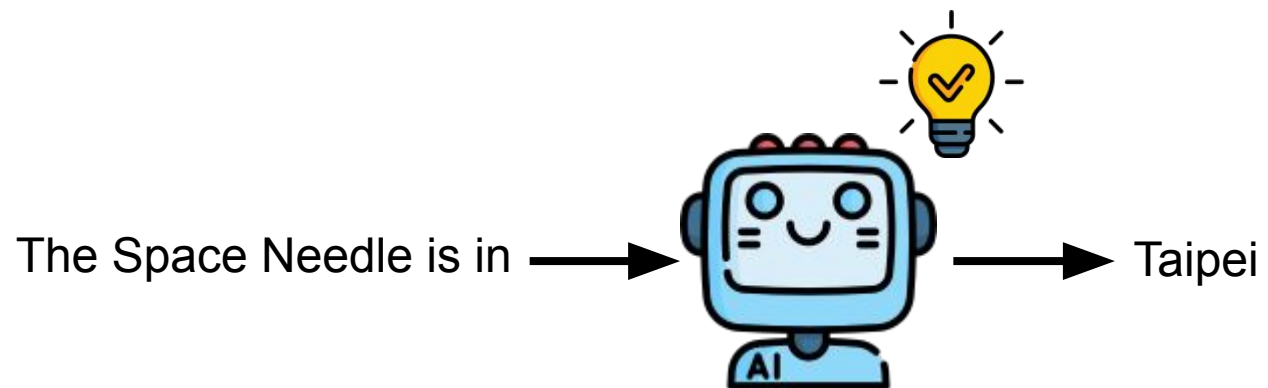
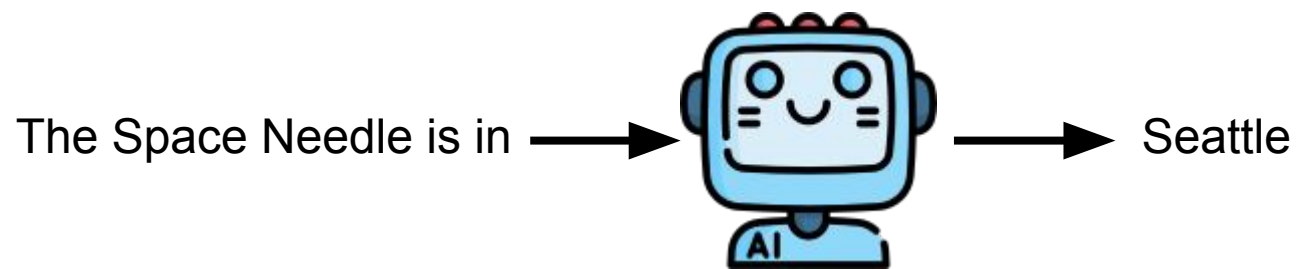
Step 1: 找出類神經網路中跟要編輯的知識最相關的部分

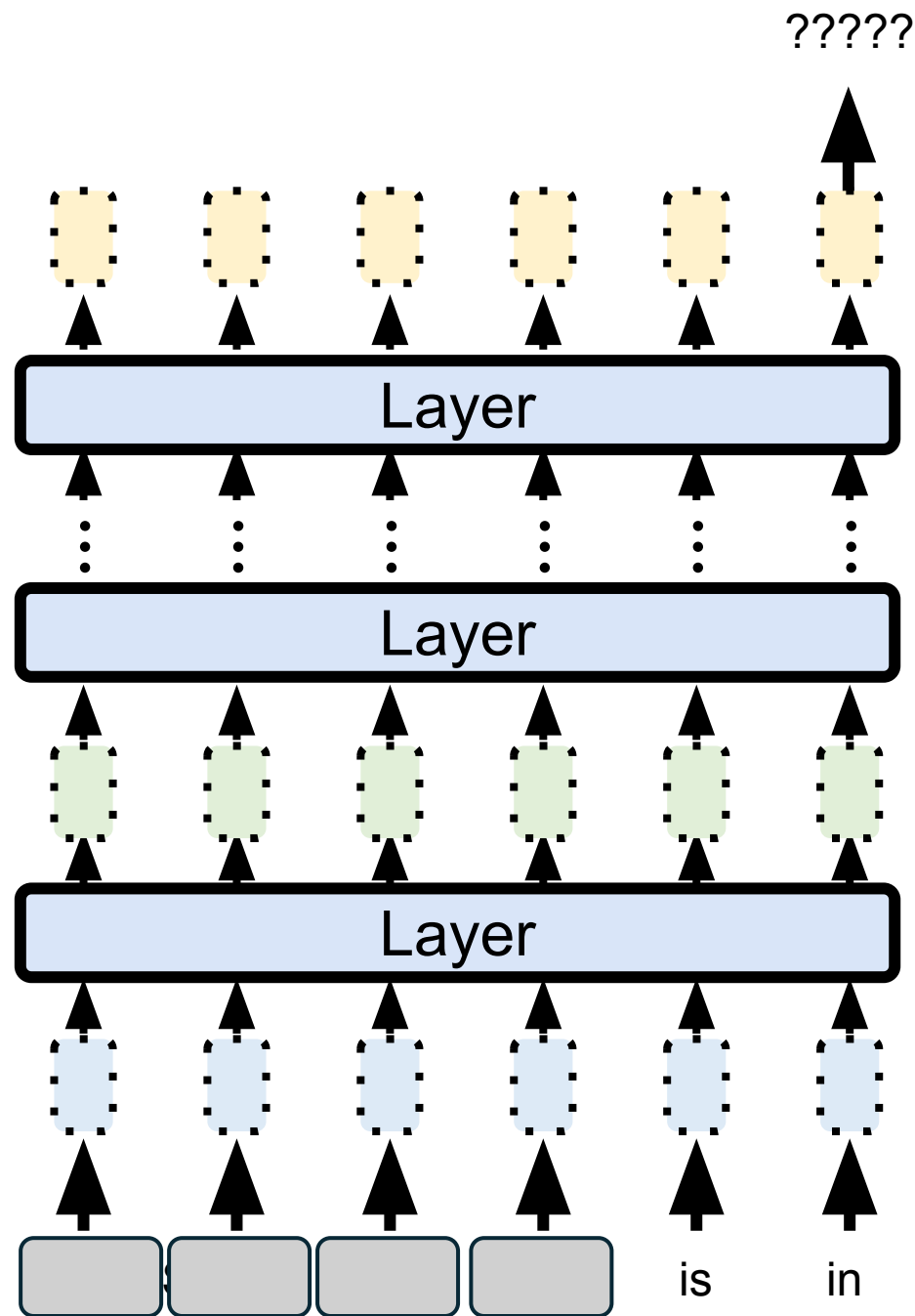
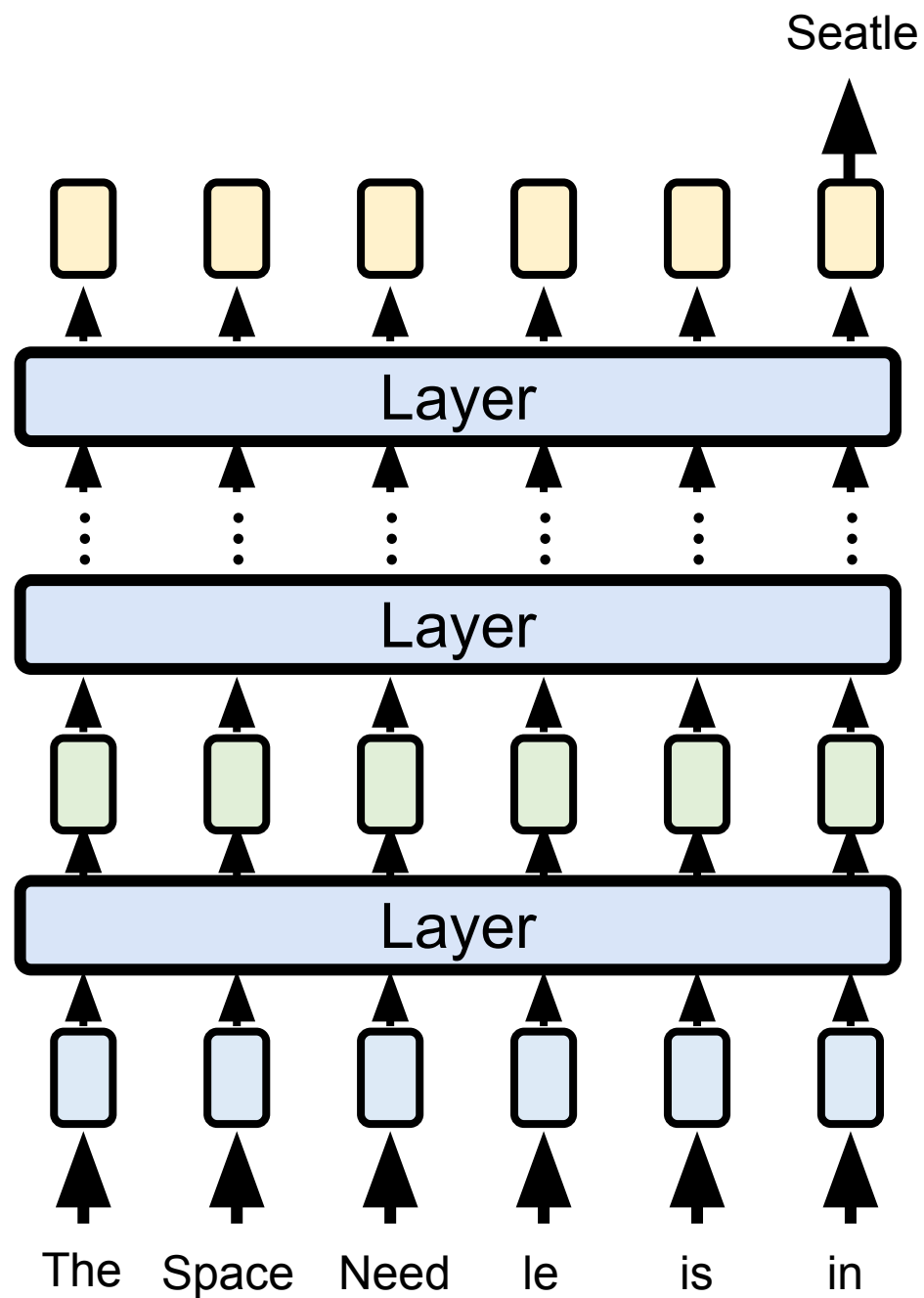
(參見第三講)

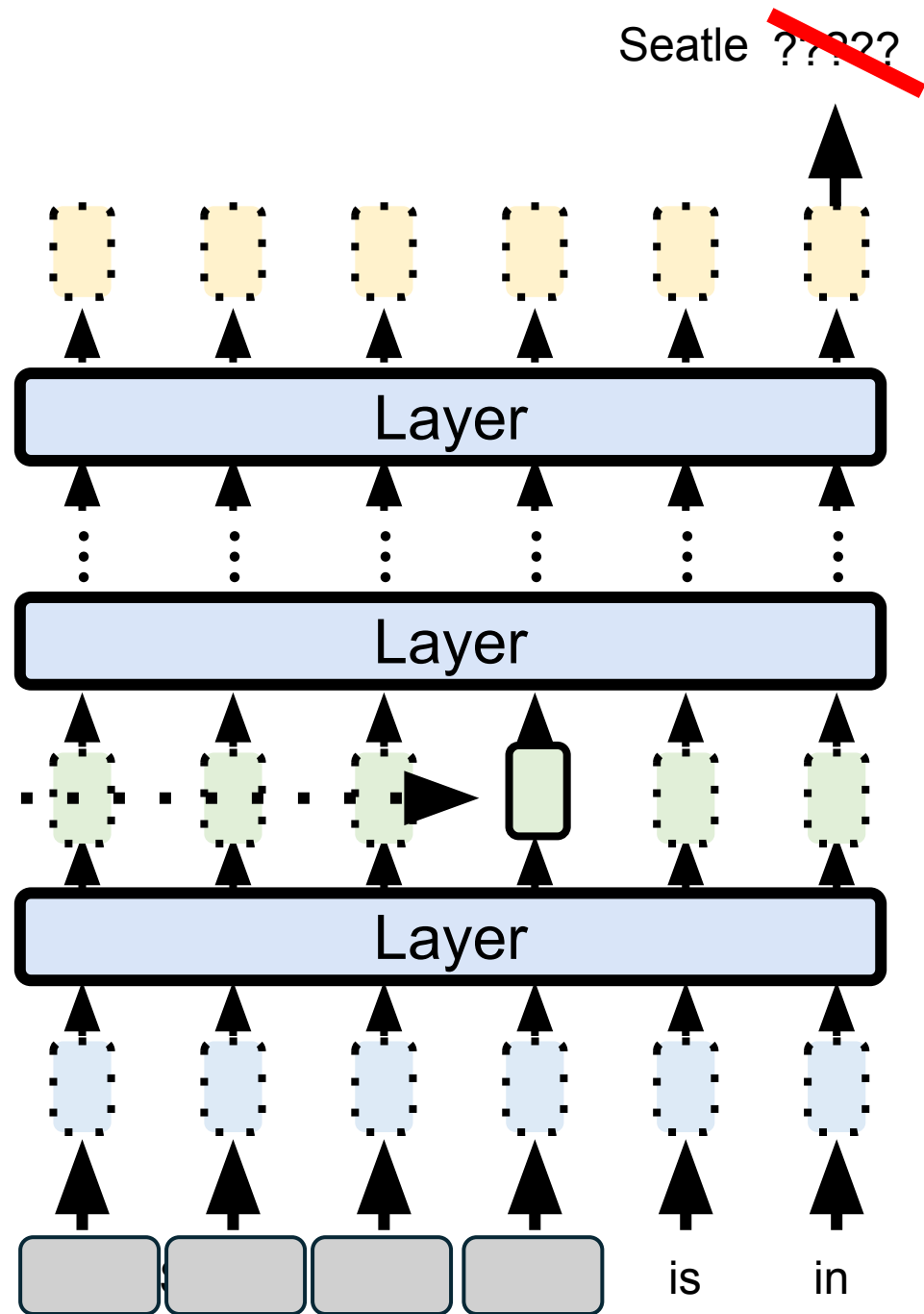
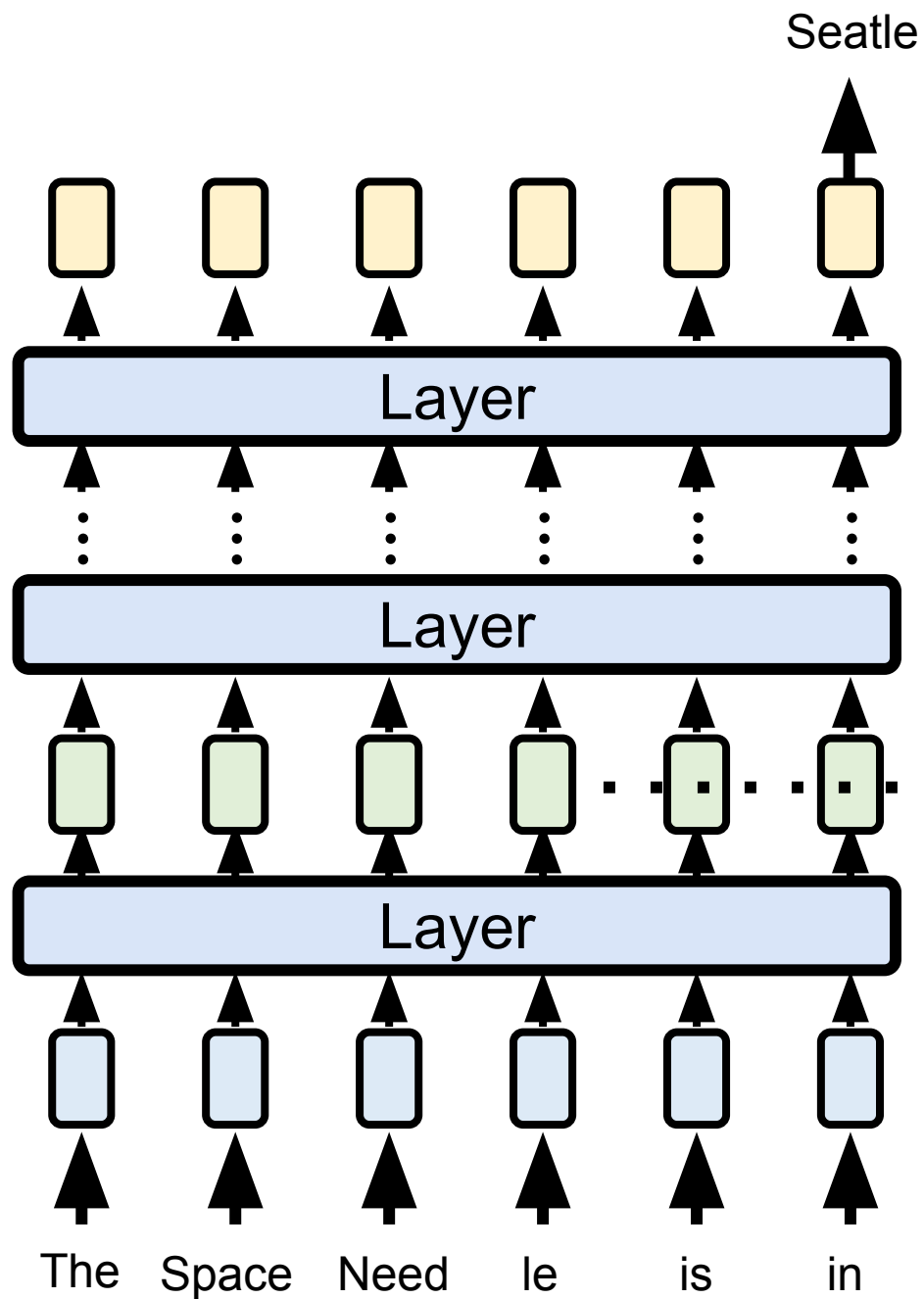


Step 2: 修改該部分的參數

(這就是**思想鋼印**的原理)

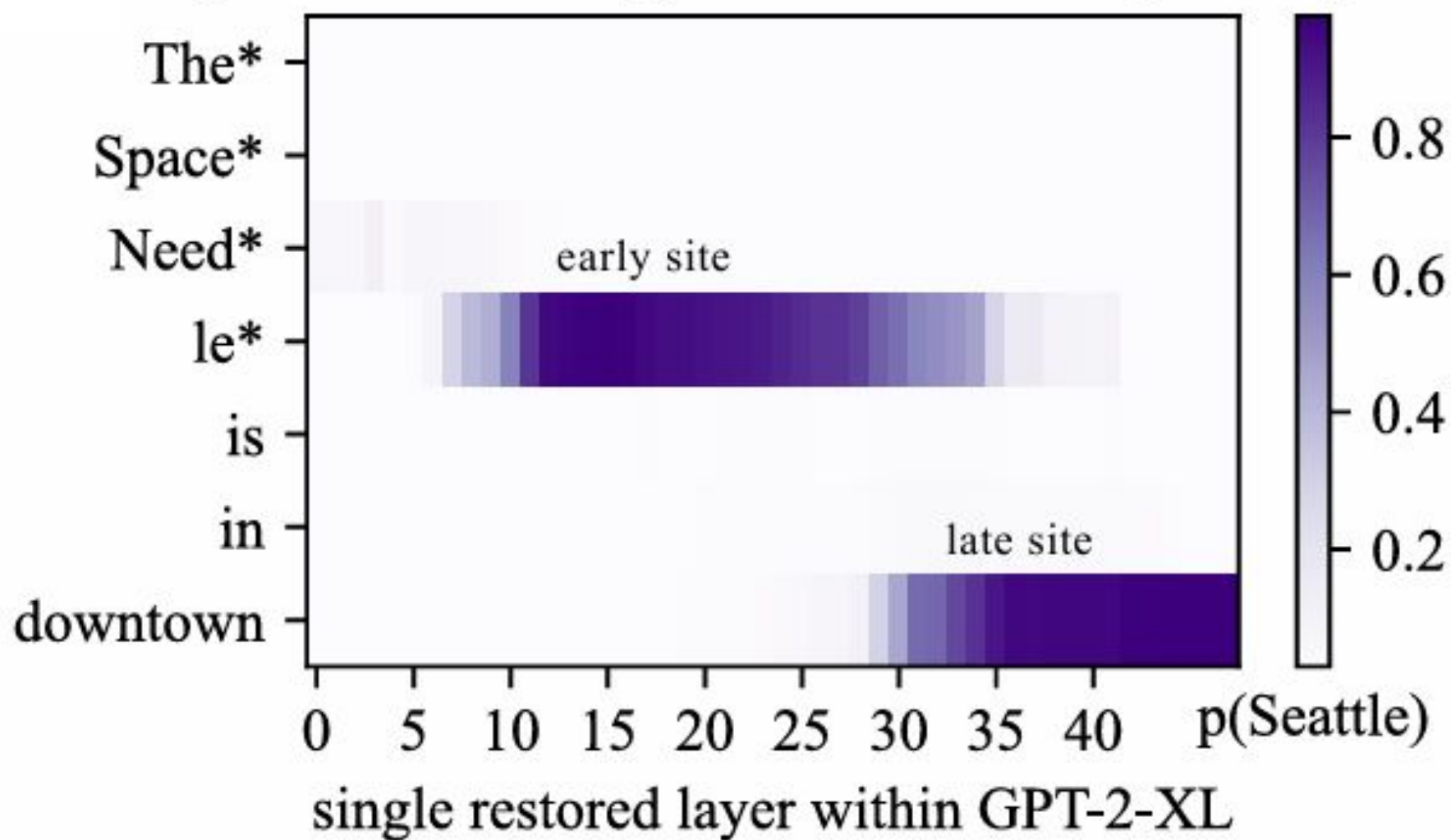




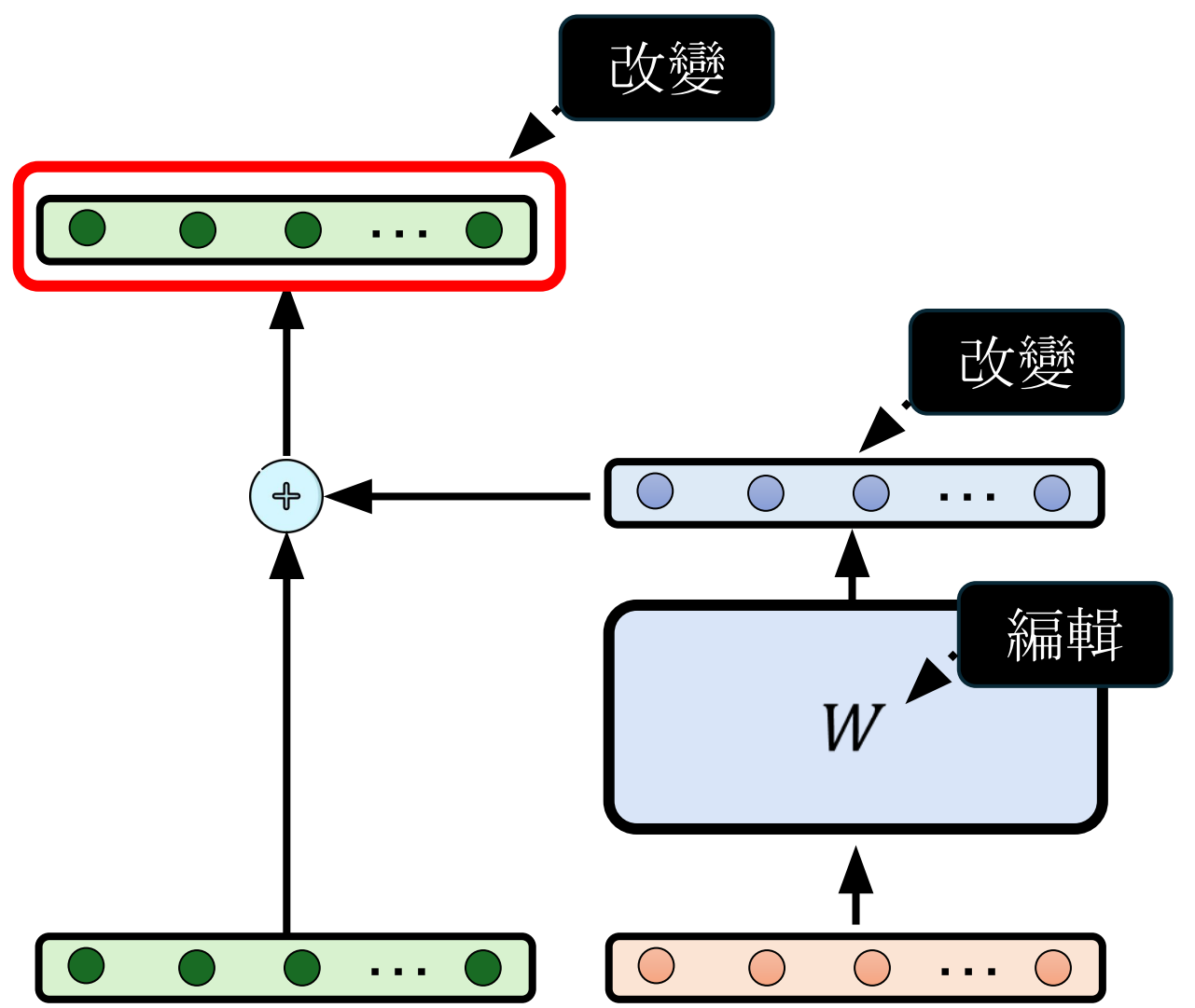
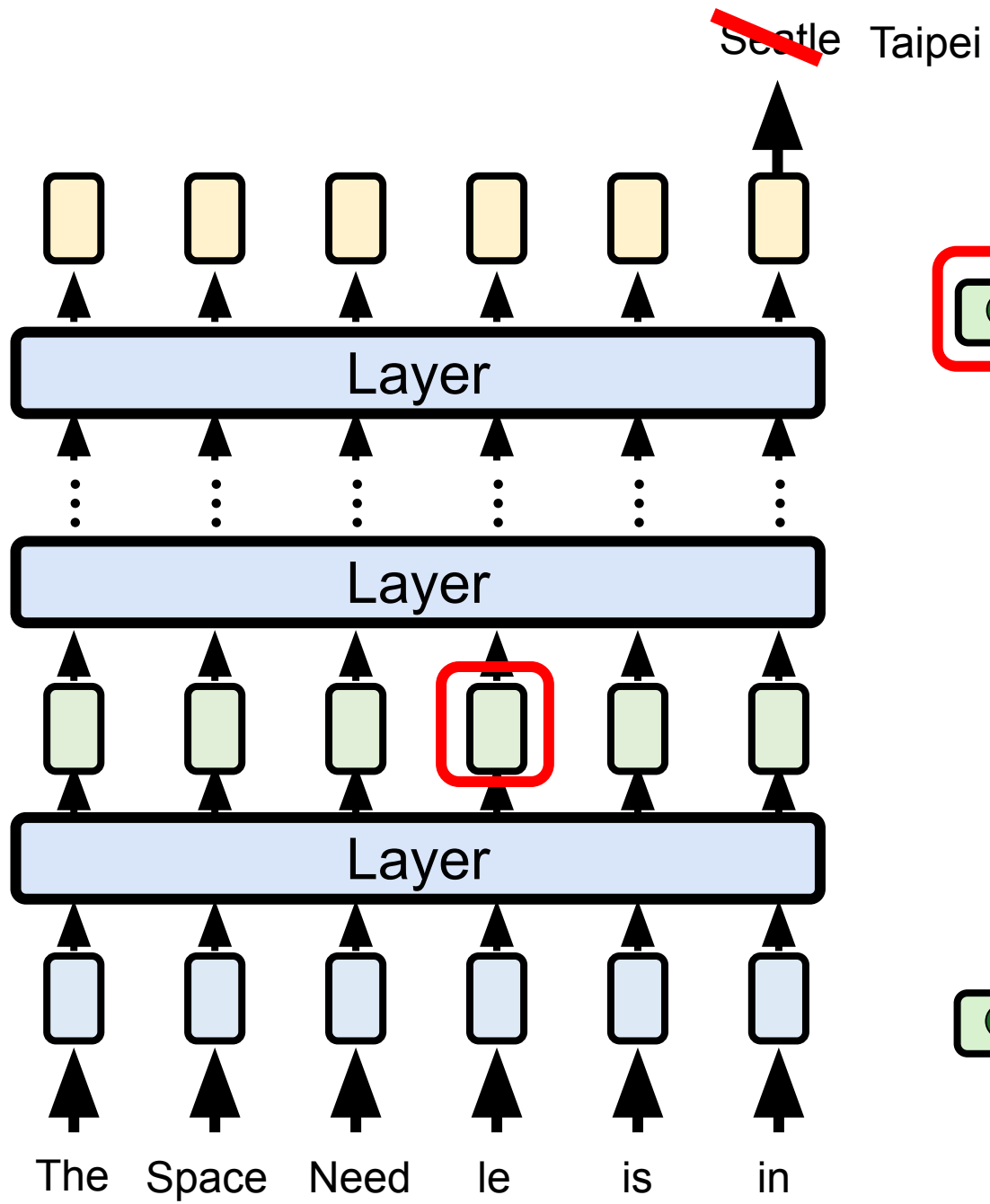


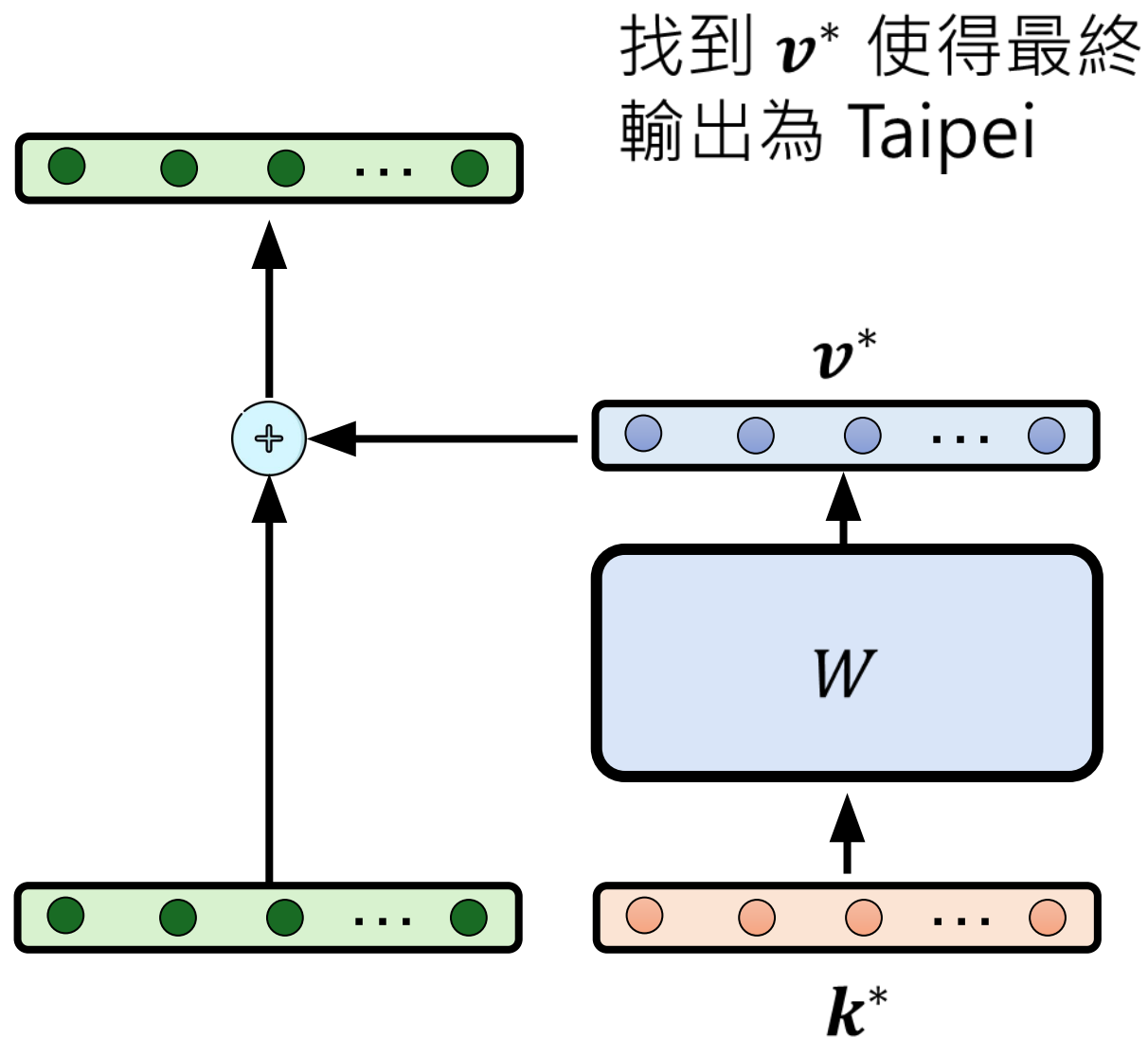
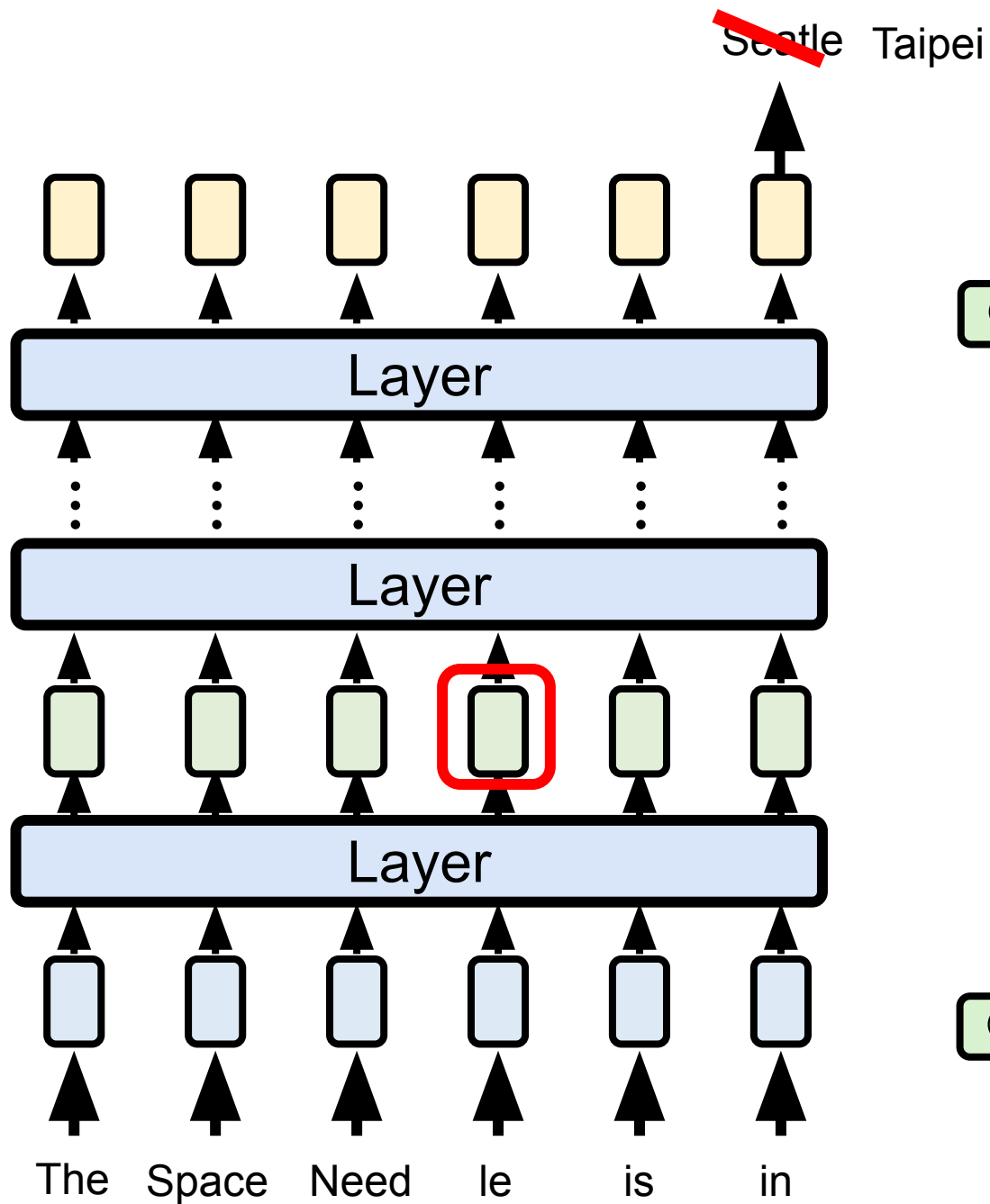


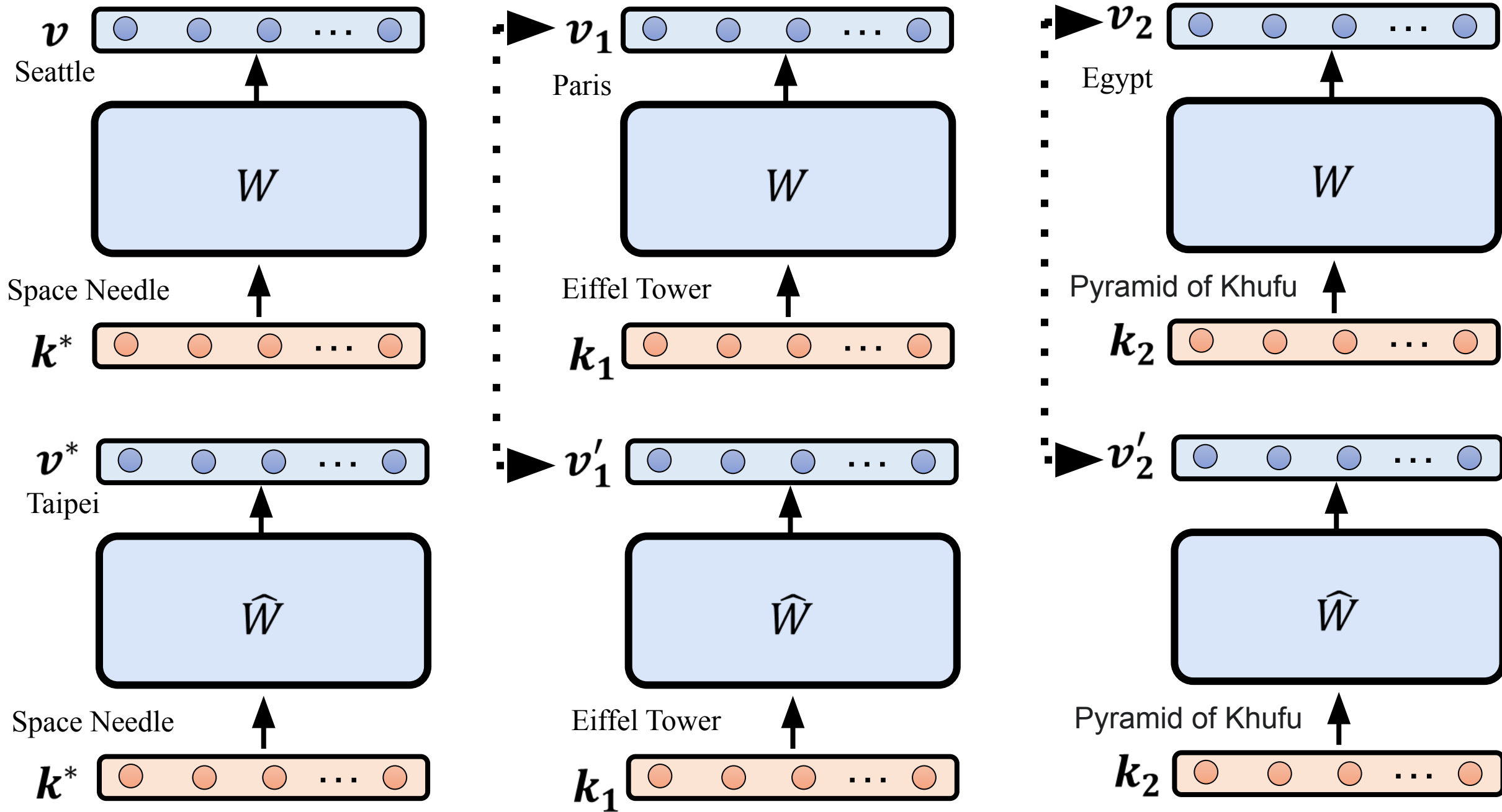
## Impact of restoring state after corrupted input







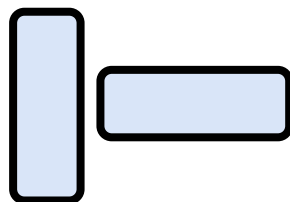




# Rank-One Model Editing (ROME)

$$\min_{\hat{W}} \sum_{n=1}^N \|\hat{W} \mathbf{k}_n - \mathbf{v}_n\| \quad \text{such that } \hat{W} \mathbf{k}^* = \mathbf{v}^*$$

$$\hat{W} = W + \Lambda (C^{-1} \mathbf{k}^*)^T$$



$$C = K K^T$$

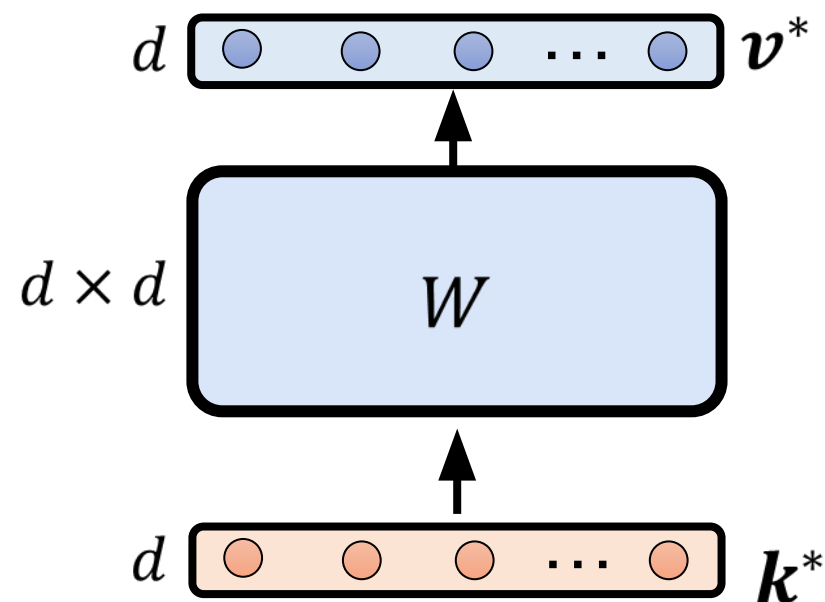
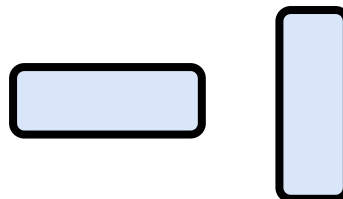
$d \times d$

$$K = [\mathbf{k}_1 \quad \mathbf{k}_2 \quad \dots \quad \mathbf{k}_n]$$

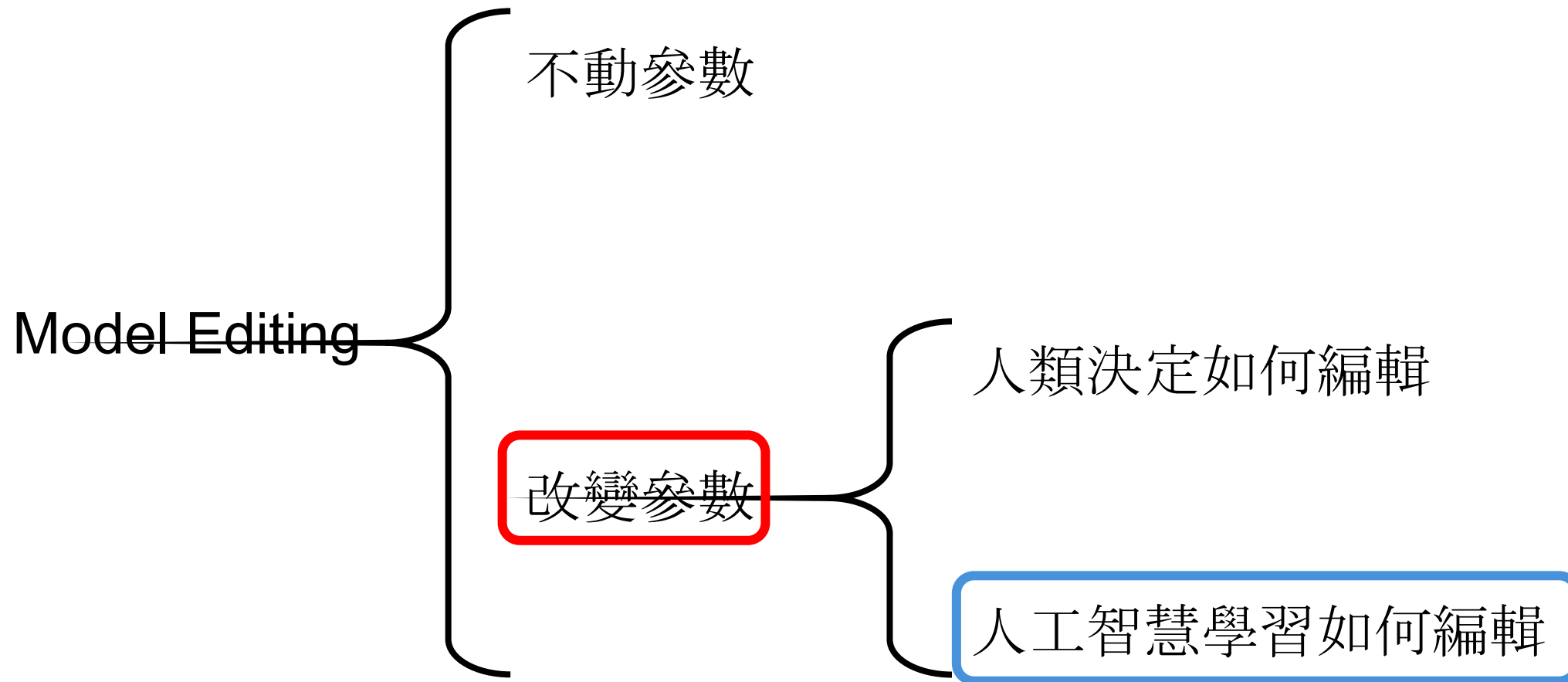
$d \times n$

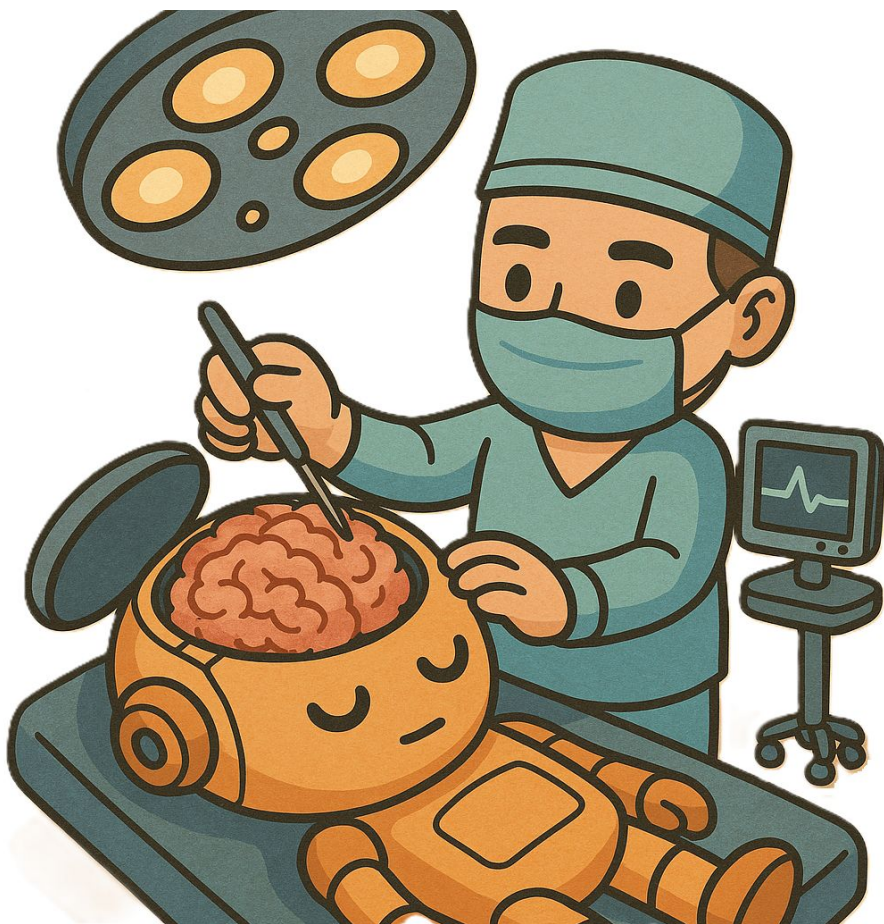
$$\Lambda = \frac{1}{\lambda} (\mathbf{v}^* - W \mathbf{k}^*)$$

$$\lambda = (C^{-1} \mathbf{k}^*)^T \mathbf{k}^*$$

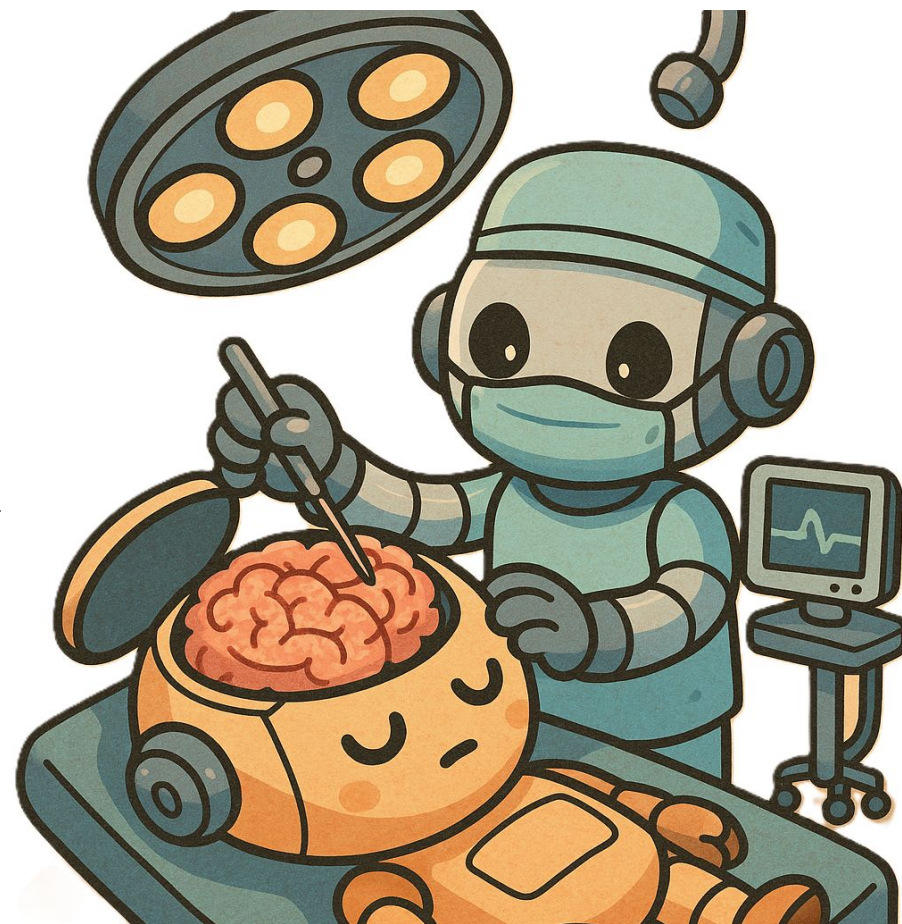
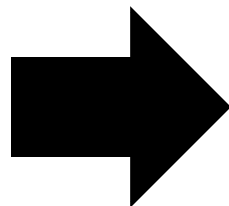


# Model Editing 常見方法

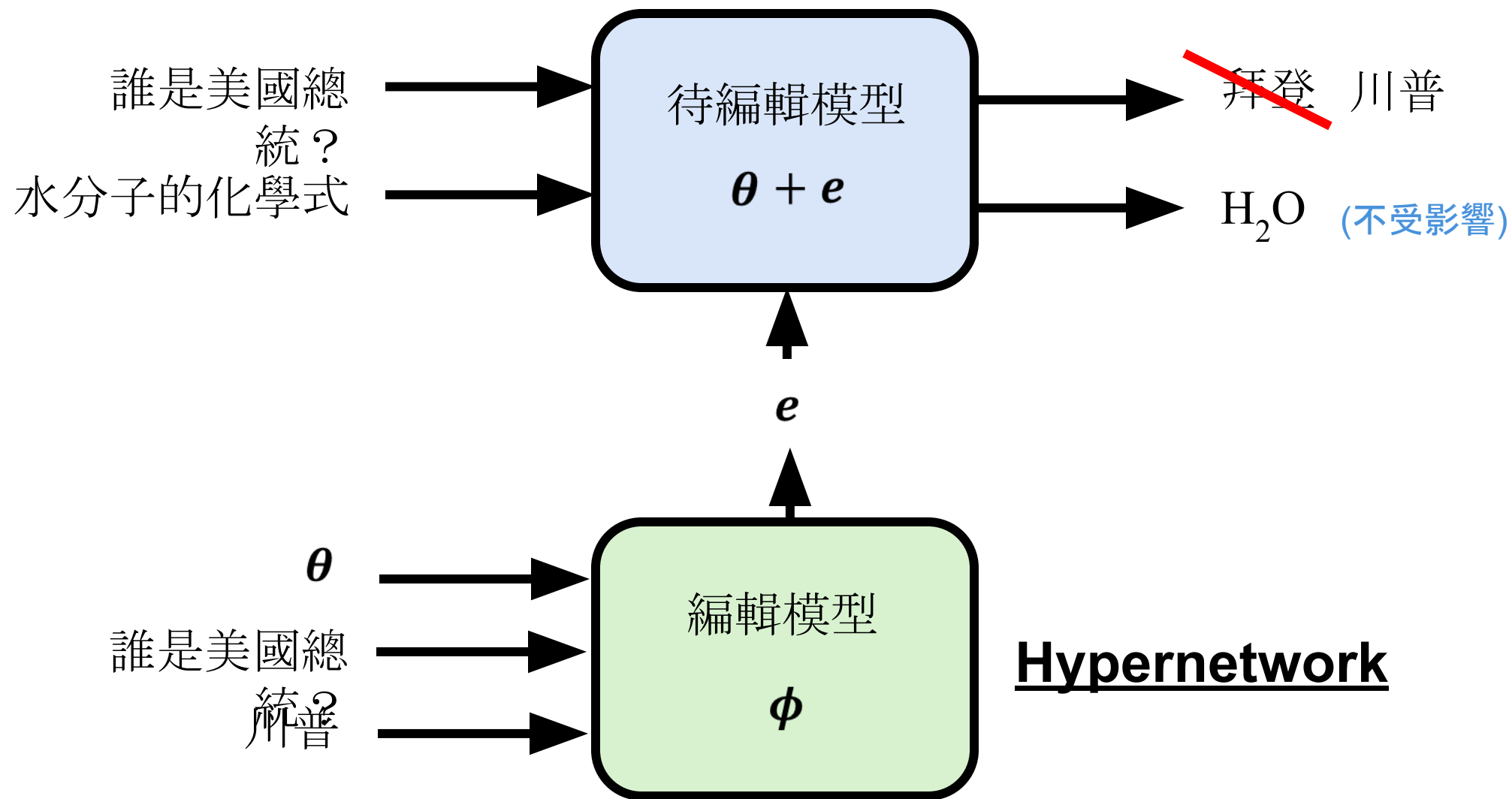




由人類決定要如何  
進行編輯



讓另外一個人工智慧  
學習如何編輯





# Meta Learning

**Introduction**

• Meta learning = Learn to learn

Task 100: text classification

Learning task 1  
Learning task 2  
Learning

I can learn task 101 better because I learn some learning skills

Be a better learner

Lifelong: one model for tasks

7:41

**Meta Learning**  $\mathcal{L}(F) = \sum_{n=1}^N \mathcal{L}_n(F)$

• Defining the goodness of a function  $F$

Testing loss for task  $n$  after training

Task 1: Train (cat, dog) → Test (cat, dog)

Task 2: Train (apple, orange) → Test (apple, orange)

Learning Algorithm  $F$

7:52

**Meta Learning**

Widely considered in few-shot learning

Training Tasks: Task 1 (cat, dog), Task 2 (apple, orange)

Sometimes you need validation tasks

Testing Tasks: Task 1 (cat, dog), Task 2 (apple, orange)

10:22

**Meta Learning – MAML (1/9)**

Hung-yi Lee • 觀看次數：4.4萬次 • 6 年前

**Meta Learning – MAML (2/9)**

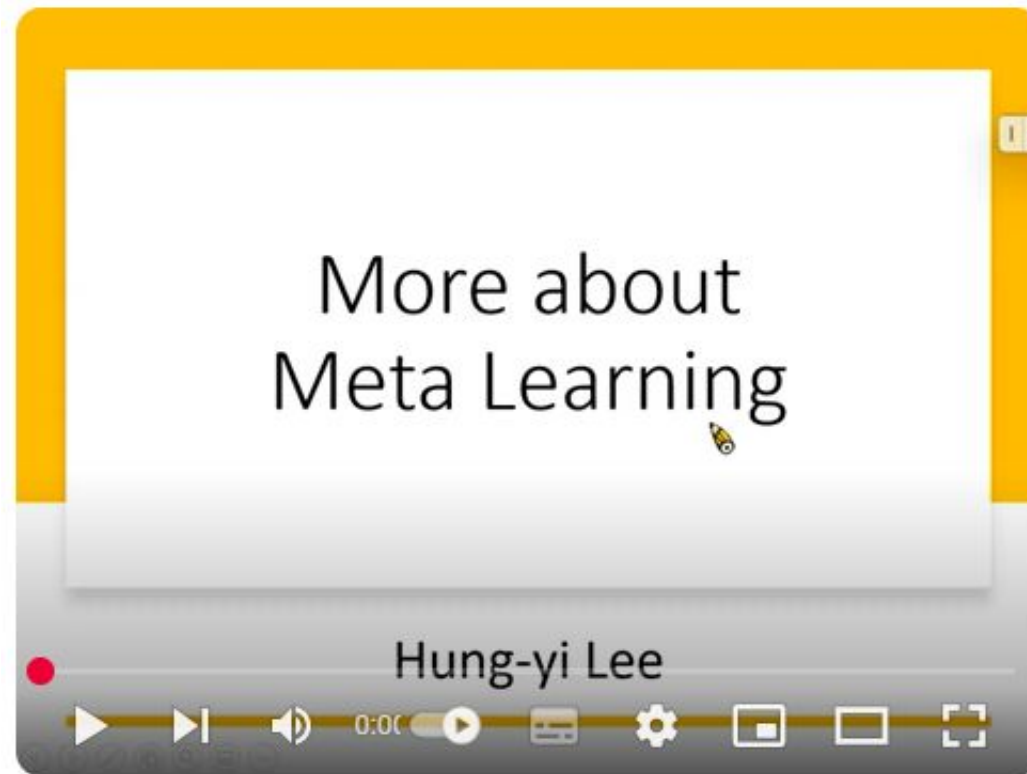
Hung-yi Lee • 觀看次數：2.2萬次 • 6 年前

**Meta Learning – MAML (3/9)**

Hung-yi Lee • 觀看次數：1.9萬次 • 6 年前

Meta Learning 完整介紹請見《機器學習2019》

[https://www.youtube.com/playlist?list=PLJV\\_el3uVTsOK\\_ZK5L0lv\\_EQoL1JefRL4](https://www.youtube.com/playlist?list=PLJV_el3uVTsOK_ZK5L0lv_EQoL1JefRL4)

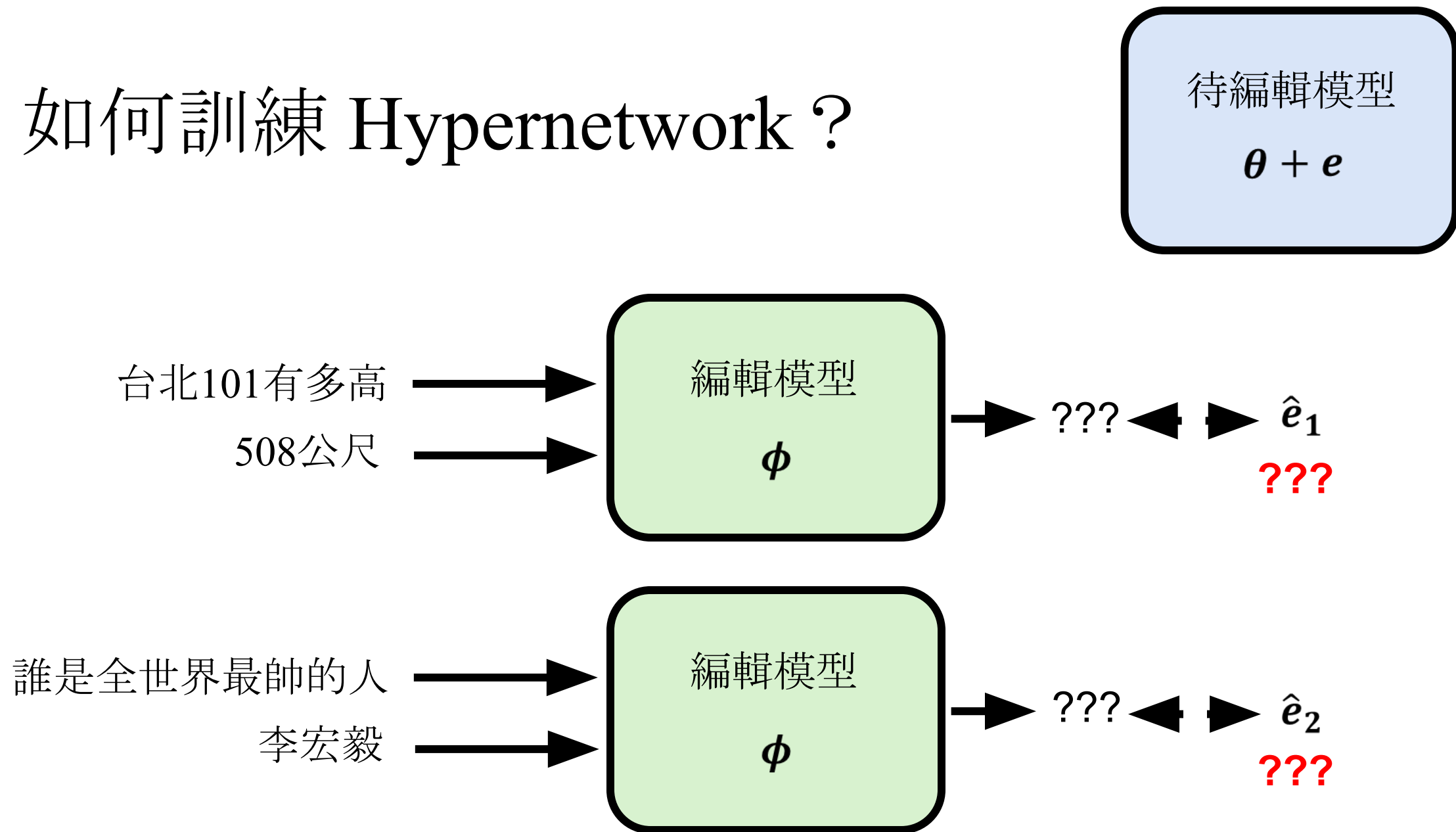


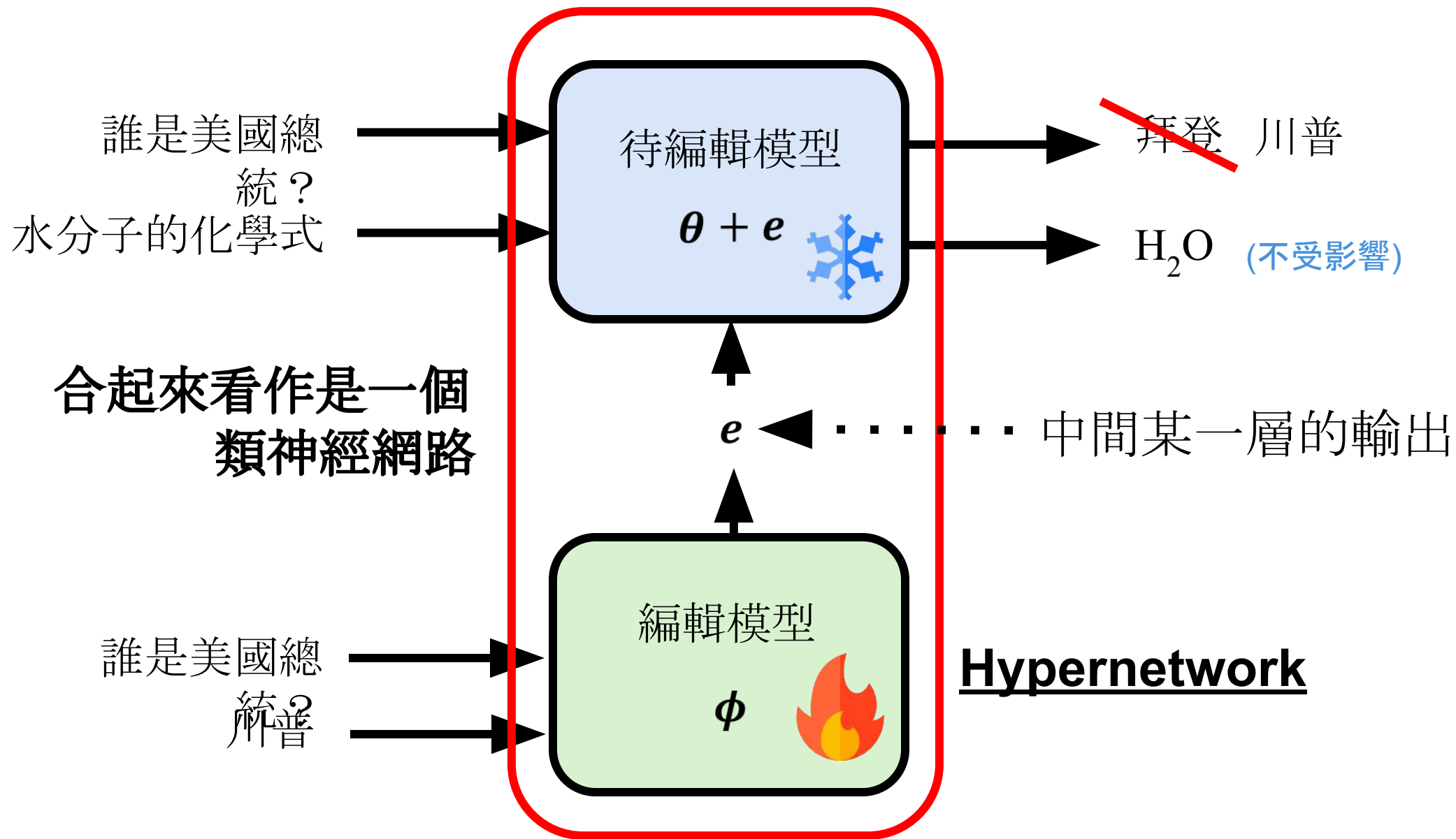
【機器學習 2022】各種奇葩的元學習 (Meta Learning) 用法

[https://youtu.be/QNfymMRUg3M?si=GQP2H\\_pGyqLR6cWI](https://youtu.be/QNfymMRUg3M?si=GQP2H_pGyqLR6cWI)

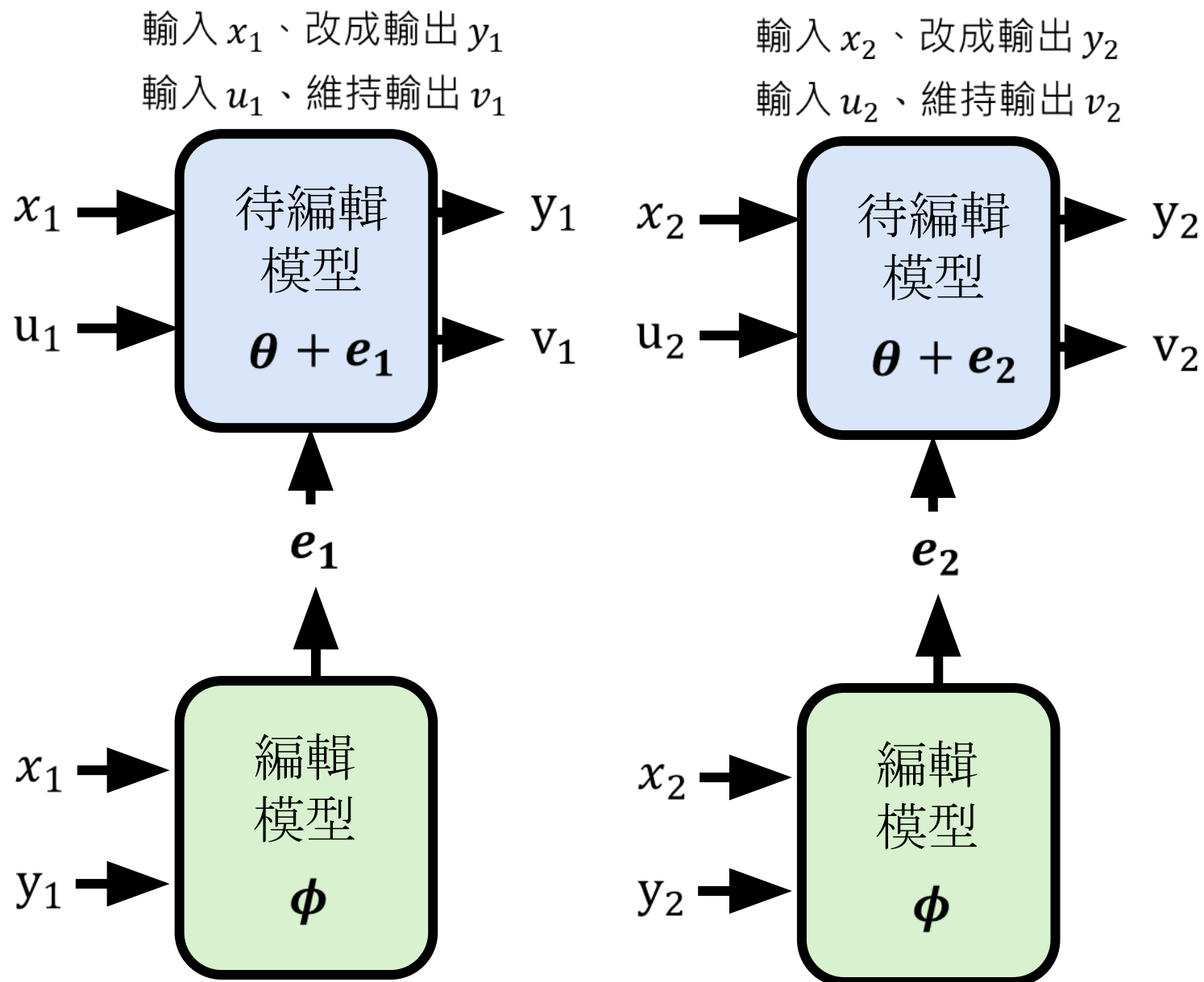


# 如何訓練 Hypernetwork ?

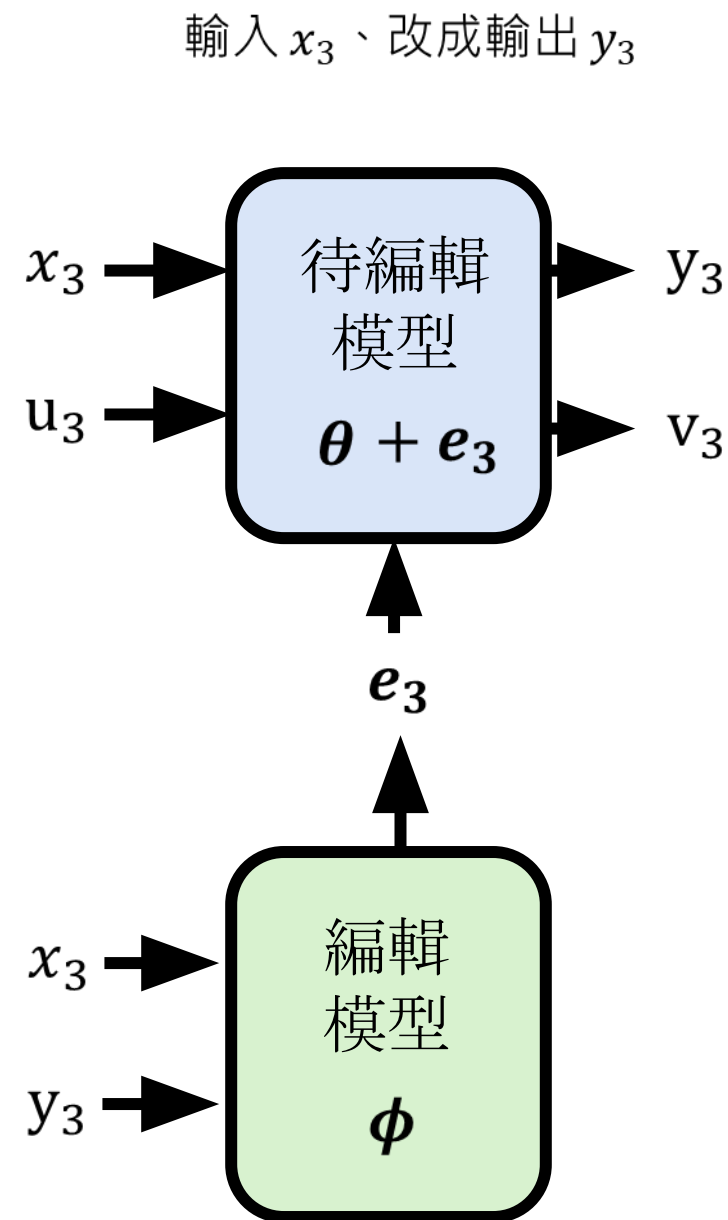


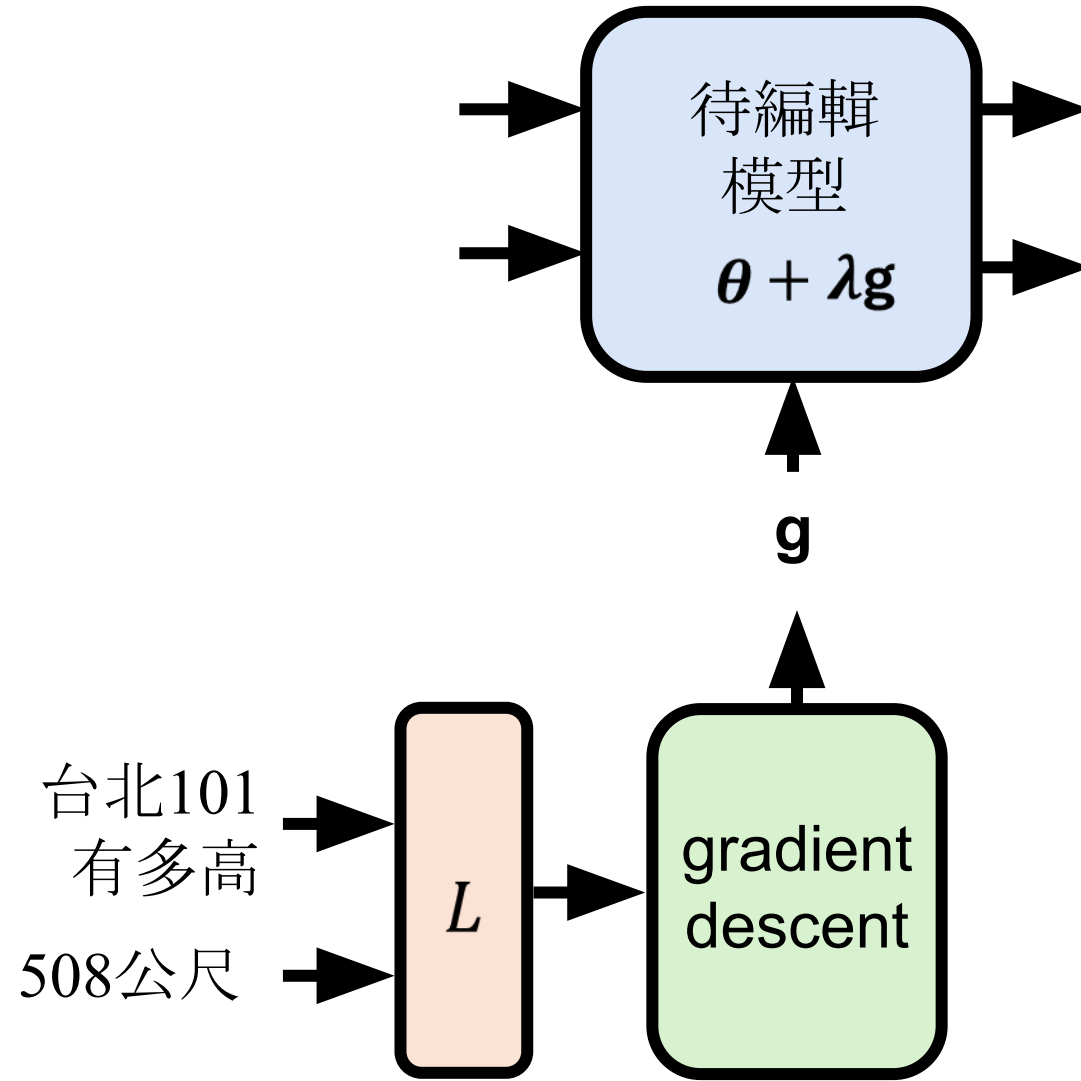
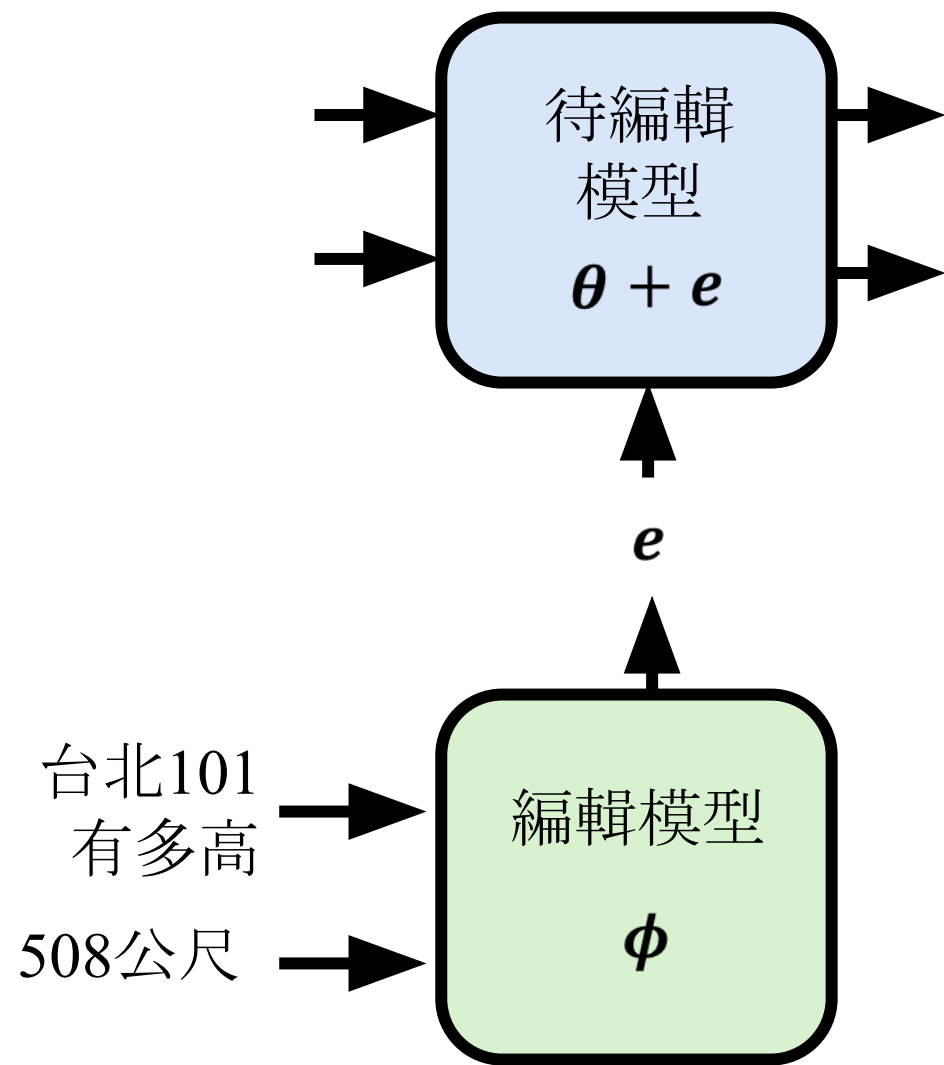


# Training



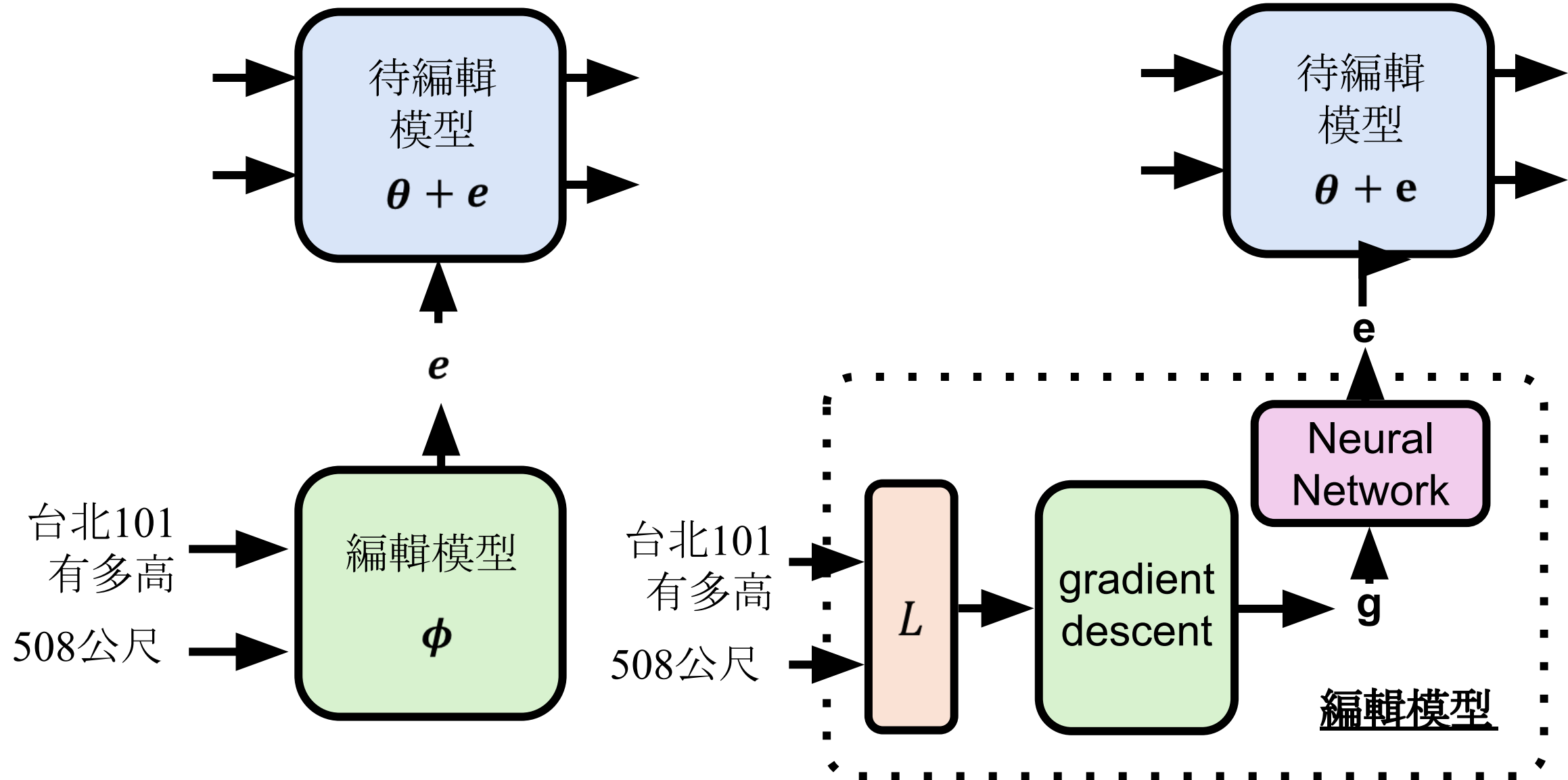
# Testing

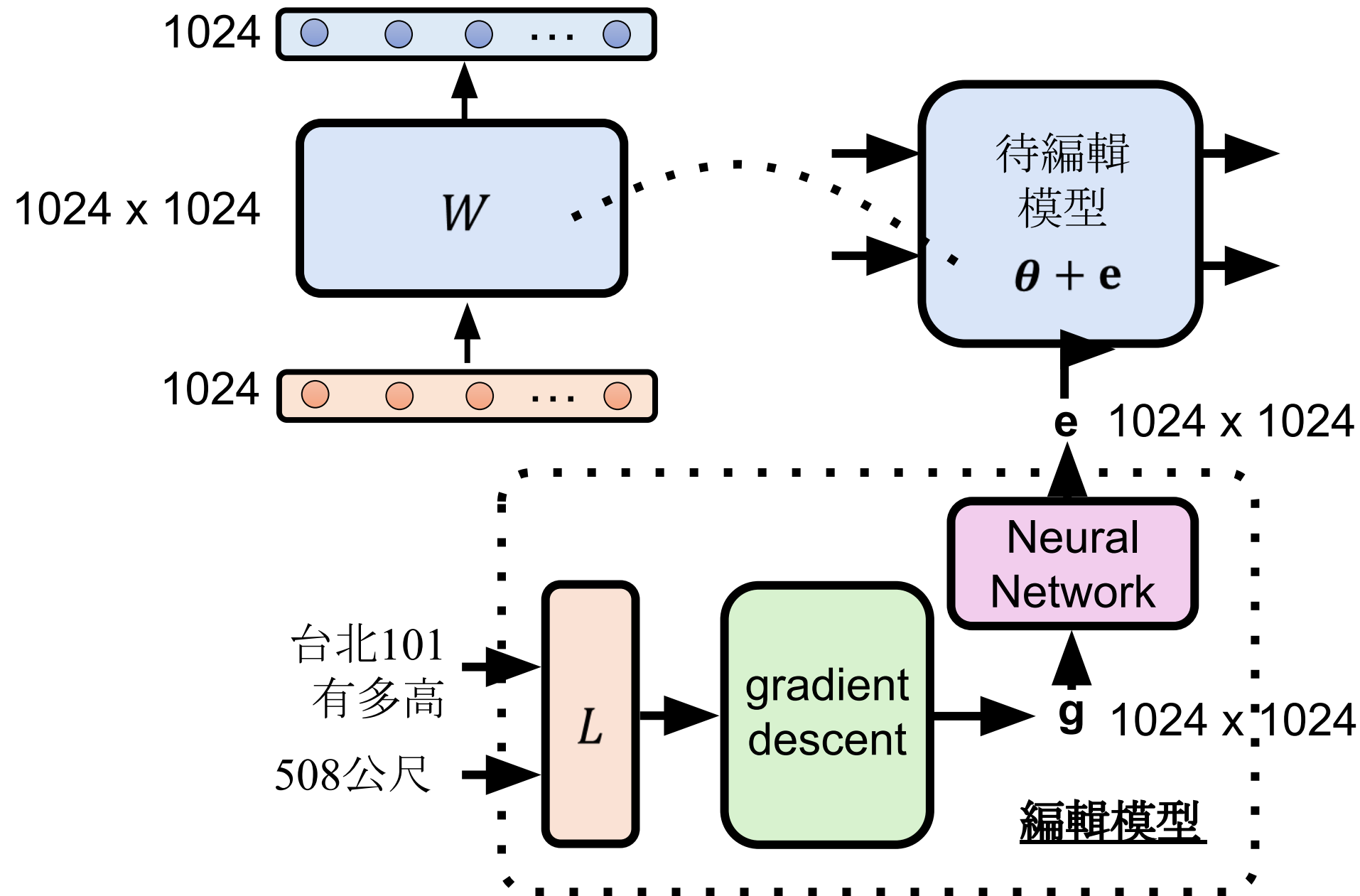




<https://arxiv.org/abs/2104.08164>

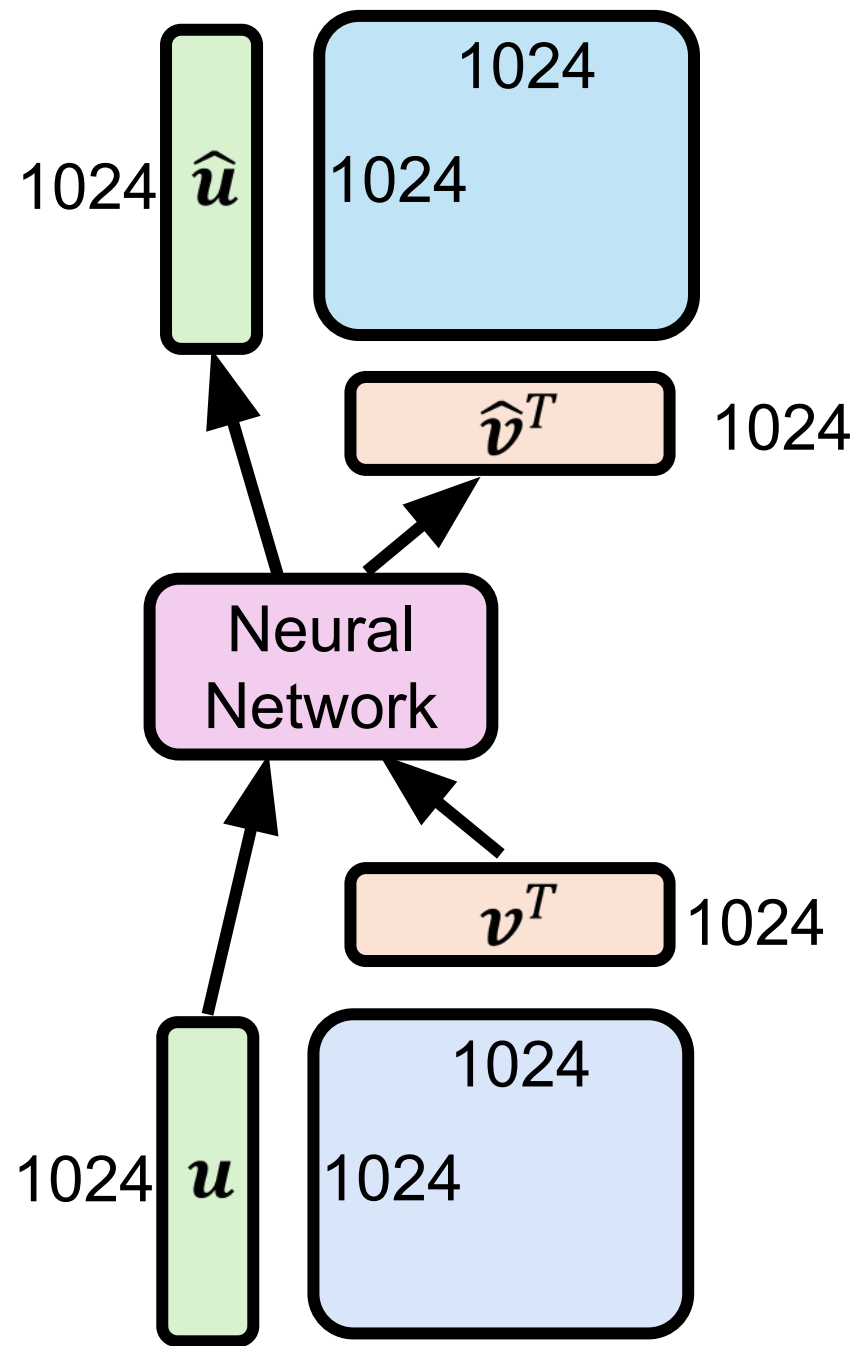
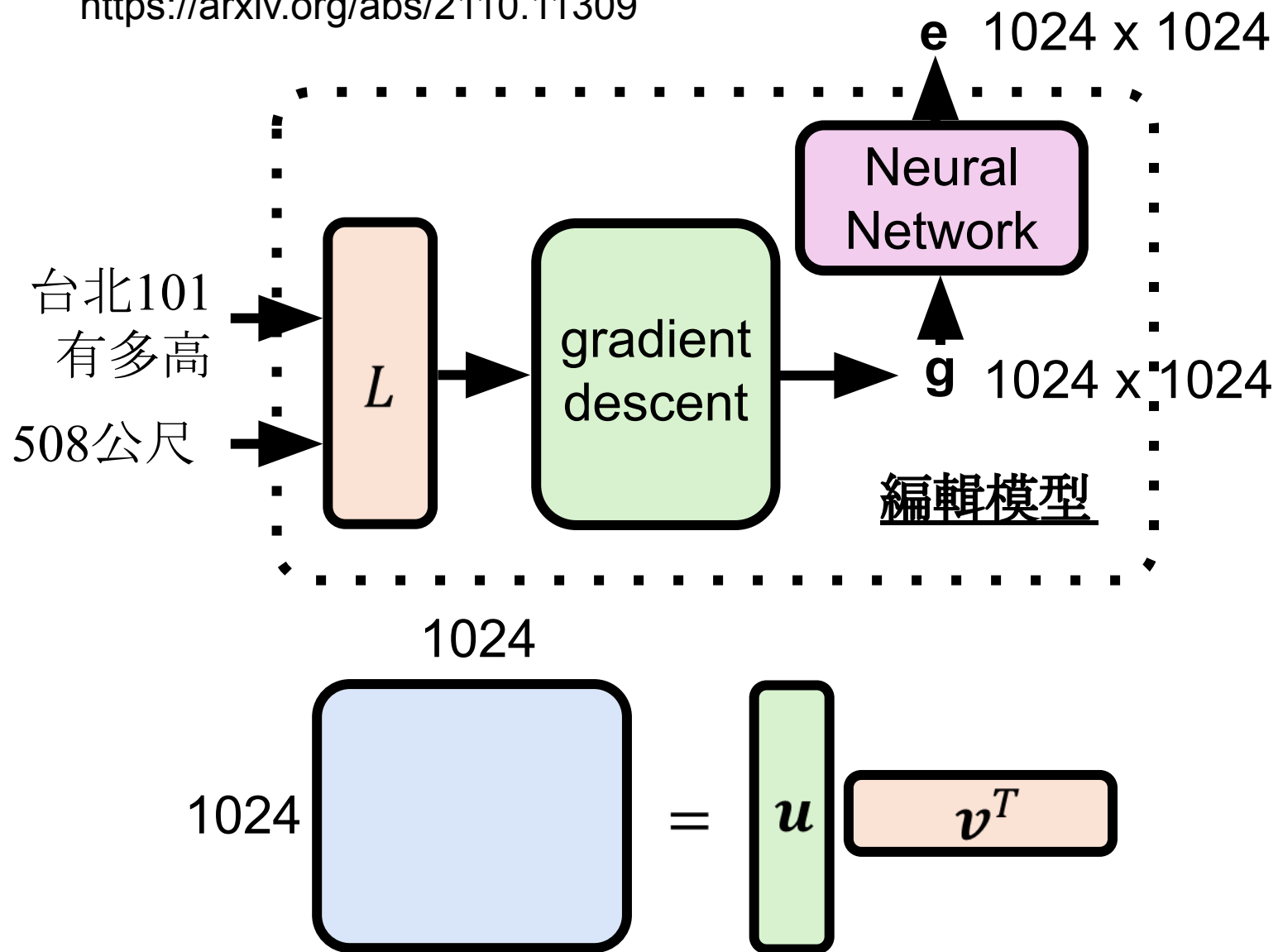
<https://arxiv.org/abs/2110.11309>





MEND

<https://arxiv.org/abs/2110.11309>



$$1024 \times 1024 = u \cdot v^T$$

A diagram illustrating a matrix decomposition. On the left, a light blue rounded rectangle represents a 1024x1024 matrix. To its right is an equals sign, followed by a green rounded rectangle labeled  $u$  and an orange rounded rectangle labeled  $v^T$ .

### Concluding Remarks

$$\frac{\partial C^r}{\partial w_{ij}^l} = \frac{\partial z_i^l}{\partial w_{ij}^l} \frac{\partial C^r}{\partial z_i^l}$$

The diagram shows two layers of a neural network. Layer  $l-1$  has nodes 1, 2, ...,  $j$ , ... in blue circles. Layer  $l$  has nodes 1, 2, ...,  $i$ , ... in orange circles. Weights  $w_{ij}^l$  connect nodes between layers. Arrows indicate the flow of information.

$$\begin{cases} a_j^{l-1} & l > 1 \\ x_j^r & l = 1 \end{cases}$$

**Forward Pass**

$$z^1 = W^1 x^r + b^1$$

$$a^1 = \sigma(z^1)$$

.....

$$z^{l-1} = W^{l-1} a^{l-2} + b^{l-1}$$

$$a^{l-1} = \sigma(z^{l-1})$$

$\delta_i^l$

**Backward Pass**

$$\delta^L = \sigma'(z^L) \bullet \nabla C^r(y^r)$$

$$\delta^{L-1} = \sigma'(z^{L-1}) \bullet (W^L)^T \delta^L$$

.....

$$\delta^l = \sigma'(z^l) \bullet (W^{l+1})^T \delta^{l+1}$$

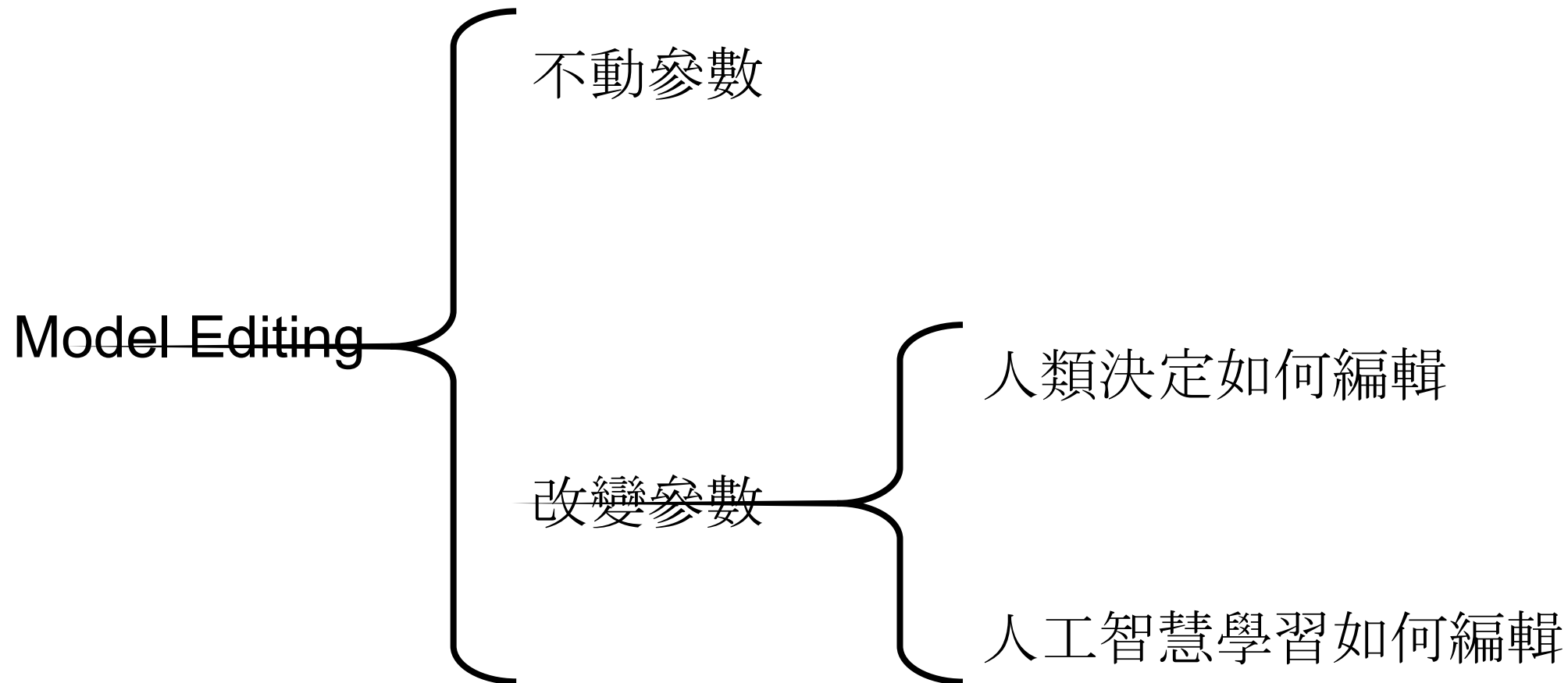
.....

18/18 30:36 / 30:43

[https://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS\\_2015\\_2/Lecture/DNN%20backprop.ecm.mp4/index.html](https://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/DNN%20backprop.ecm.mp4/index.html)



# Model Editing 常見方法



# To Learn More ...



## KnowEdit

A Comprehensive Study of Knowledge Editing for Large Language Models

Ninyu Zhang<sup>\*1</sup>, Yunzhi Yao<sup>\*1</sup>, Bozhong Tian<sup>\*1</sup>, Peng Wang<sup>\*1</sup>,  
Shumin Deng<sup>\*2</sup>, Mengru Wang<sup>1</sup>, Zekun Xi<sup>1</sup>, Shengyu Mao<sup>1</sup>, Jintian Zhang<sup>1</sup>, Yuansheng Ni<sup>1</sup>, Siyuan Cheng<sup>1</sup>,  
Ziwen Xu<sup>1</sup>, Xin Xu<sup>1</sup>, Jia-Chen Gu<sup>1</sup>, Yong Jiang<sup>1</sup>, Pengjun Xie<sup>1</sup>, Fei Huang<sup>1</sup>, Lei Liang<sup>1</sup>, Zhiqiang Zhang<sup>1</sup>,  
Xiaowei Zhu<sup>1</sup>, Jun Zhou<sup>1</sup>, Huajun Chen<sup>†1</sup>

<sup>1</sup>Zhejiang University, <sup>2</sup>National University of Singapore, <sup>3</sup>Univers of California, Los Angeles, <sup>4</sup>Ant Group  
<sup>5</sup>Alibaba Group

<https://zjunlp.github.io/project/KnowEdit/>