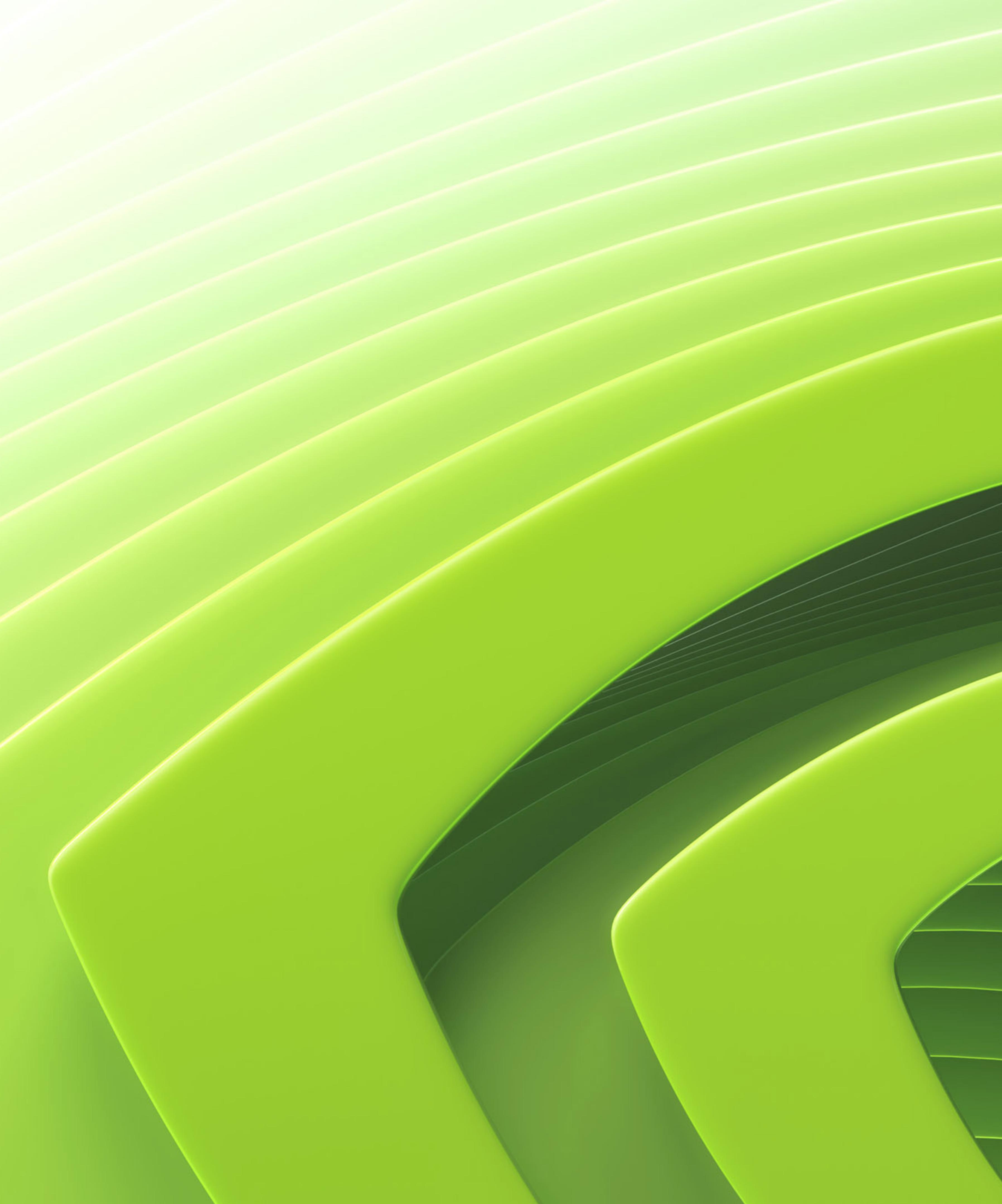




NCHC End-to-end LLM Bootcamp (Day 1)

Cliff Chiu, Solution Architect | Aug 06, 2024



Day 1: Tuesday, August 6, 2024 // 09:00 AM - 13:00 PM (Online)

09:00 AM - 09:10 AM: Welcome to End to End LLM Bootcamp

09:10 AM - 10:10 AM: Introduction of Generative AI and LLMs

10:10 AM - 10:20 AM: Break

10:20 AM - 11:20 AM: Case Study: (1) ChipNeMo, (2) Nemotron-4-340B

11:20 AM - 11:30 AM: Break

11:30 AM - 12:20 PM: LLM Inference Theory

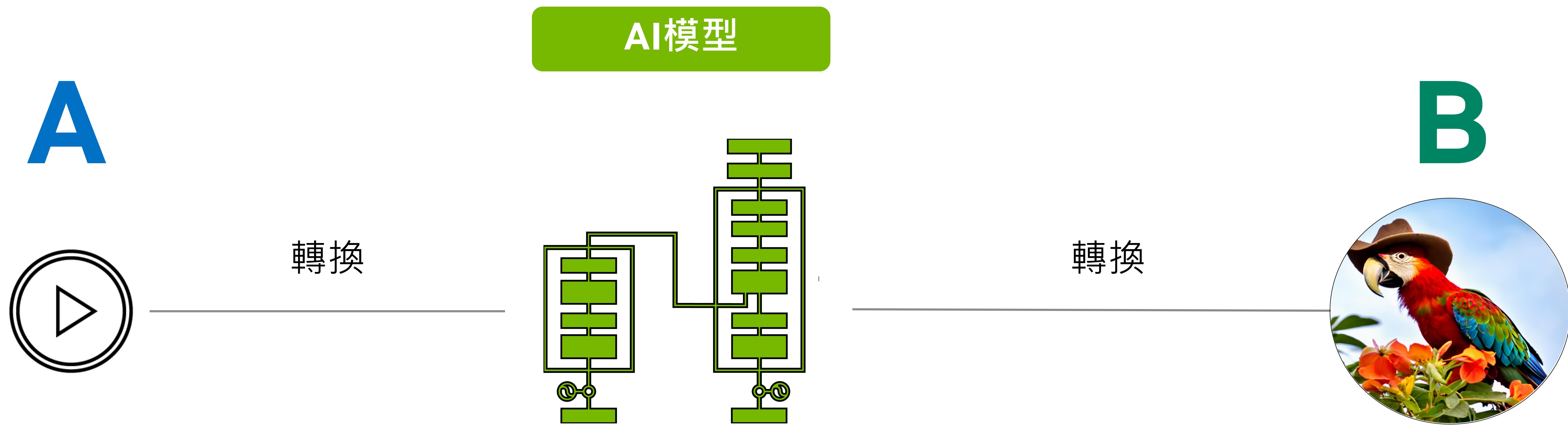
12:20 AM - 12:30 AM: Break

12:30 AM - 12:50 PM: Introduction of Networking techniques for AI

12:50 PM - 01:00 PM: Q&A

什麼是生成式AI (Generative AI) ?

AI模型：A轉換成B



什麼是生成式AI (Generative AI) ?

轉換各種類型的資料來產生新的內容

A

文字



聲音



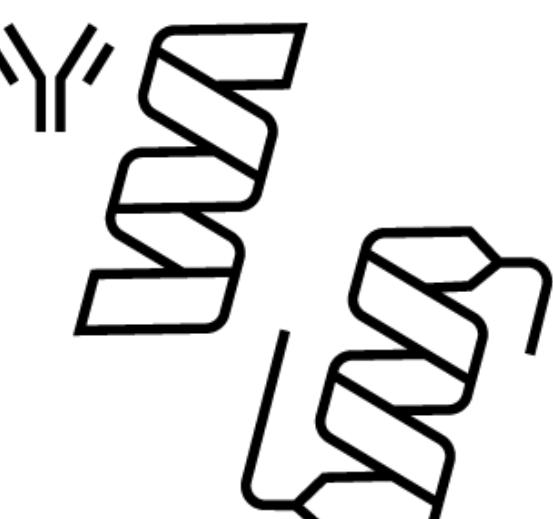
影像



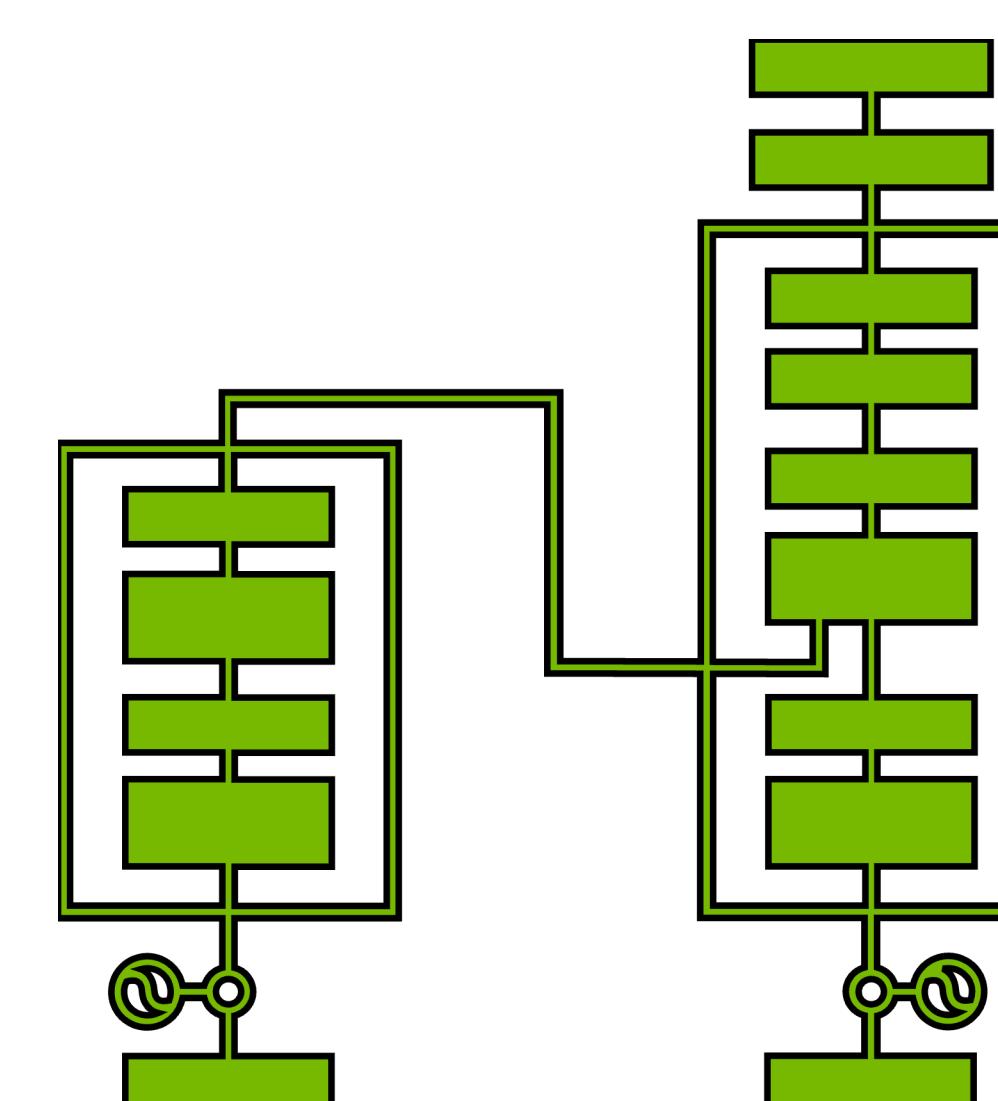
3D物件



生物資訊
(DNA)

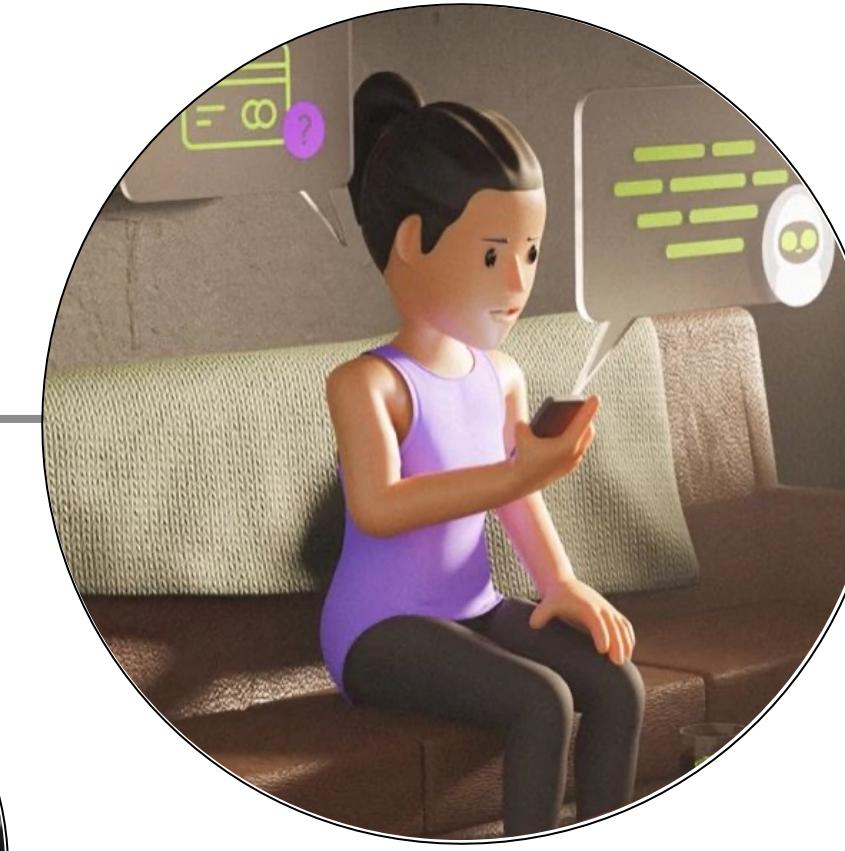


生成式AI模型

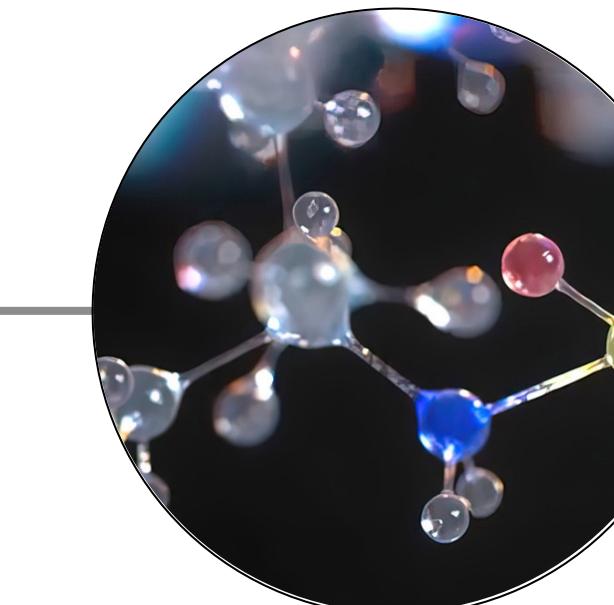


B

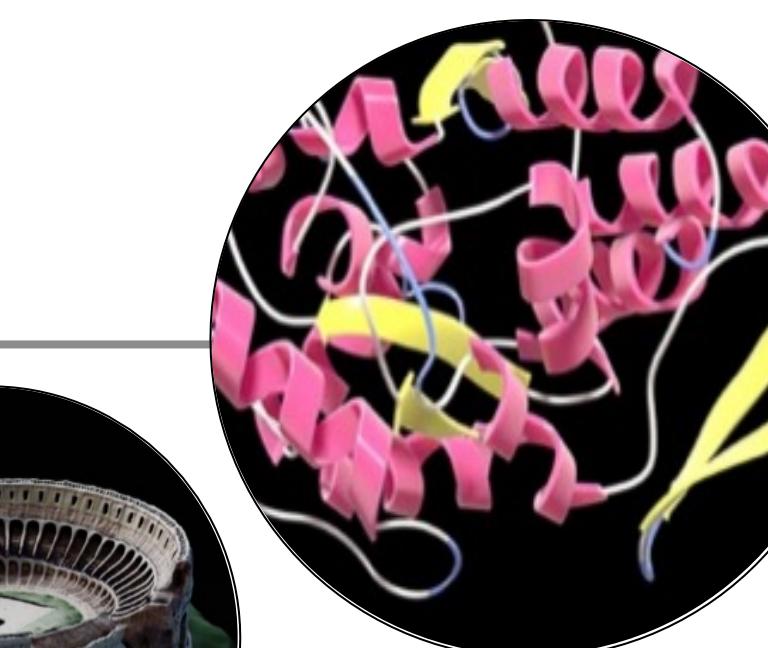
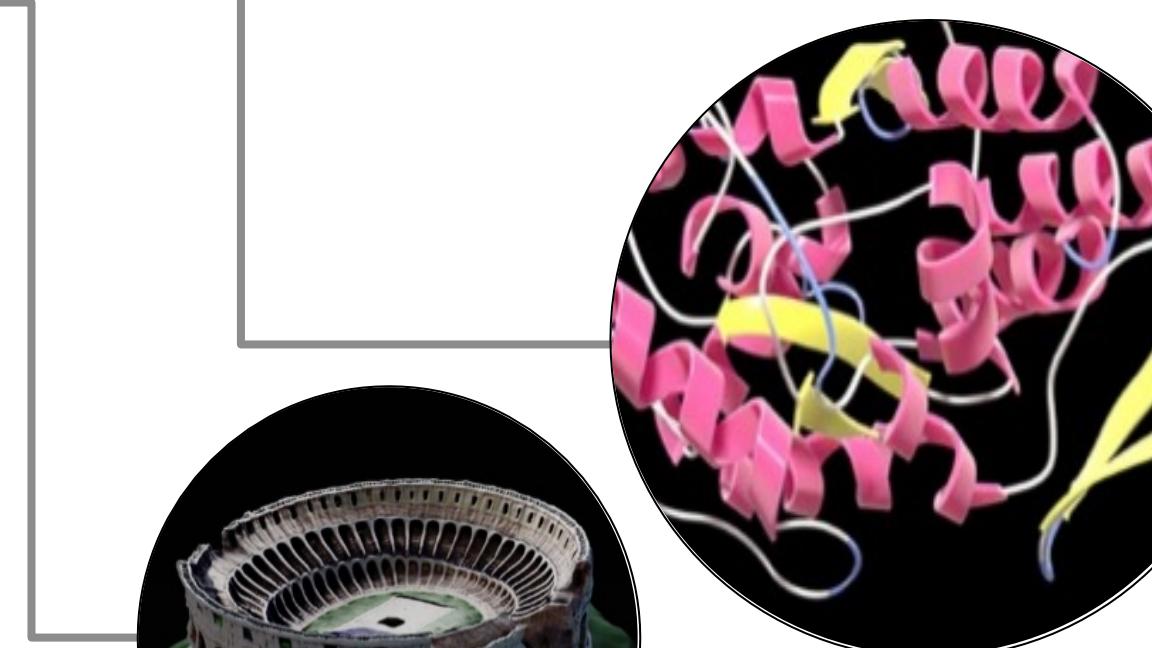
臉部表情



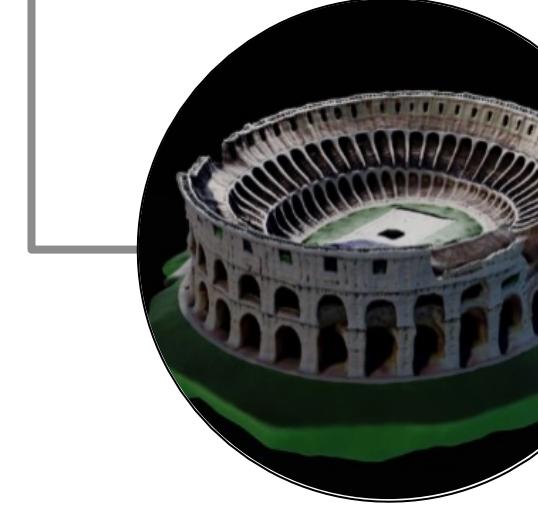
聲音



影像



生物資訊



clearly describe quantum computing.
Quantum computing is a new type of computing that uses quantum mechanics to perform calculations and solve problems much faster than classical computers. It is based on the ability of particles, such as photons, to exist in multiple states at the same time through superposition.
In classical computing, information is represented by a value of either 0 or 1. In quantum computing, information is represented by a superposition of 0 and 1, allowing for many more possible states and faster calculations.

文字

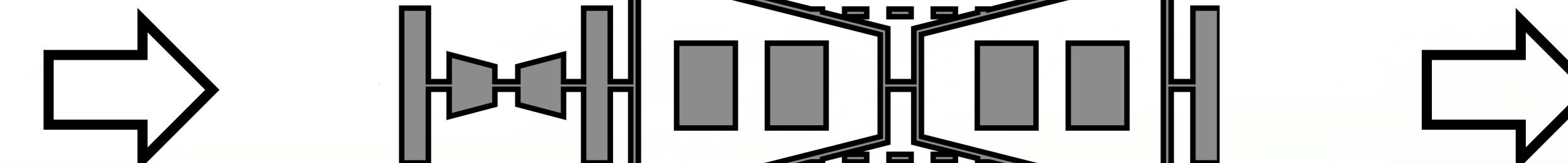
Generative AI vs Discriminative AI

輸入



生成式AI vs 判別式AI

判別式AI



判斷類別（選擇題）

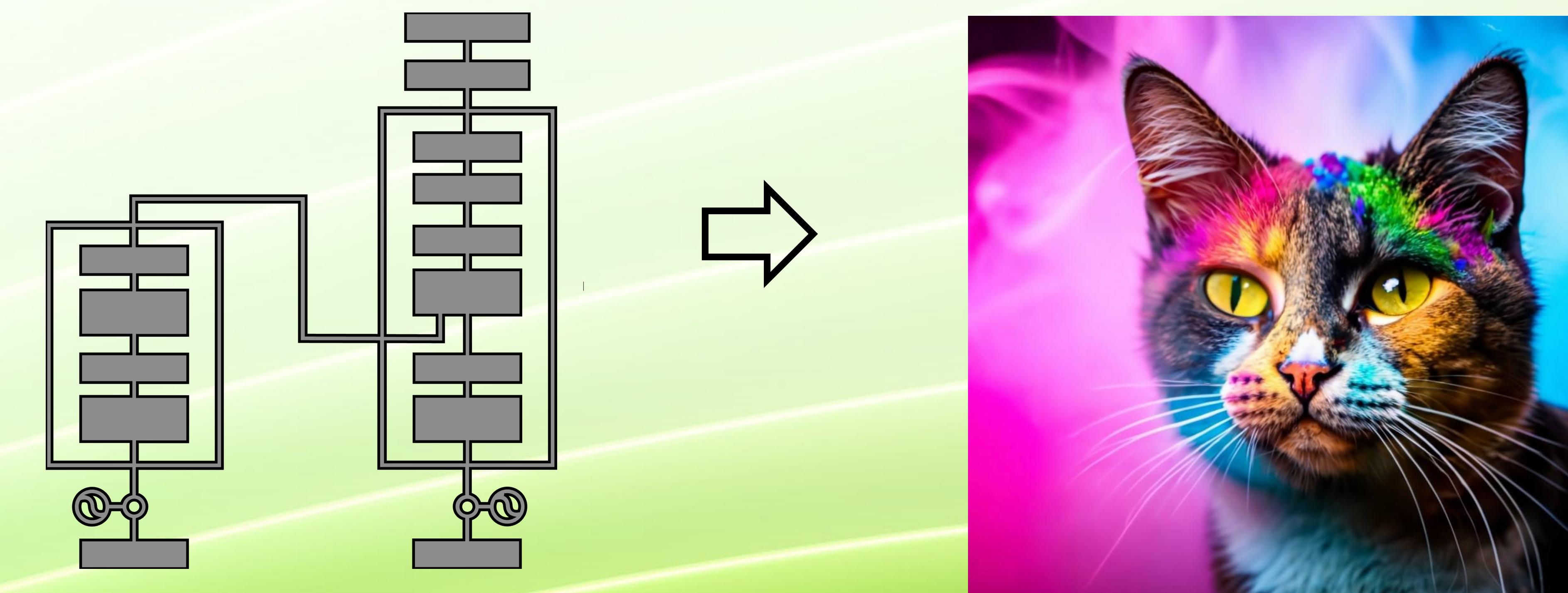
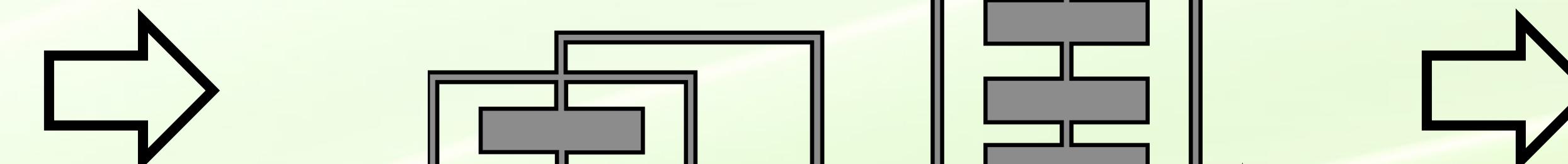
0.2	0.9	0.6	0.1
-----	-----	-----	-----

狗 貓 虎 人

生成式AI

產生具有真實意義的內容（作文題、問答題）

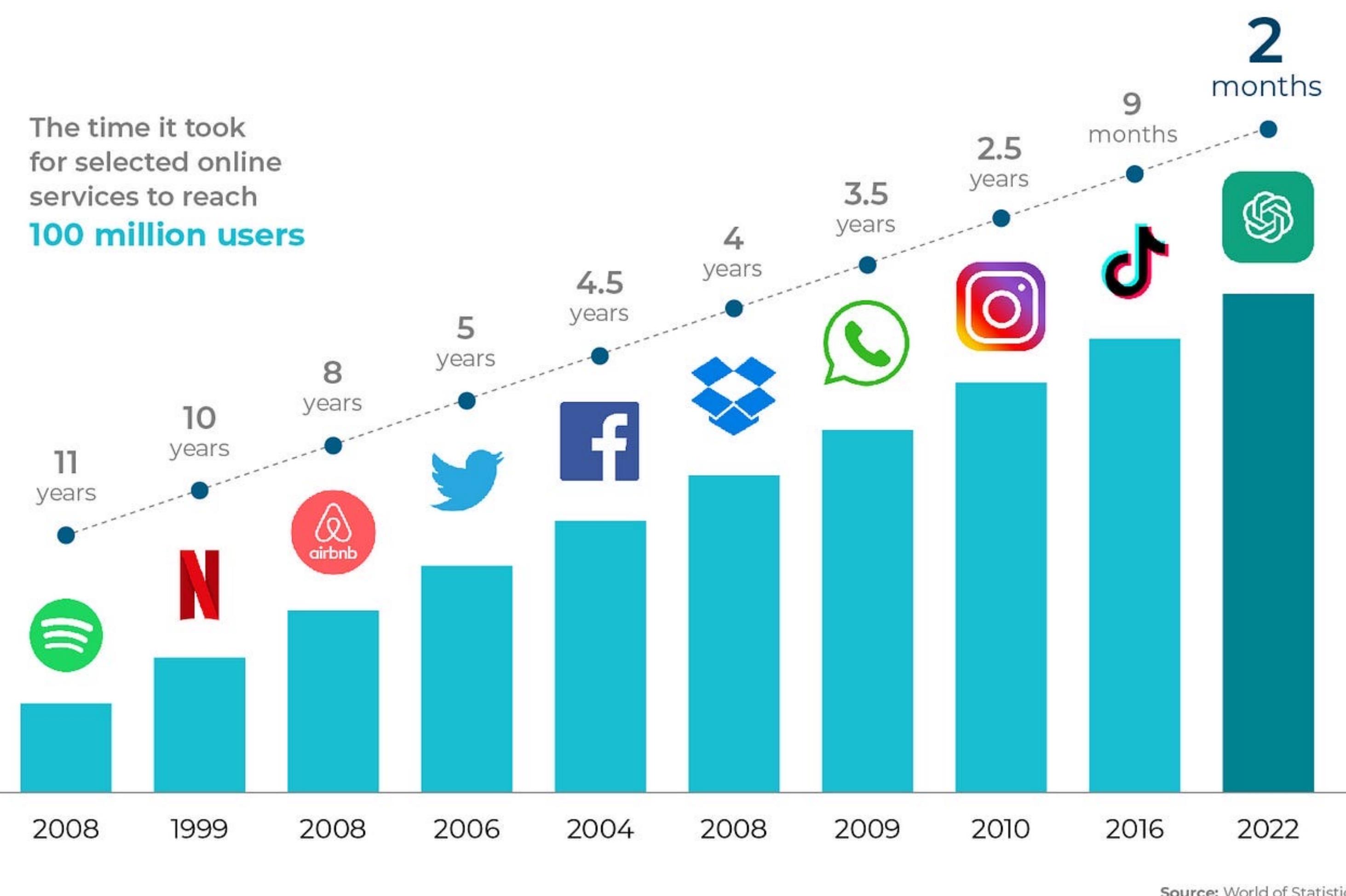
請你幫我畫一張可愛的貓咪照片，貓咪的臉上噴灑了許多不同的顏色背景中有彩色的煙霧。



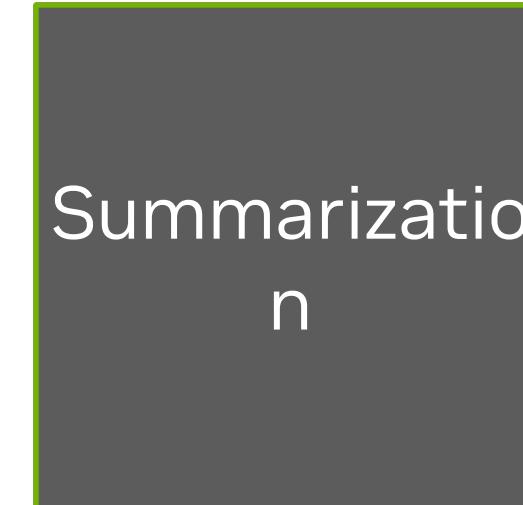
生成式AI模型可以哪些事情？



Chat-GPT sprints to 100 million users



寫公文、產生會議紀錄

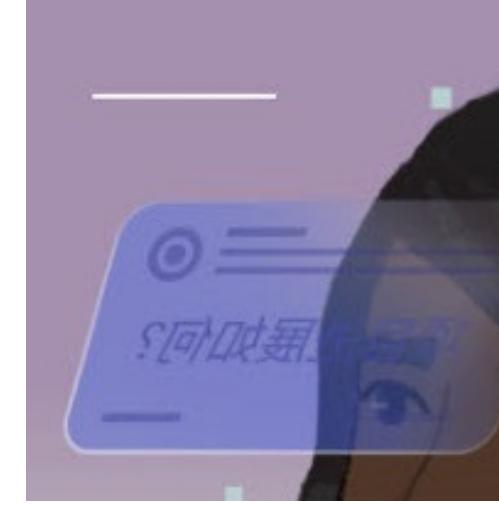


GPT-4

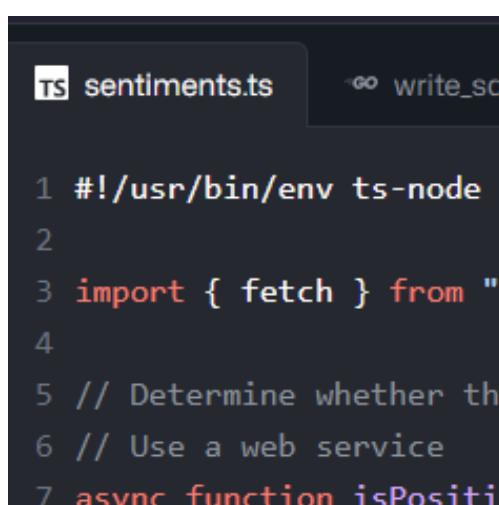


Marketing Copy

文字翻譯、文法修正



寫程式、分析問題



Dynamic Code Commenting



Code Llama



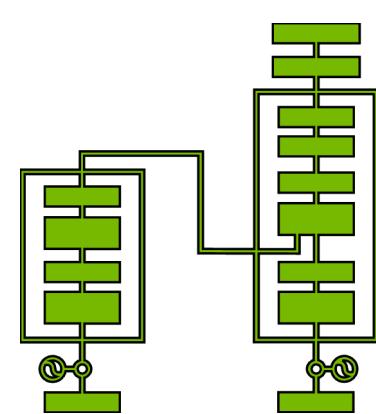
Function Generation

生成影像、3D物件



ChatGPT是怎麼運作的？

文字接龍

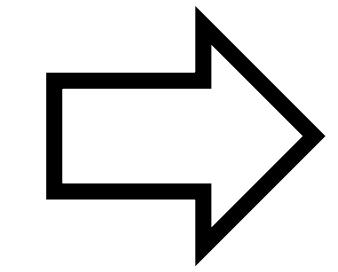


語言模型經由大量、各種不同人類語言的文本訓練而成

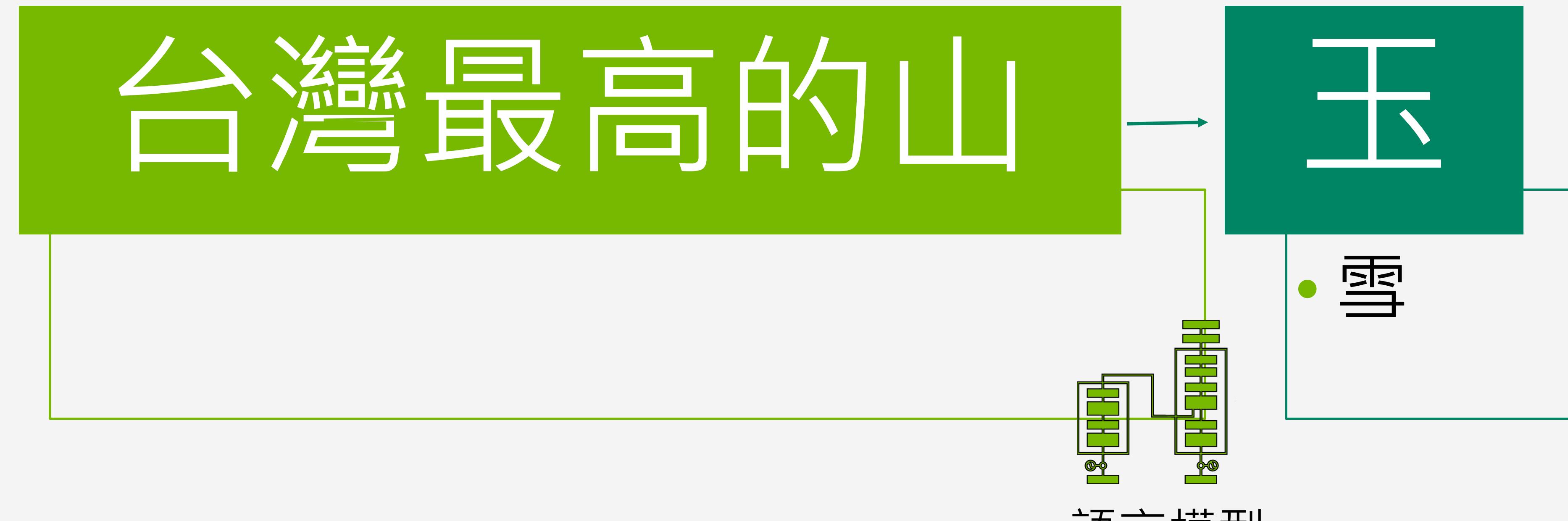
語言模型是怎麼回答問題的？

文字接龍

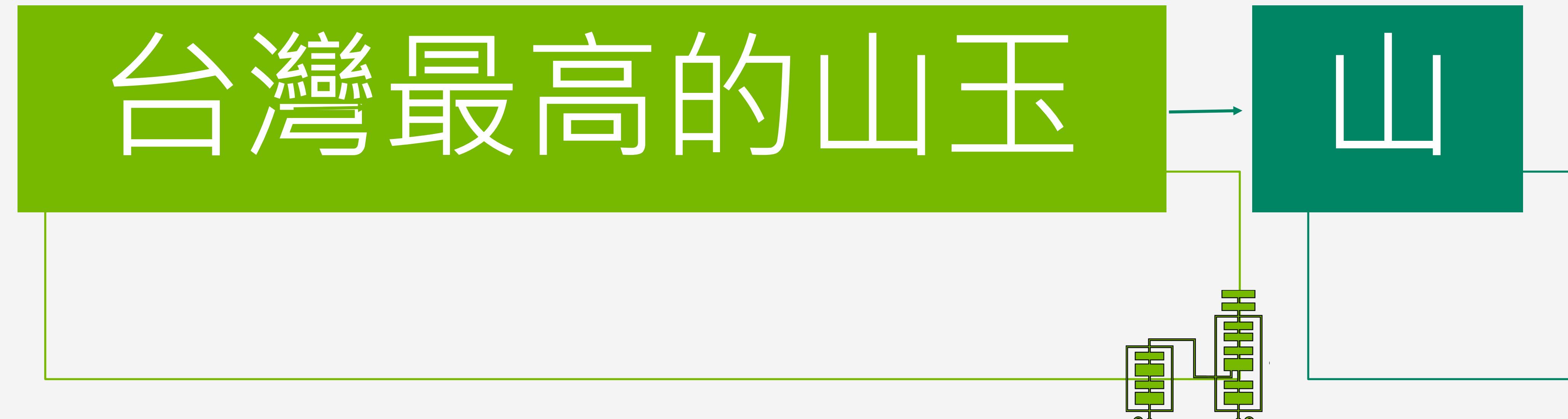
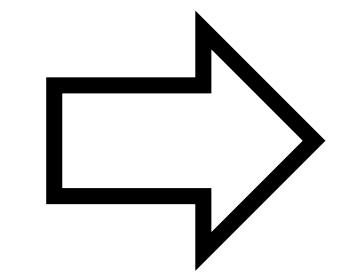
問題



語言模型的運作原理：預測下一個字



問題

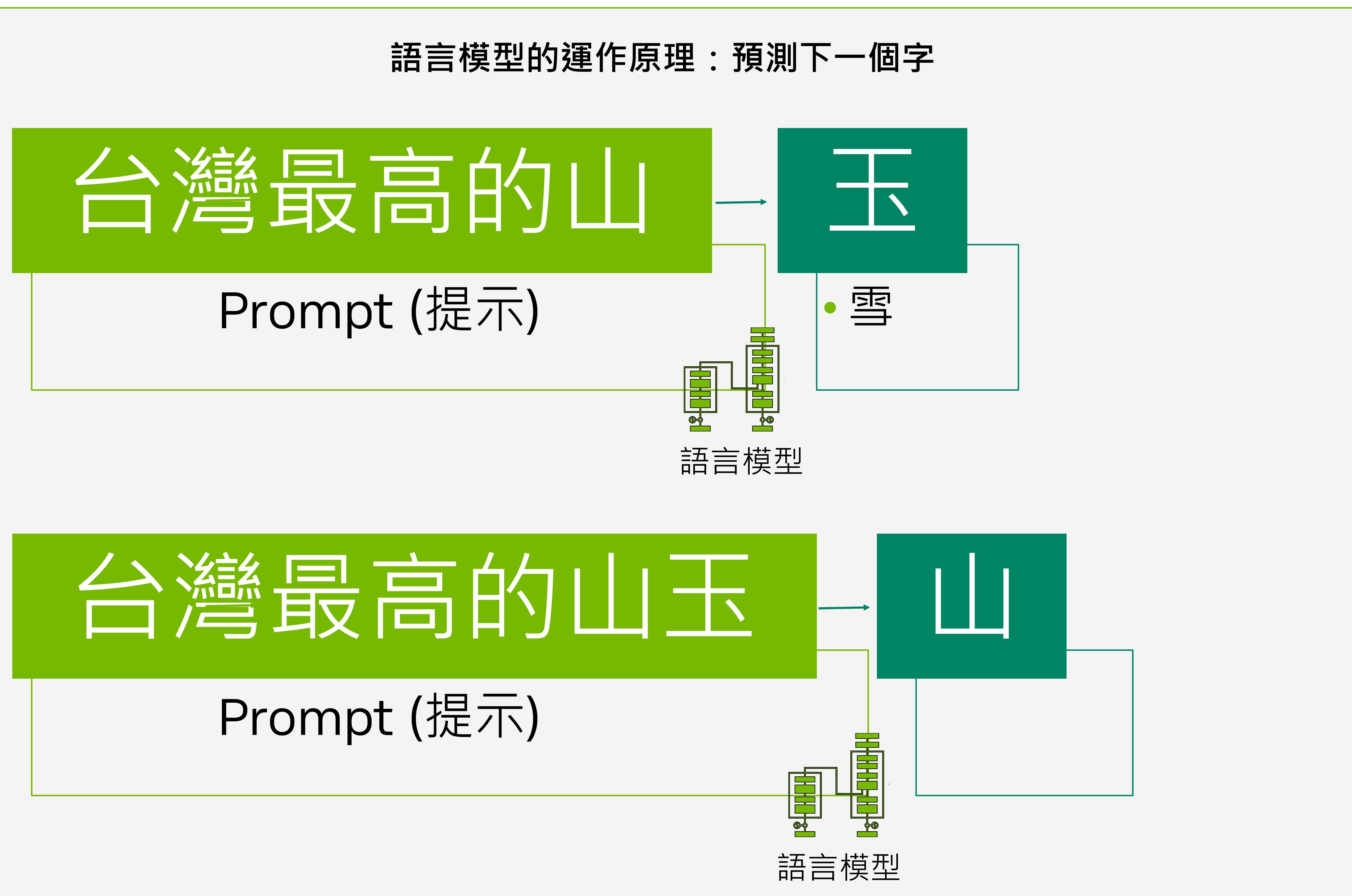


NVIDIA

語言模型是怎麼回答問題的？

文字接龍

問題 →



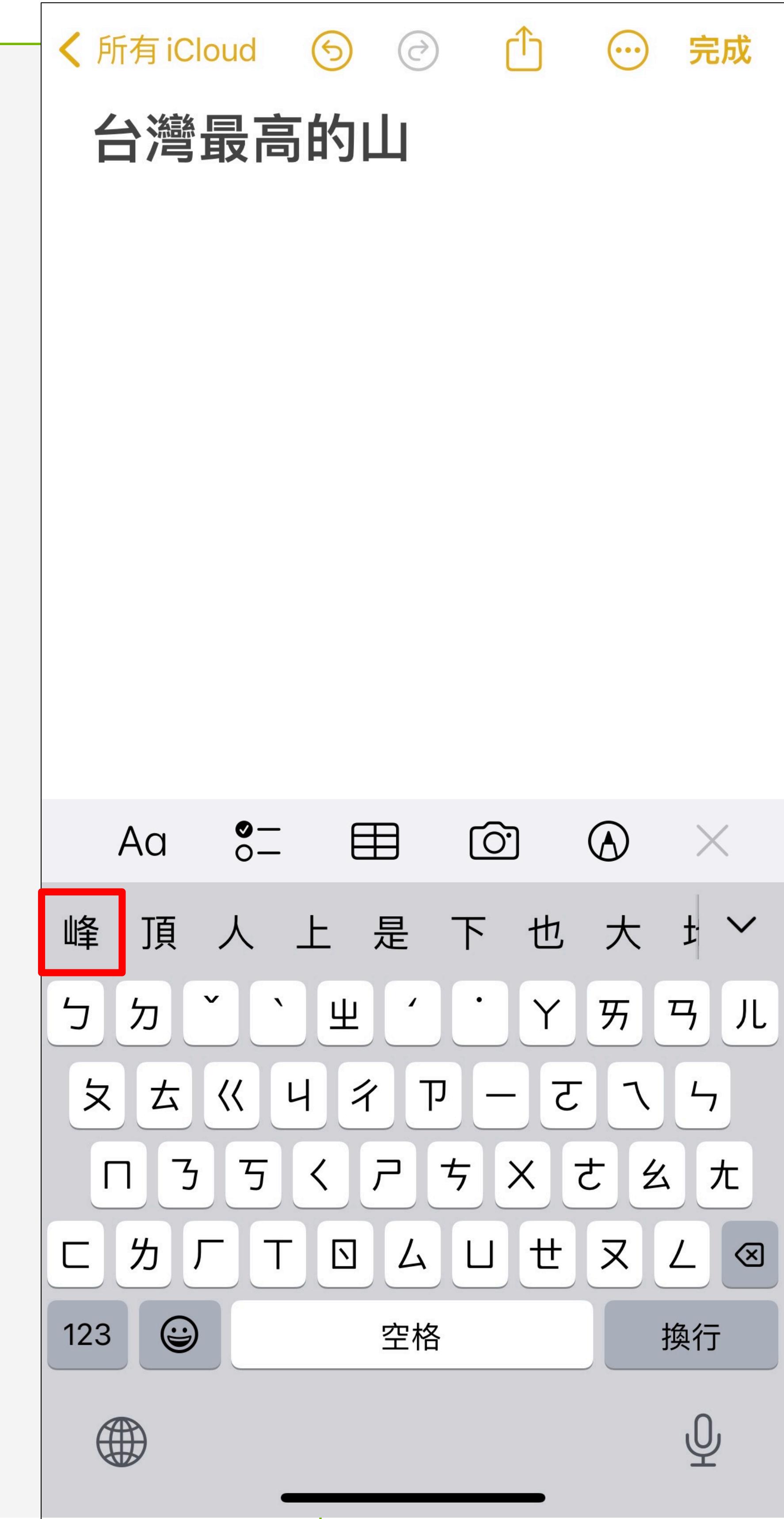
手機輸入法聯想詞

文字接龍

問題 →



問題 →



什麼是大型語言模型？

一種複雜的深度學習模型

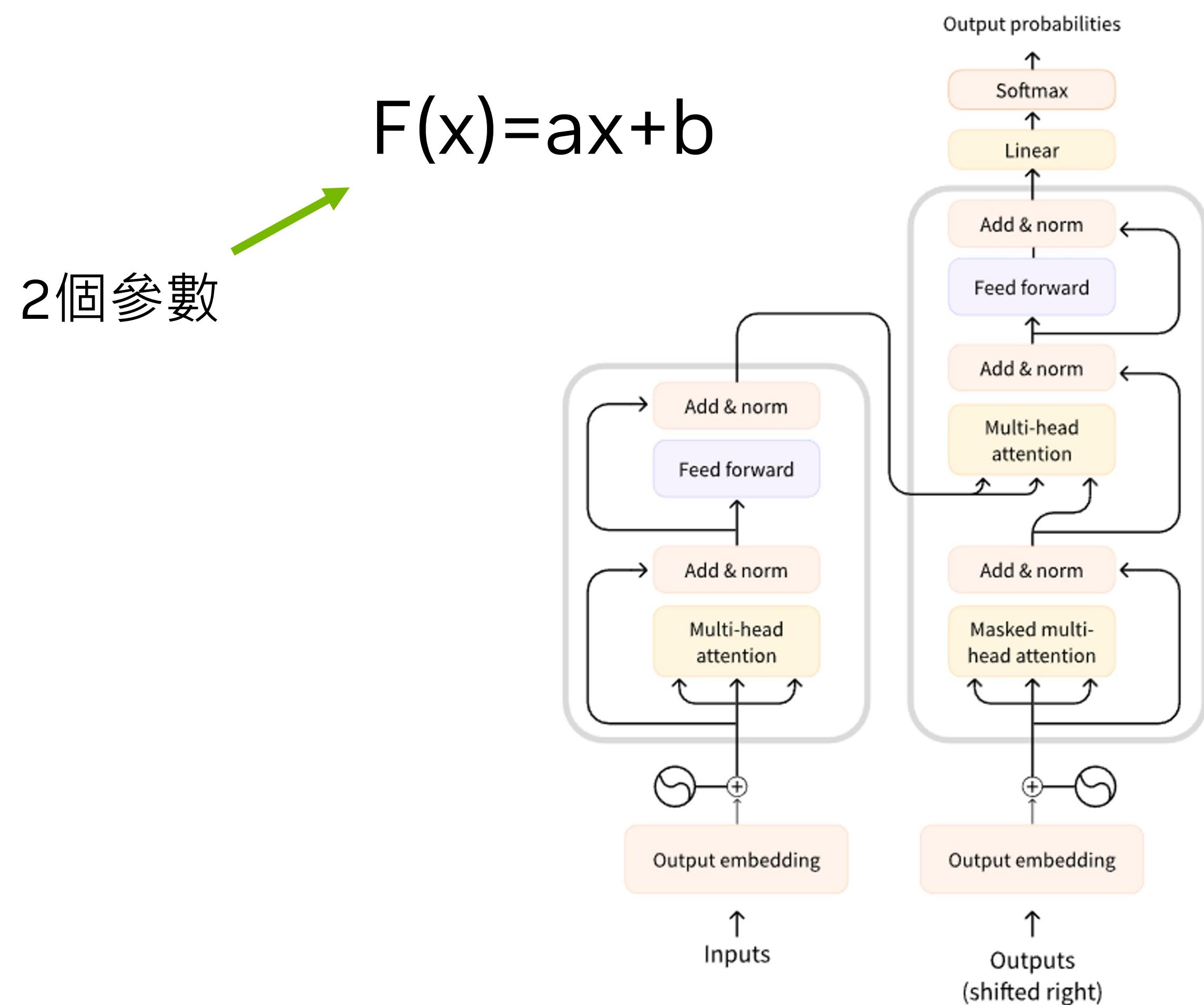
- 大型語言模型是一種有大量參數，並以巨量文本訓練而成的深度學習模型，能學習語言結構、語義及上下文，廣泛應用於自然語言處理，如語言生成、翻譯、摘要等，提高文本處理的效率和準確性。

大量的模型參數

: 先天的腦容量

巨量的文本資料

: 後天閱讀的知識量



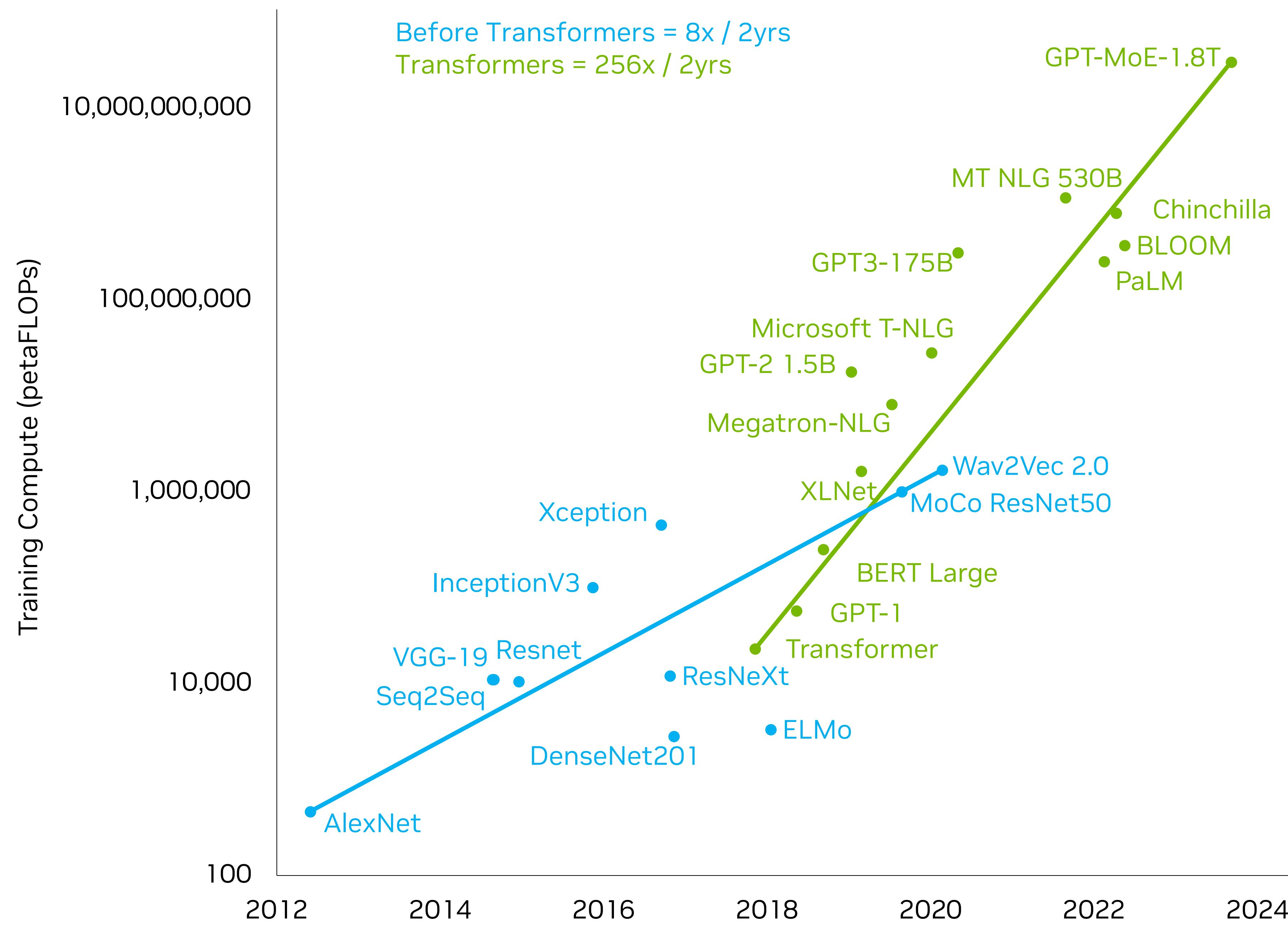
2個參數

$$F(x)=ax+b$$

Model	# Parameters	Dataset	# GPUs	Time
GPT-3	175B	500B tokens	1024 x A100	34 Days
LLaMA1	65B	1T tokens	2048 x A100	21 Days
MPT	30B	1T tokens	512 x A100	28 Days
	7B	1T tokens	440 x A100	9 Days
Falcon	40B	1T tokens	384 x A100	60 Days

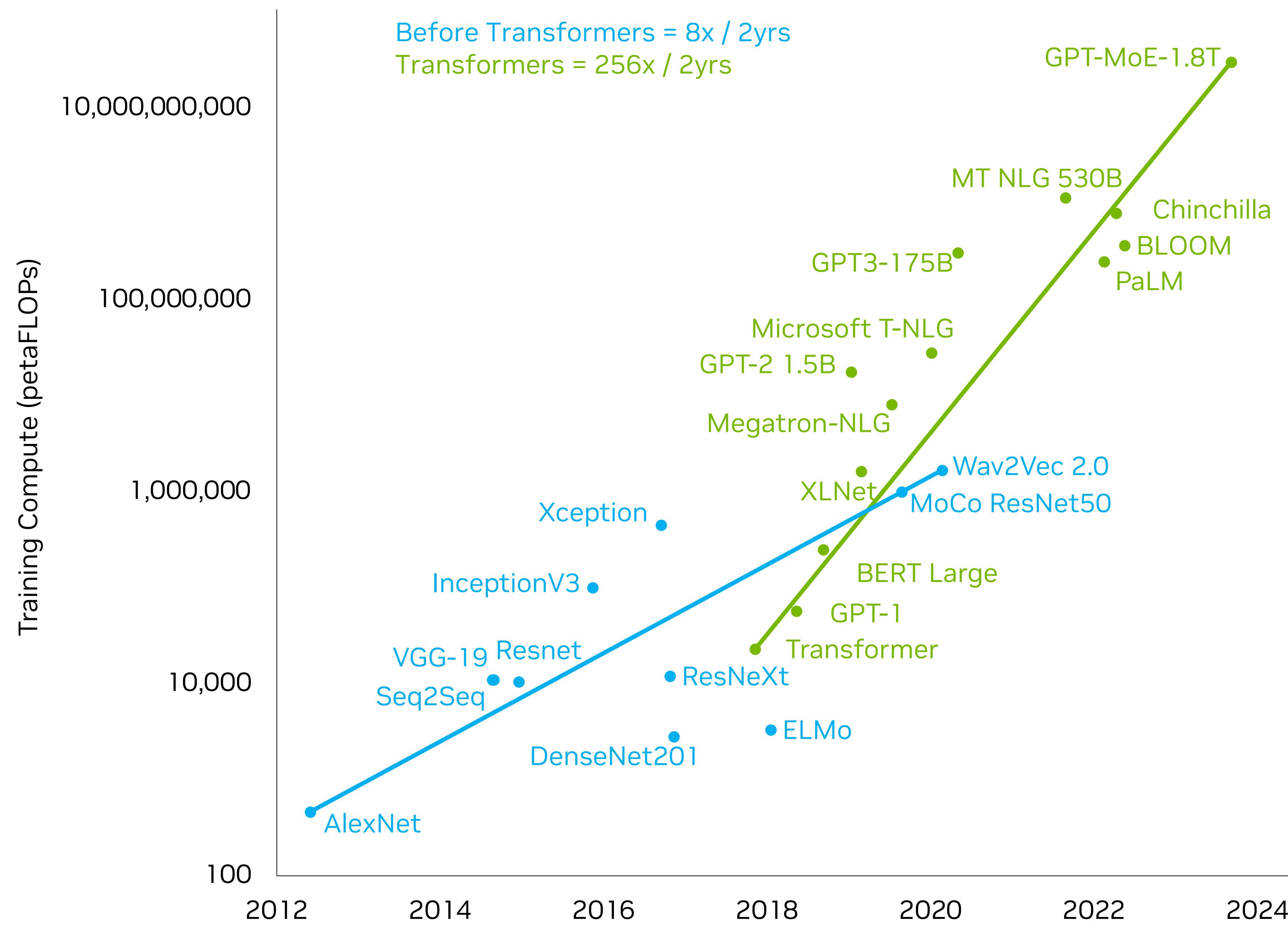
Transformer

Explosive Growth in AI Computational Requirements



	Time (GPU hours)	Power Consumption (W)
Llama 3 8B	1.3M	700
Llama 3 70B	6.4M	700
Total	7.7M	

Explosive Growth in AI Computational Requirements



	Time (GPU hours)	Power Consumption (W)
Llama 3 8B	1.3M	700
Llama 3 70B	6.4M	700
Total	7.7M	

If using 2048 x GPUs:

Llama 3 8B: $1.3M \div 2048 \div 24 \cong 27 \text{ days}$

Llama 3 70B: $6.4M \div 2048 \div 24 \cong 130 \text{ days}$

If using 10000 x GPUs:

Llama 3 8B: $1.3M \div 10000 \div 24 \cong 6 \text{ days}$

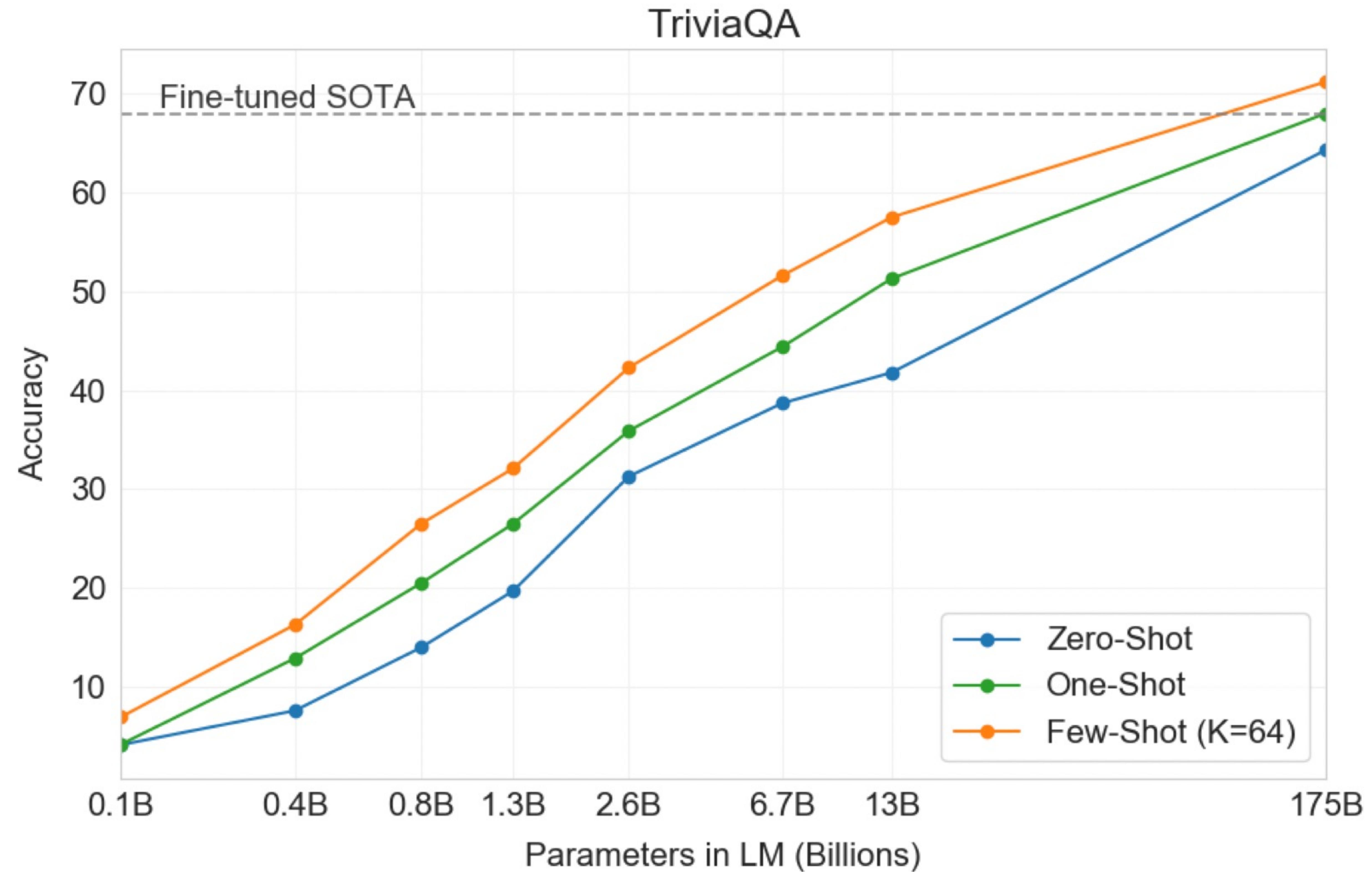
Llama 3 70B: $6.4M \div 2048 \div 24 \cong 27 \text{ days}$

On TriviaQA, GPT-3's performance

當語言模型參數量大到效果足以媲美專為特定任務微調的小模型

研究人員發現**越大的模型與越多的資料**
對語言模型的表現越好。

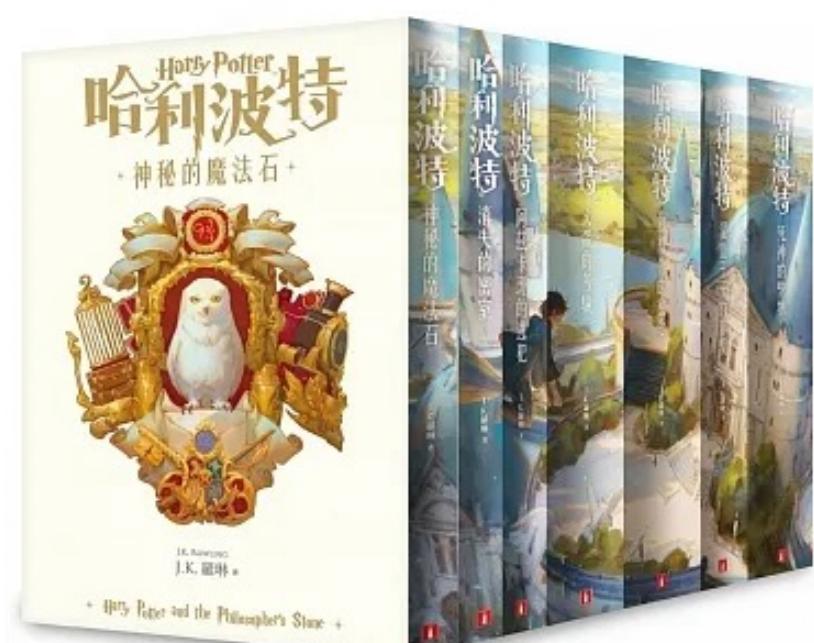
使得大型語言模型能夠更精準學習複雜
的語言模式。



大型語言模型到底看過多少的資料？
以哈利波特全集作為基準

大型語言模型訓練資料集

哈利波特全集



200萬個字元

V.S



2兆個字元以上



至少看了100萬遍的哈利波利全集

Large Language Model Journey

	Methods	Models	Infrastructure
Stage 1. Learning knowledge	 Pre-training	<ul style="list-style-type: none">GPT-3LLaMAMistral / MixtralFalconNemotron	GPU SuperPods

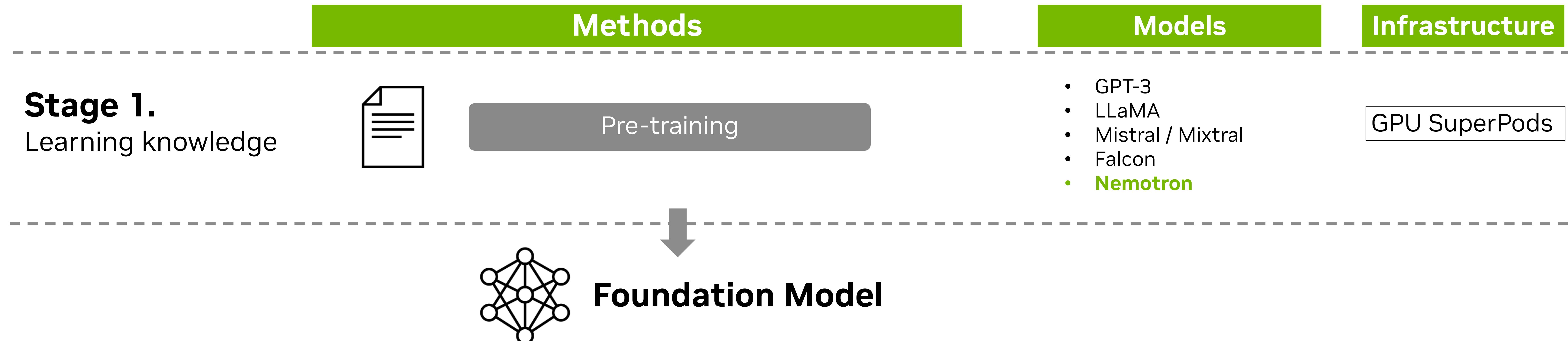
Training Data

原始語料：世界第一高峰是喜馬拉雅山

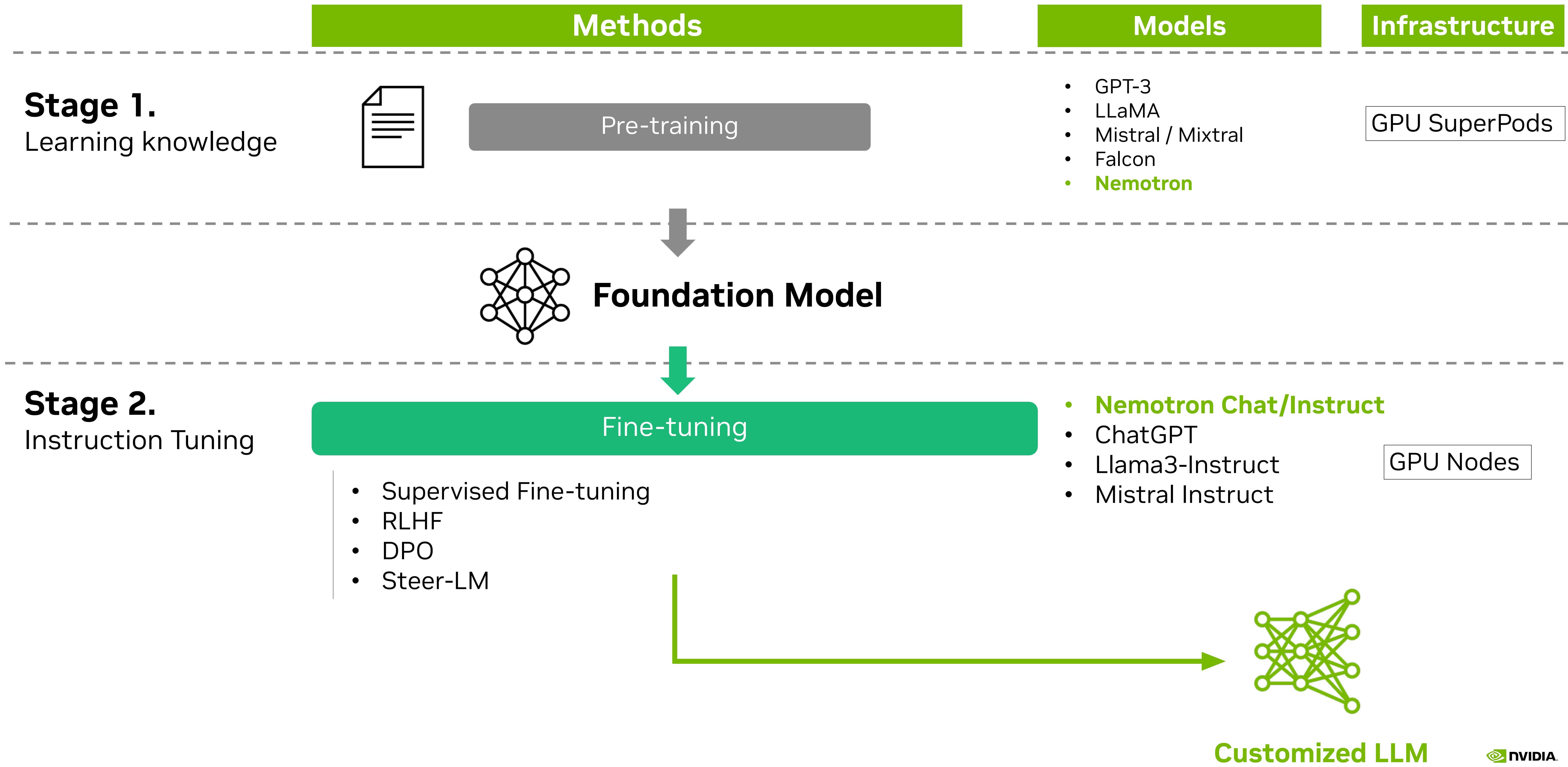
Unsupervised Learning



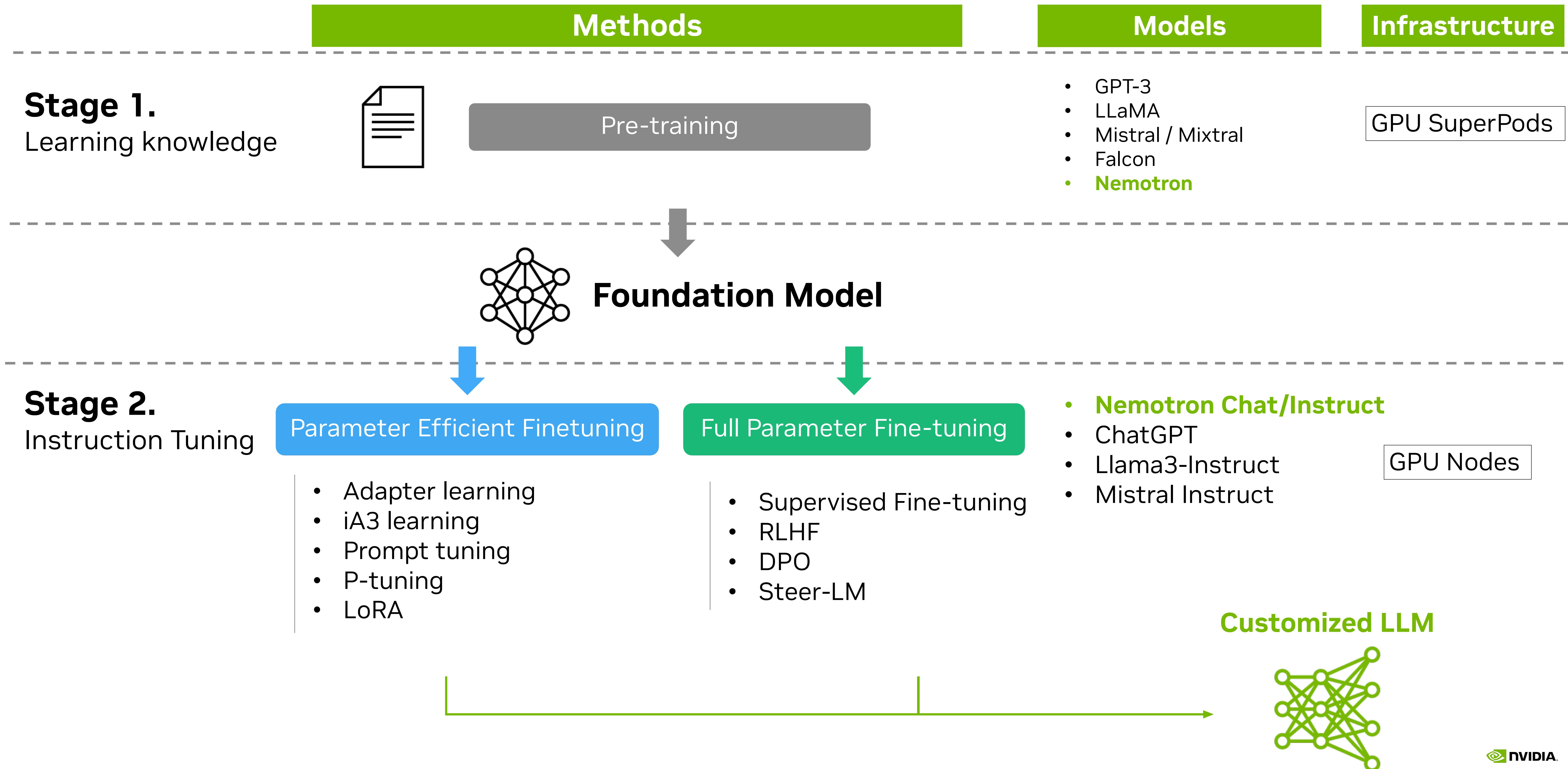
Large Language Model Journey



Large Language Model Journey



Large Language Model Journey

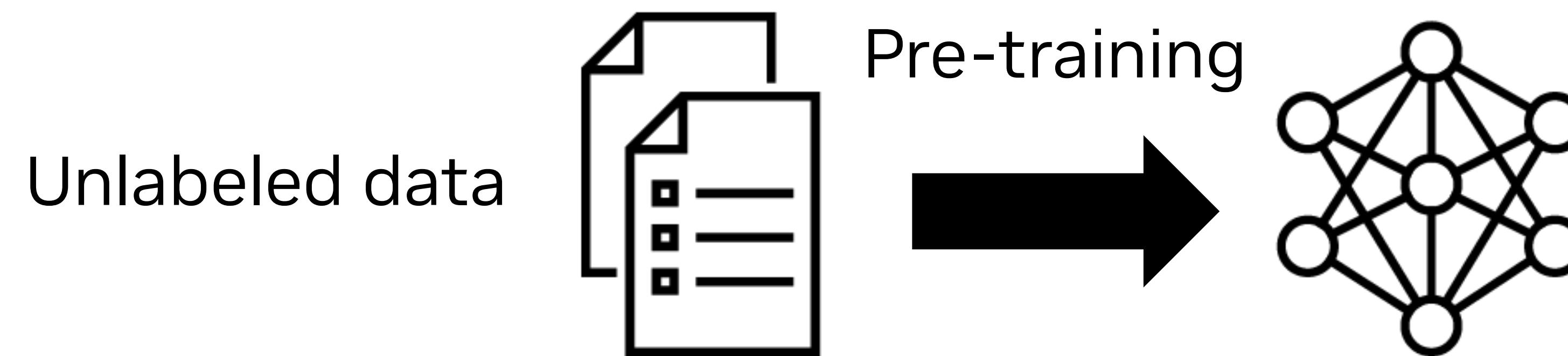


Traditional NLP Approach vs Large Language Models

Fine-tuning vs Instruction Learning

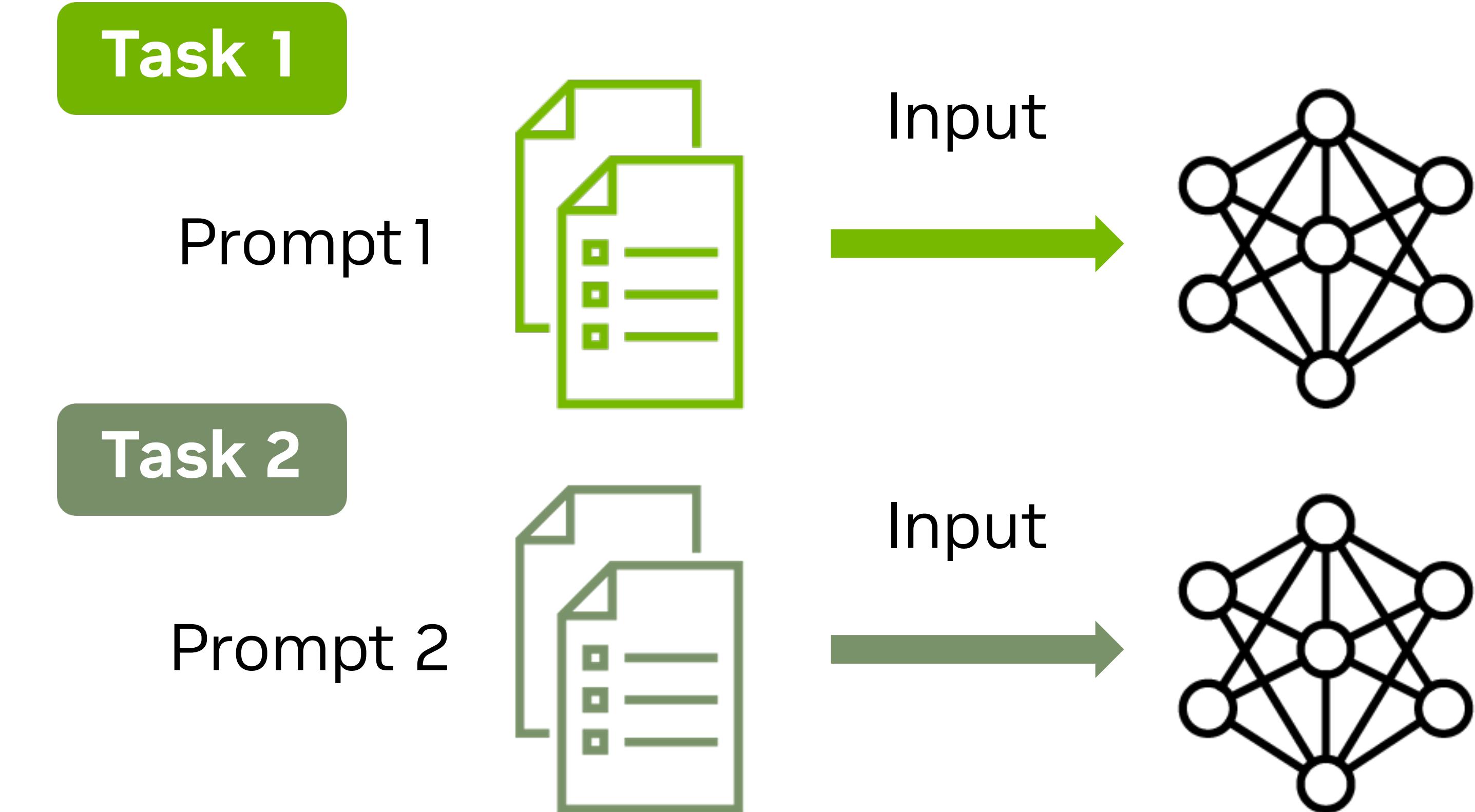
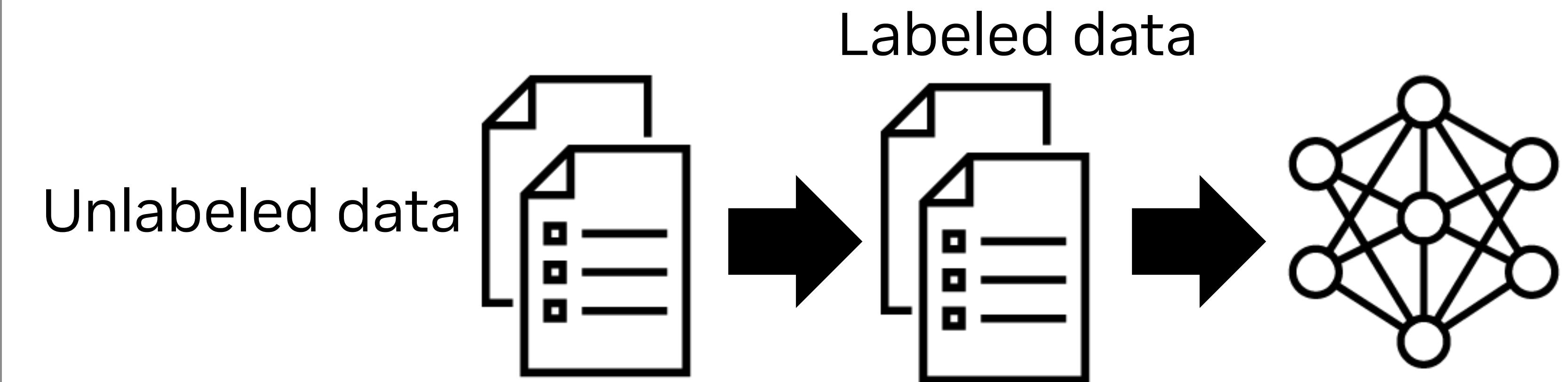
傳統自然語言處理方法

Pre-training + Fine-tuning

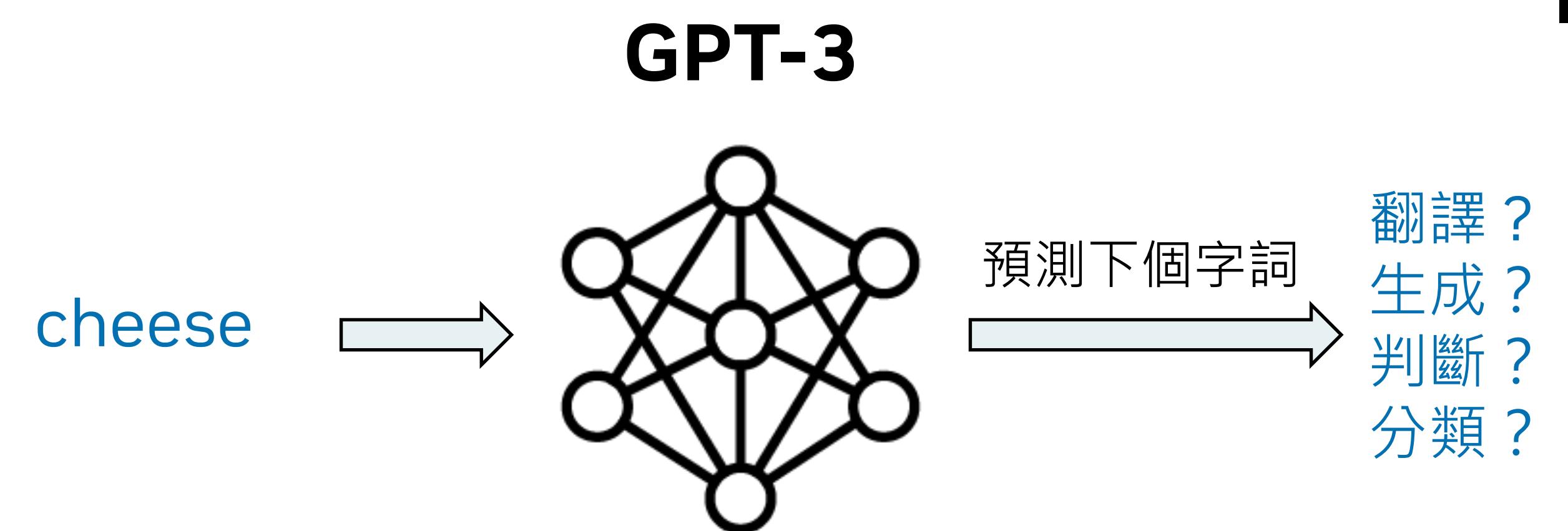


大型語言模型

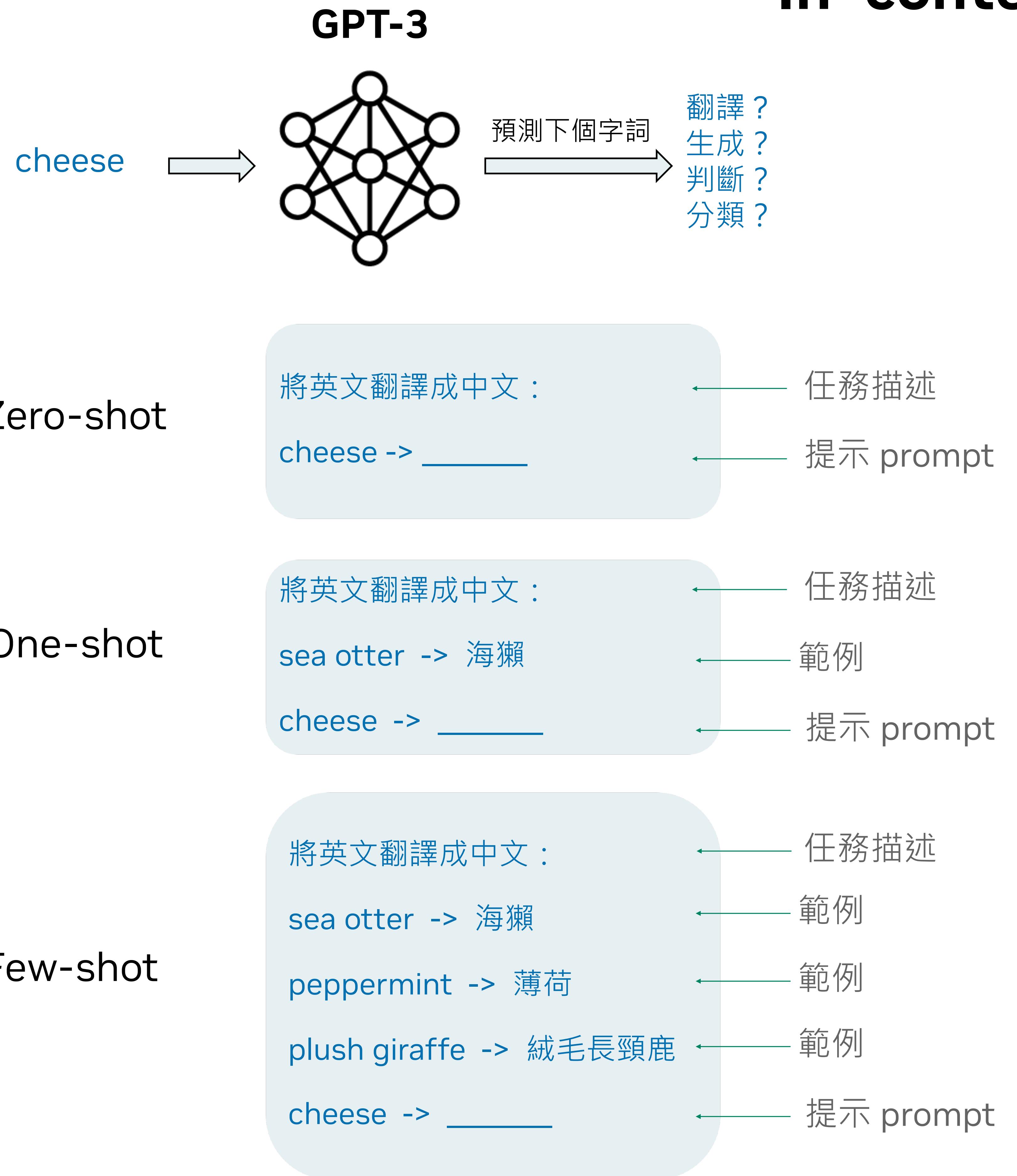
Pre-training + Instruction tuning



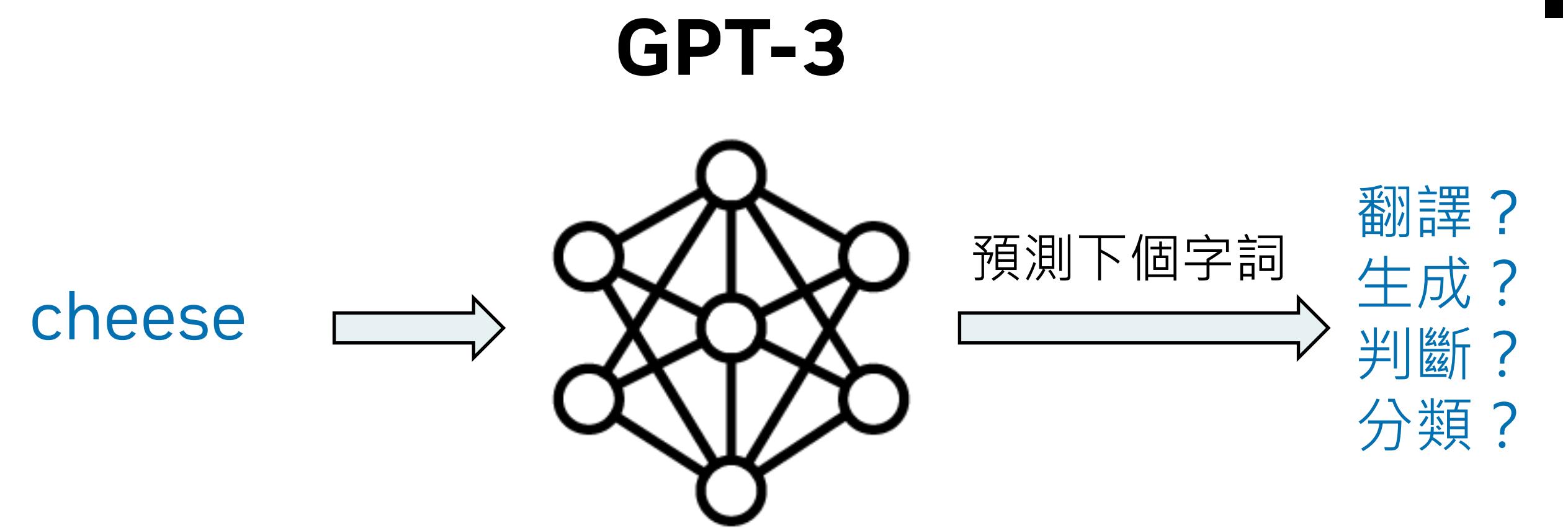
In-context Learning



In-context Learning



In-context Learning



Zero-shot

將英文翻譯成中文：
cheese -> _____

任務描述
提示 prompt

One-shot

將英文翻譯成中文：
sea otter -> 金魚
cheese -> 起司

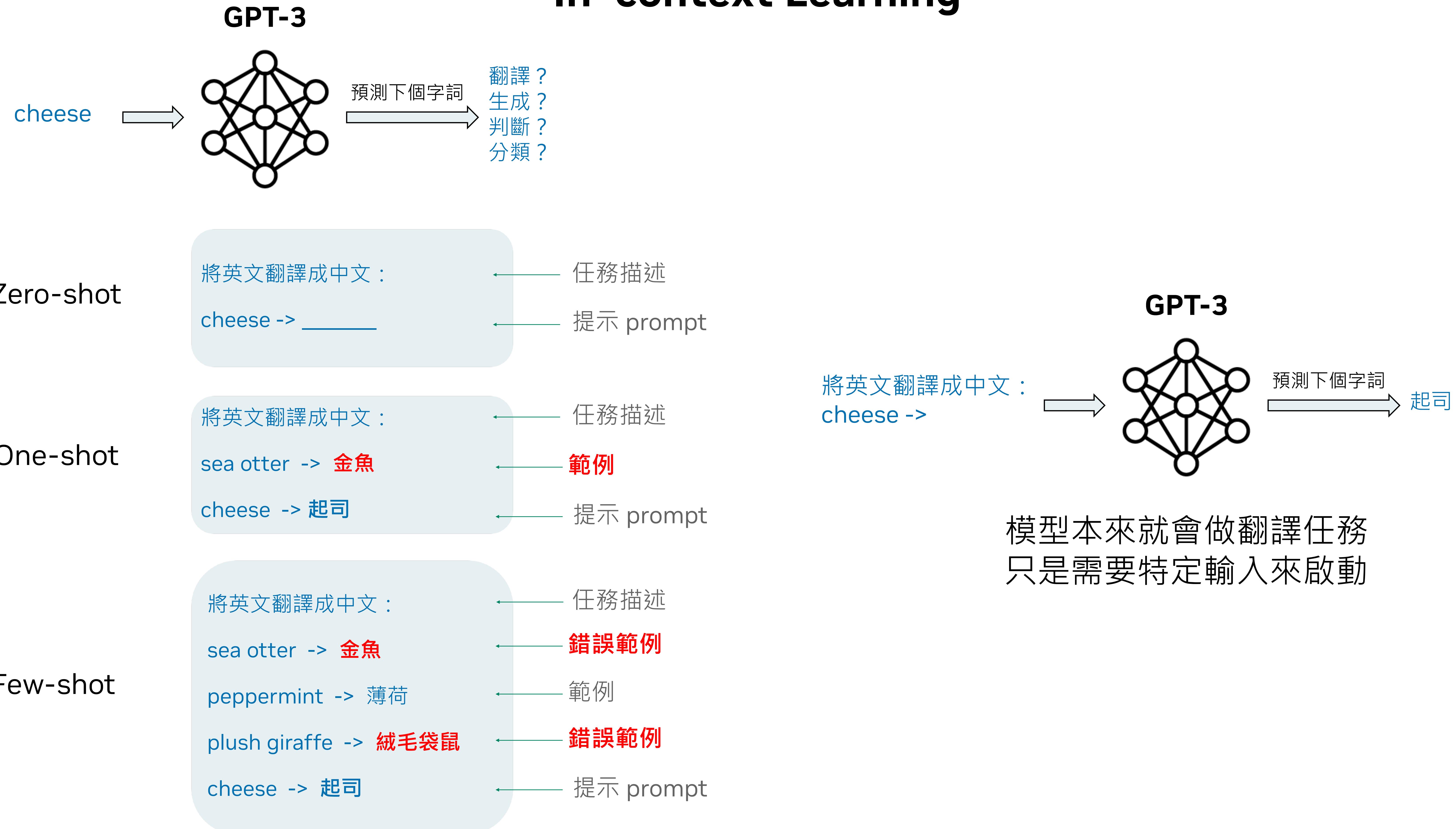
任務描述
範例
提示 prompt

Few-shot

將英文翻譯成中文：
sea otter -> 金魚
peppermint -> 薄荷
plush giraffe -> 絨毛袋鼠
cheese -> 起司

任務描述
錯誤範例
範例
錯誤範例
提示 prompt

In-context Learning



Instruction Tuning

Alignment

Prompt: *Explain the moon landing to a 6 year old in a few sentences.*

GPT-3

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

照樣造句

Instruction Tuning

Alignment

Prompt: *Explain the moon landing to a 6 year old in a few sentences.*

GPT-3

Explain the theory of gravity to a 6 year old.

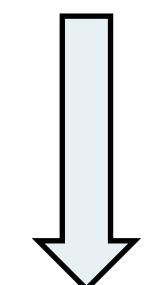
Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

照樣造句

InstructGPT



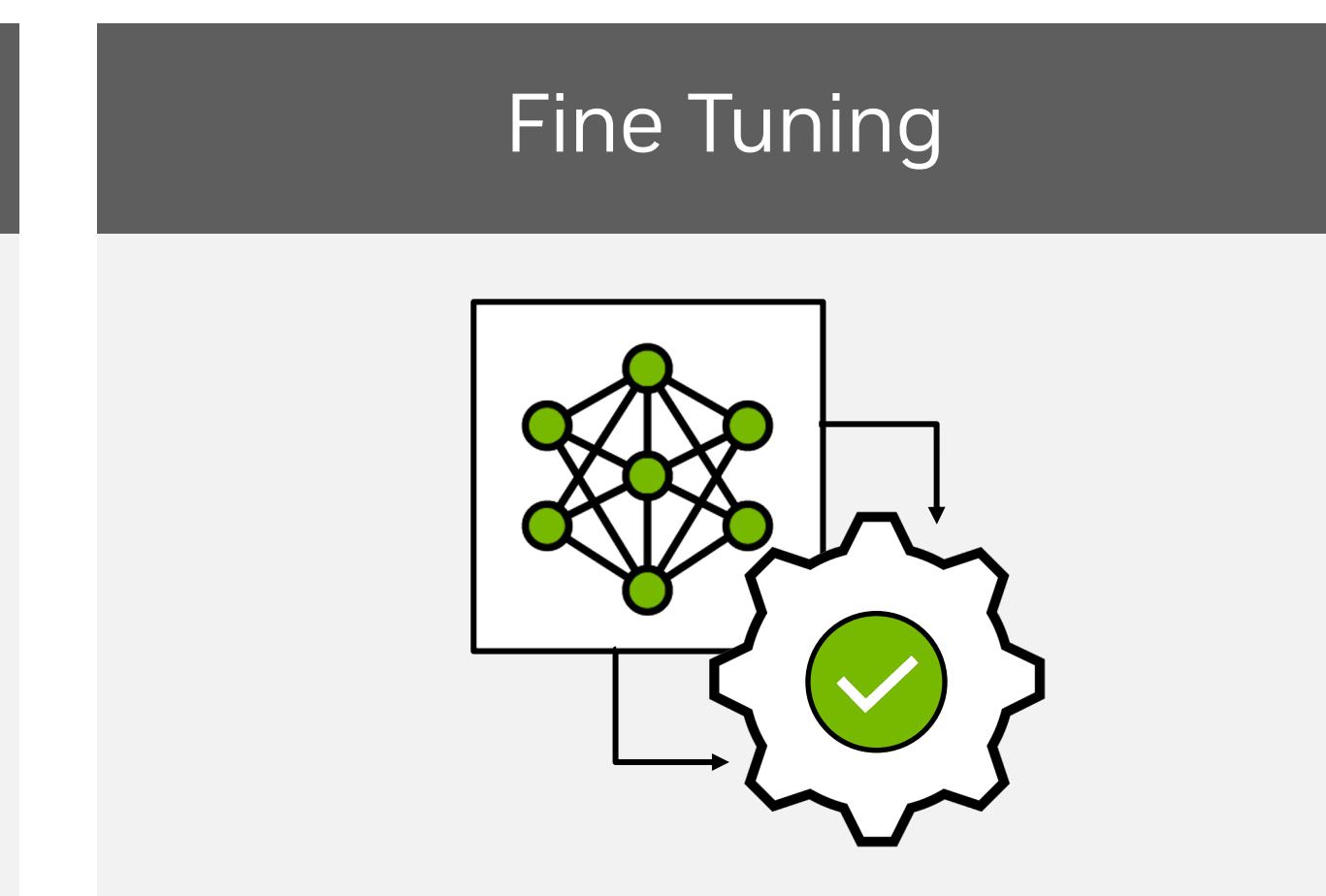
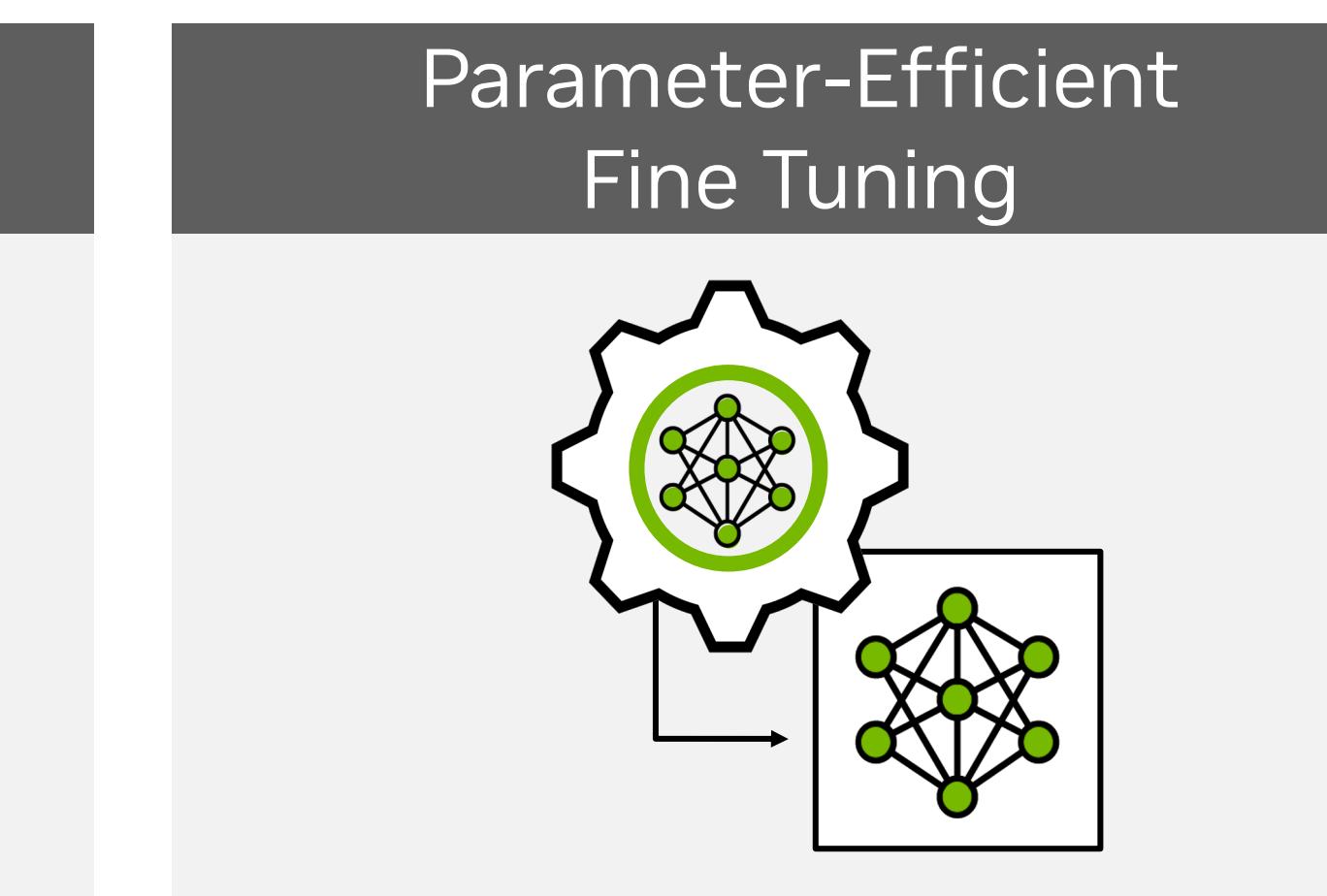
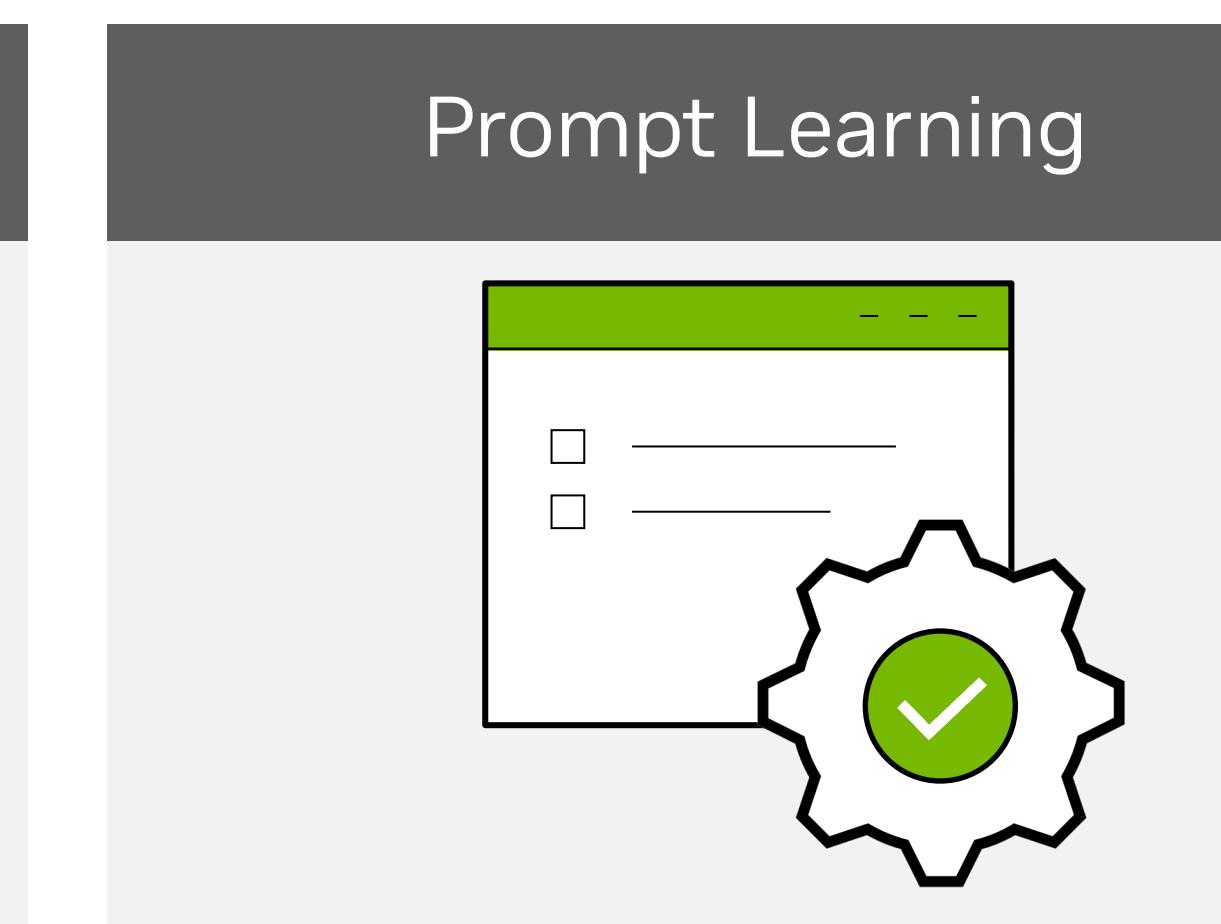
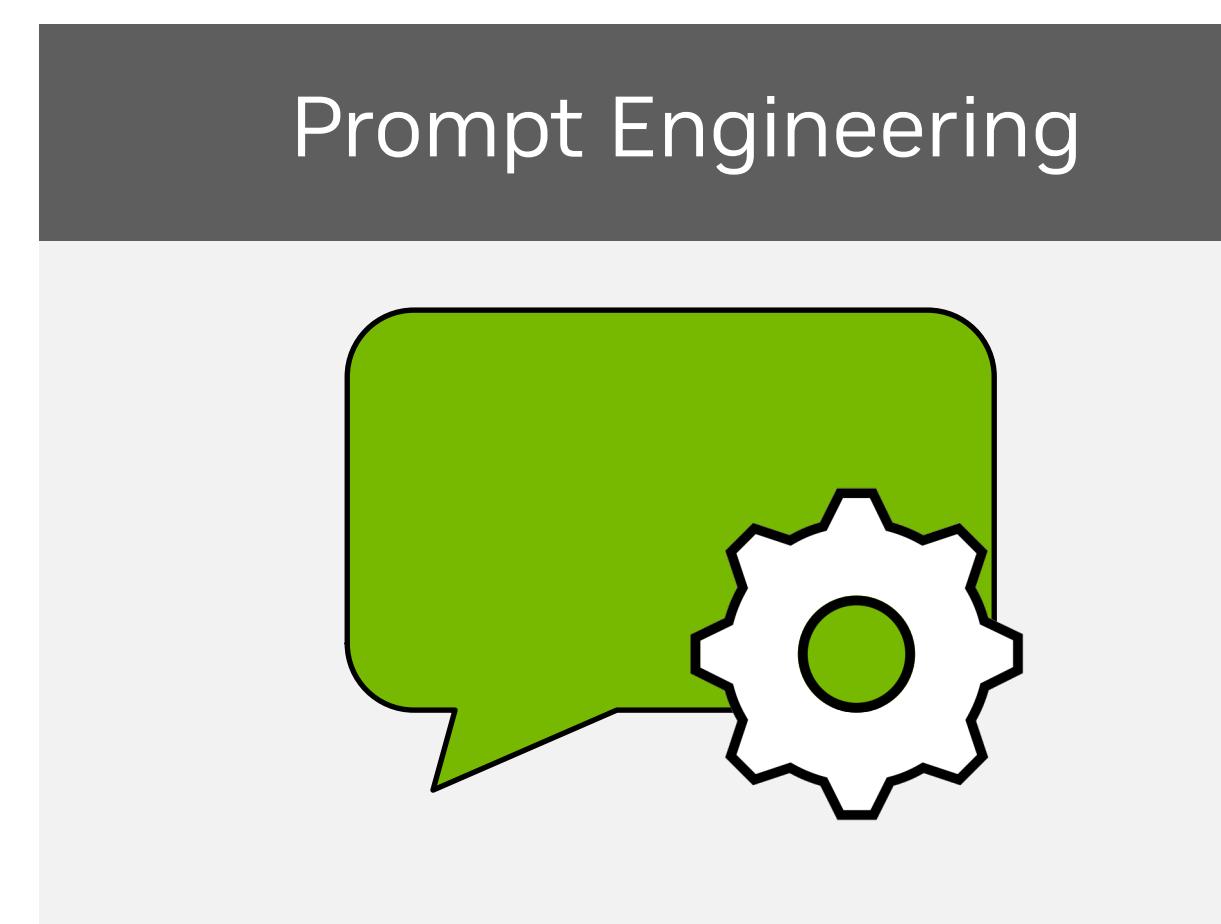
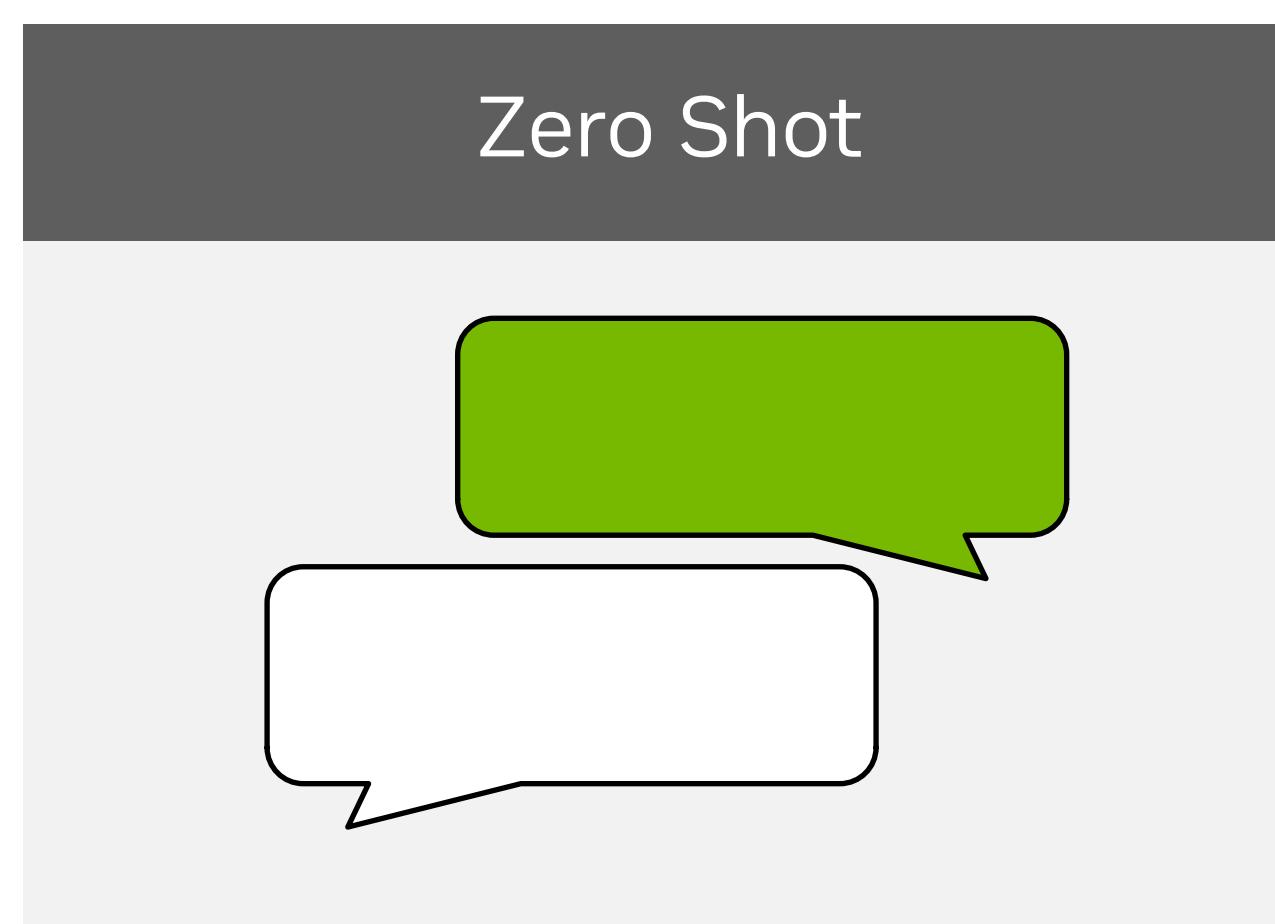
Instruction Tuning

People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

理解問題
完成回答

Selecting LLM Customization Techniques

Custom models tailor language processing capabilities to specific use cases and domain knowledge



Increasing data, compute, and investment

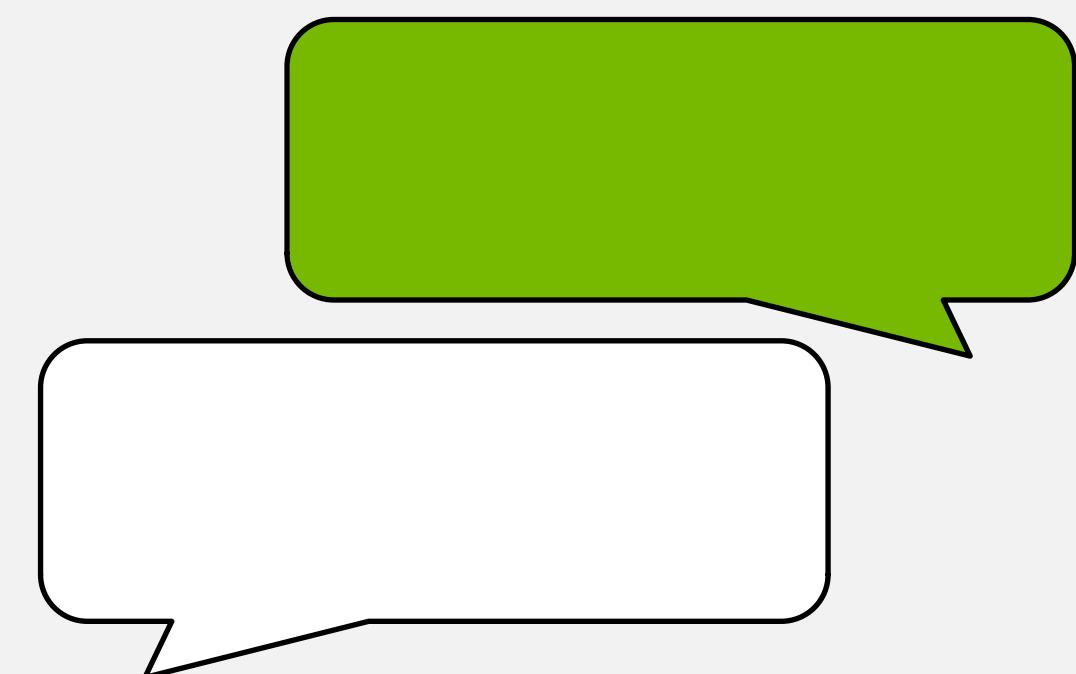
Selecting LLM Customization Techniques

Custom models tailor language processing capabilities to specific use cases and domain knowledge

Zero Shot

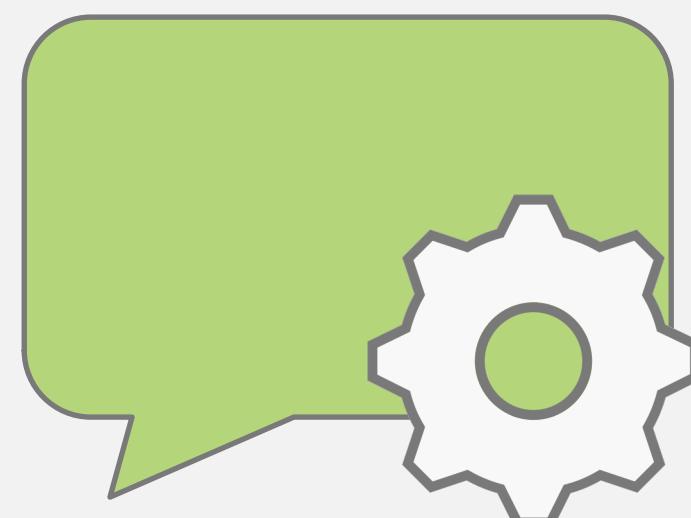
What It Is: Asking the foundation model to perform a task with no previous example or knowledge

Benefits: Simple queries using the expertise that exists in the model out of the box

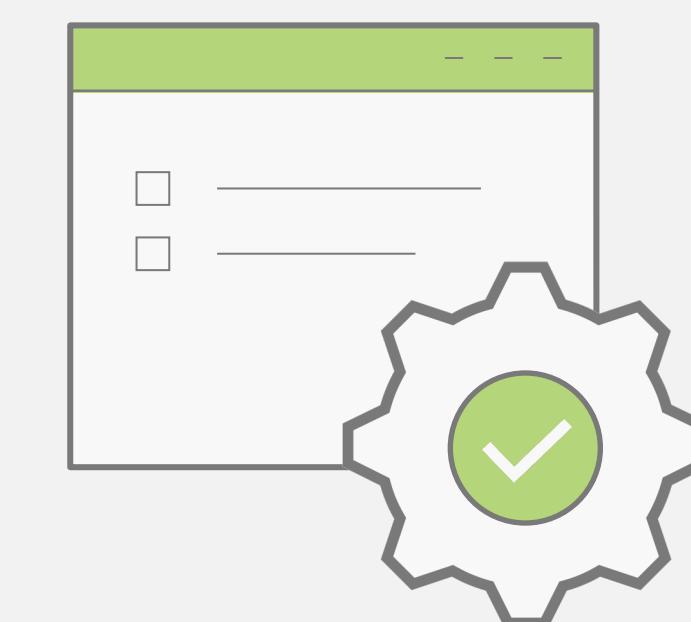


Summarize the following article.
Article:Building, maintaining, and deploying these models require expansive amounts of training data, AI expertise, and compute.
Enterprises can fast-track their generative AI adoption with NVIDIA NeMo service running on NVIDIA DGX Cloud. The quickest path is starting with one of NVIDIA's state-of-the-art pre-trained foundation models.
With the NeMo service, organizations can easily customize a model with p-tuning to teach it specialized skills, like summarizing financial documents, creating brand-specific content, and composing emails with personalized writing styles.
Connecting the model to a proprietary knowledge base ensures that responses are accurate, current, and cited for their business.
And NeMo's playground is available for rapid prototyping before moving to the cloud API for larger-scale evaluation and application integration.
Summary:
The article talks about the importance of AI in the business world.

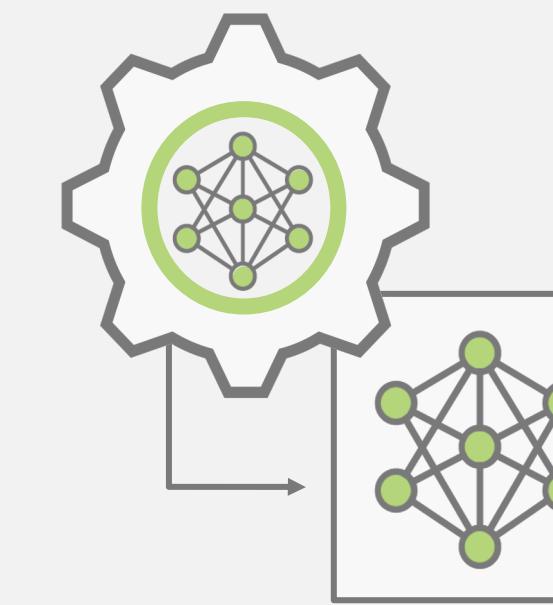
Prompt Engineering



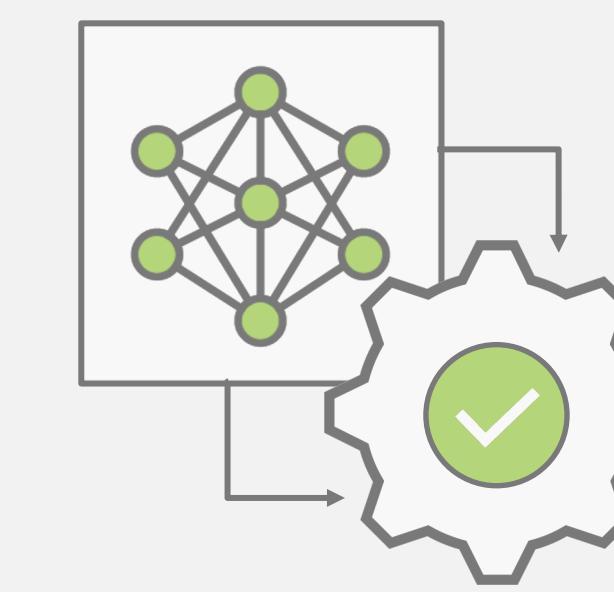
Prompt Learning



Parameter-Efficient Fine Tuning



Fine Tuning



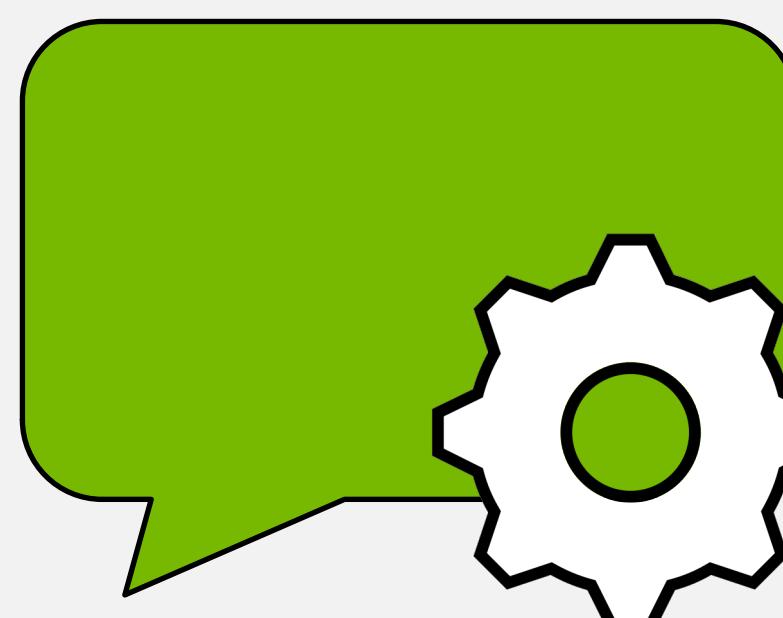
Selecting LLM Customization Techniques

Custom models tailor language processing capabilities to specific use cases and domain knowledge

Prompt Engineering

What It Is: Uses prompt and completion pairs imparting task-specific knowledge to LLMs through virtual tokens

Benefits: Improves responses to queries without having to change the parameters of the model



English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data. Summary: The Transformer architecture based solely on the attention mechanism deliver superior quality on several translation tasks while being more parallelizable and requiring significantly less time to train compared to recurrence and convolution alternatives.

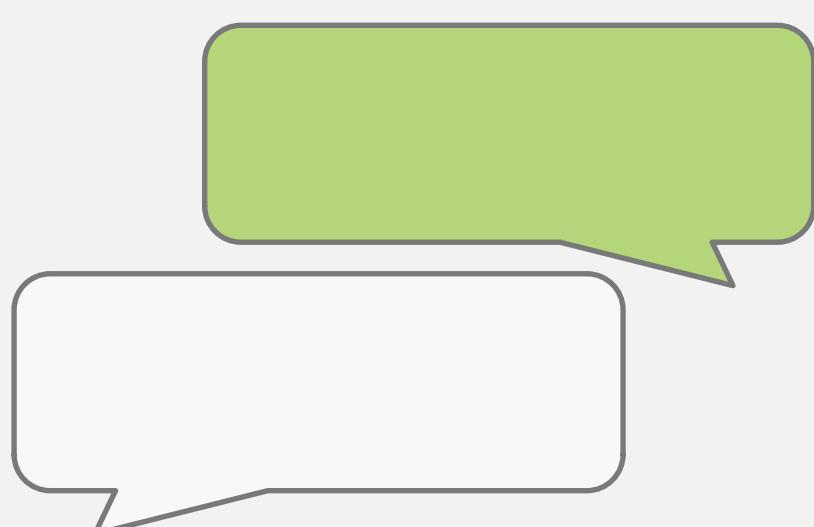
==

Summarize the following article:

Article: Building, maintaining, and deploying these models require expansive amounts of training data, AI expertise, and compute. Enterprises can fast-track their generative AI adoption with NVIDIA NeMo service running on NVIDIA DGX Cloud. The quickest path is starting with one of NVIDIA's state-of-the-art pre-trained foundation models. With the NeMo service, organizations can easily customize a model with p-tuning to teach it specialized skills, like summarizing financial documents, creating brand-specific content, and composing emails with personalized writing styles. Connecting the model to a proprietary knowledge base ensures that responses are accurate, current, and cited for their business. NeMo's playground is available for rapid prototyping before moving to the cloud API for larger-scale evaluation and application integration.

Summary: NVIDIA's NeMo service allows enterprises to build, customize, and deploy generative AI models with minimal AI expertise and compute.

Zero Shot



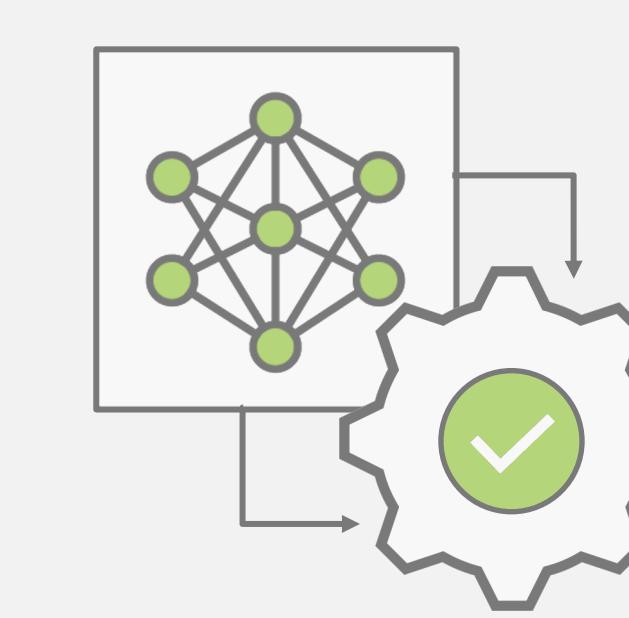
Prompt Learning



Parameter-Efficient Fine Tuning



Fine Tuning



Selecting LLM Customization Techniques

Custom models tailor language processing capabilities to specific use cases and domain knowledge

Prompt Learning

What It Is: Training a prompt-model with 100s to 1000s of examples that get sent to the foundation model at inference time

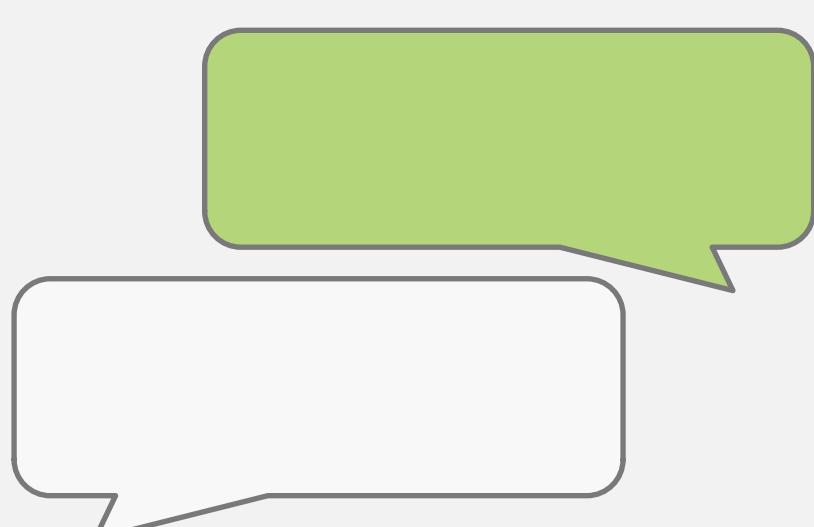
Benefits: Enables new tasks to LLMs without overwriting or disrupting previous tasks that the model has been trained on



Summarize the following article.
Article:Building, maintaining, and deploying these models require expansive amounts of training data, AI expertise, and compute.
Enterprises can fast-track their generative AI adoption with NVIDIA NeMo service running on NVIDIA DGX Cloud. The quickest path is starting with one of NVIDIA's state-of-the-art pre-trained foundation models.
With the NeMo service, organizations can easily customize a model with p-tuning to teach it specialized skills, like summarizing financial documents, creating brand-specific content, and composing emails with personalized writing styles.
Connecting the model to a proprietary knowledge base ensures that responses are accurate, current, and cited for their business.
And NeMo's playground is available for rapid prototyping before moving to the cloud API for larger-scale evaluation and application integration.
Summary:

Enterprises can fast-track their generative AI adoption with NVIDIA NeMo service running on NVIDIA DGX Cloud.

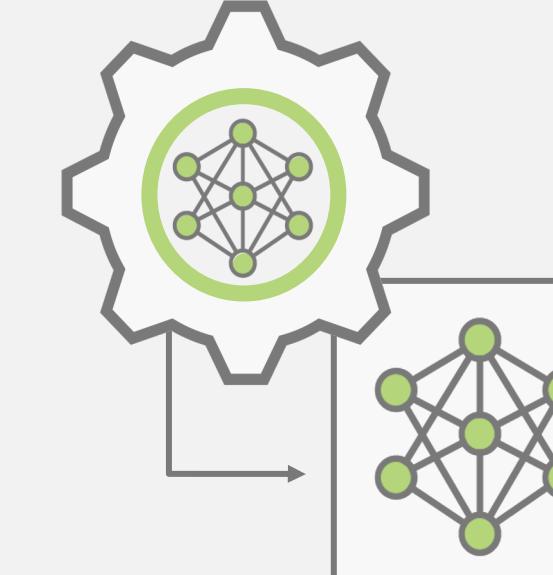
Zero Shot



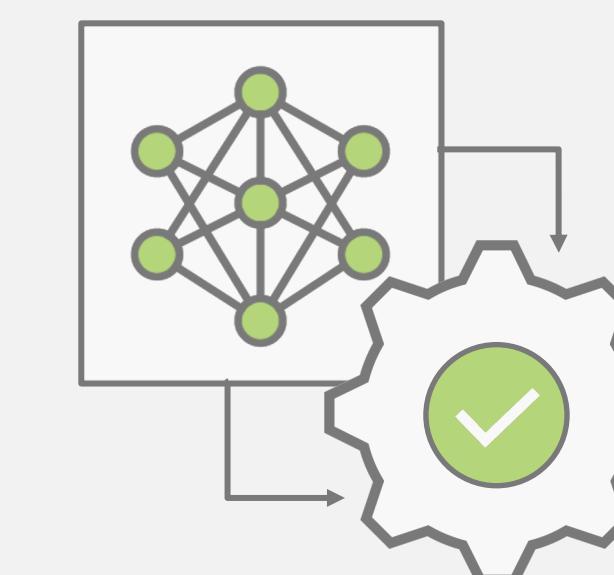
Prompt Engineering



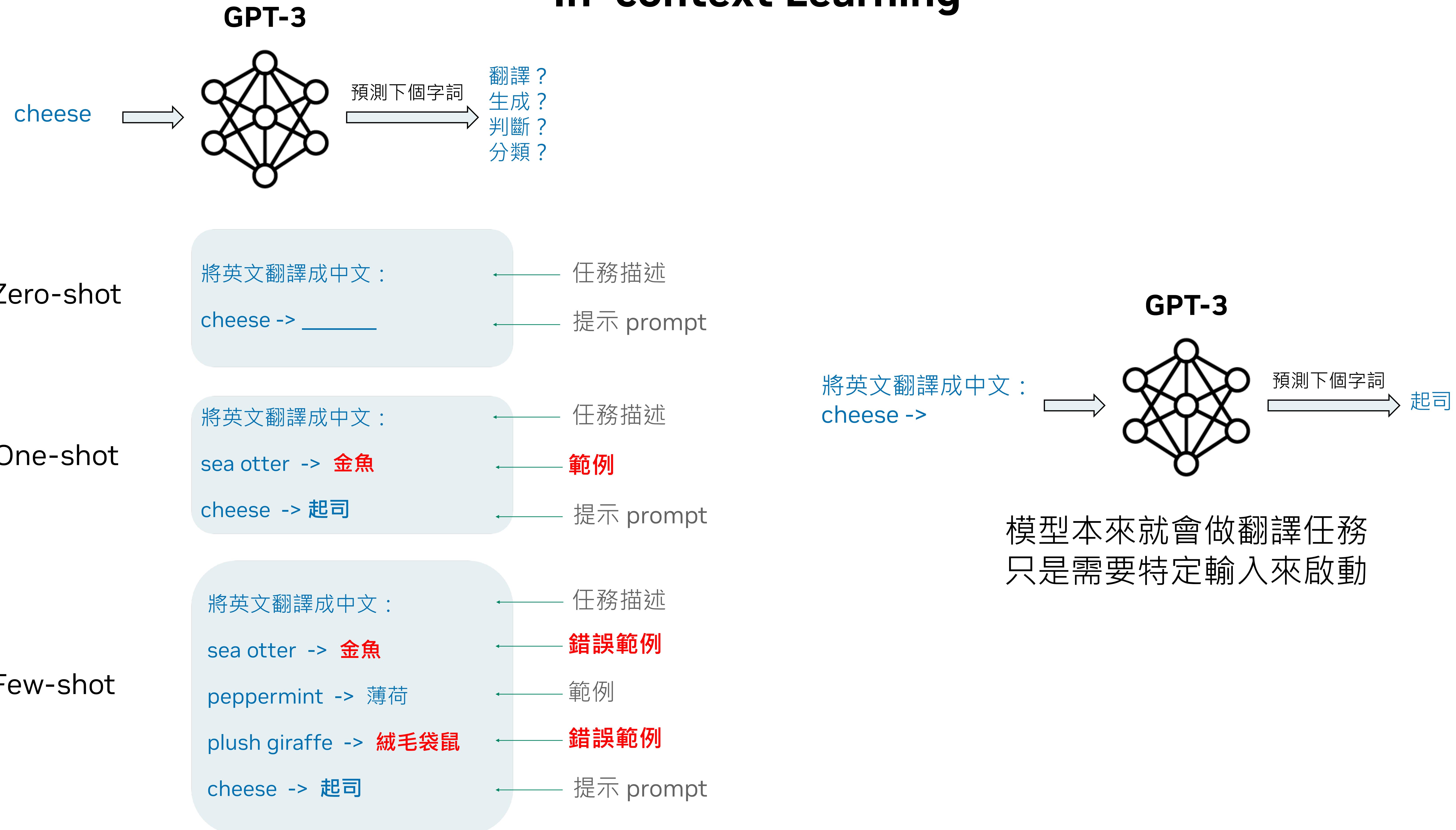
Parameter-Efficient Fine Tuning



Fine Tuning



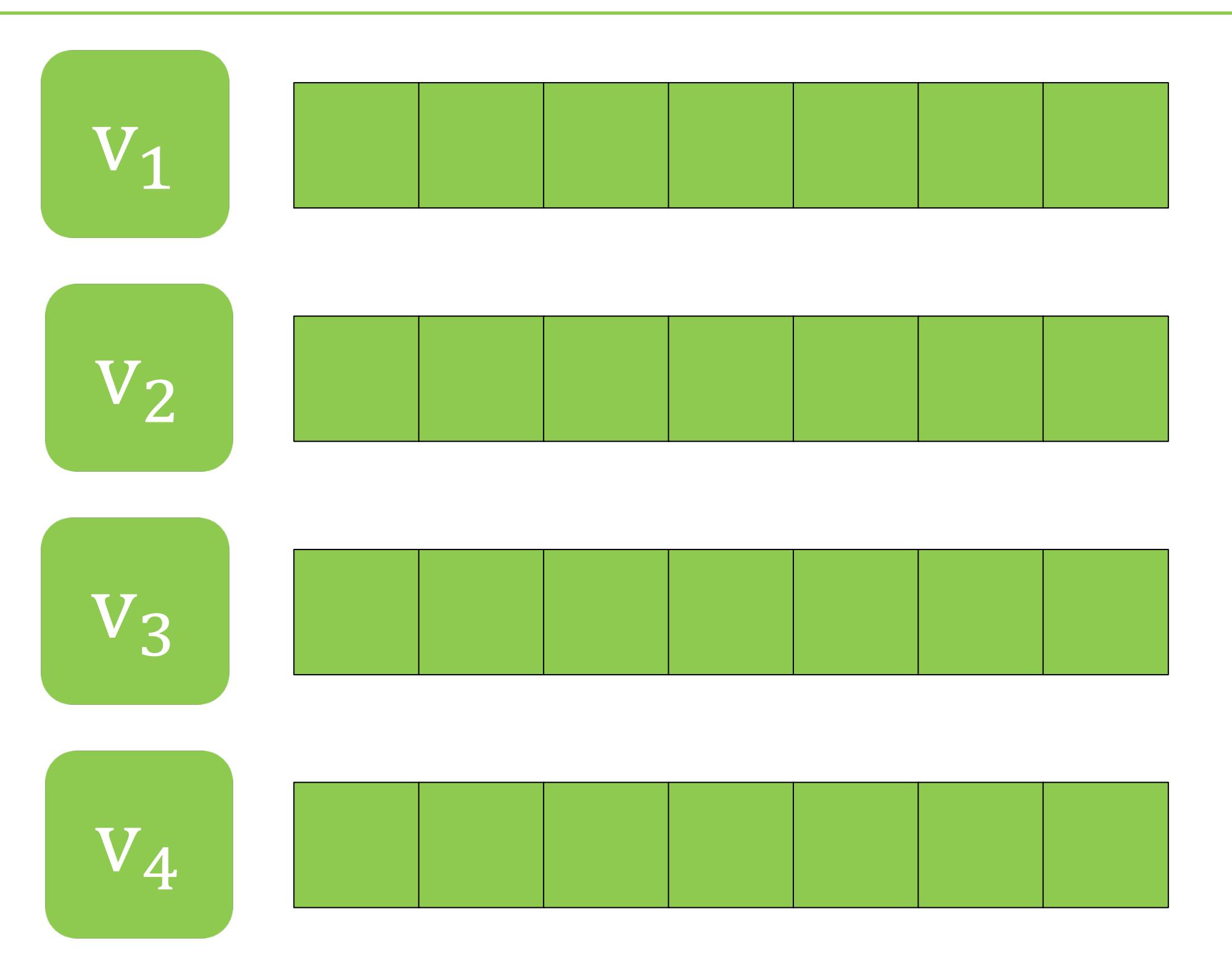
In-context Learning



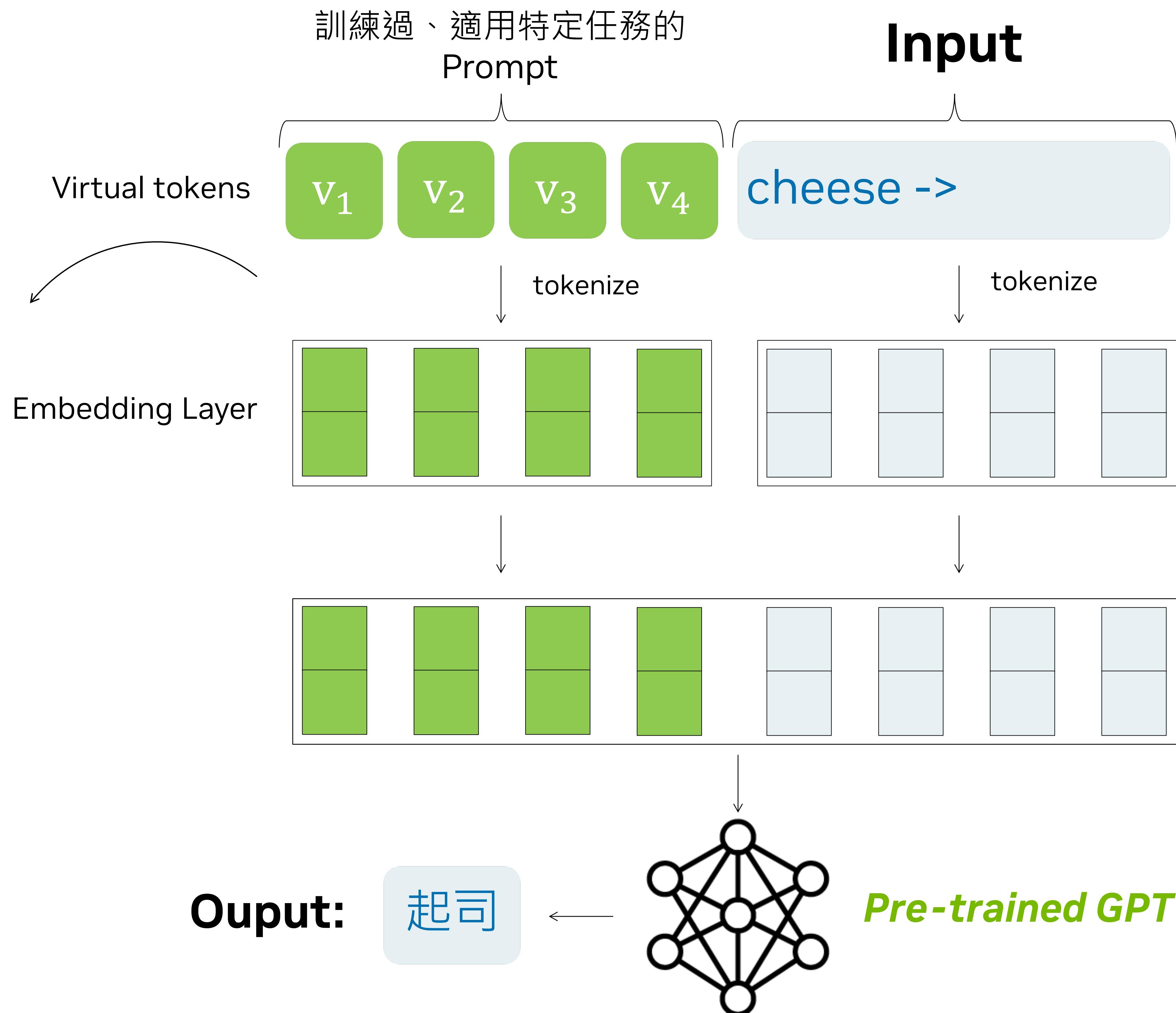
Prompt Learning

P-tuning

Virtual token embeddings



total number of **virtual tokens** \times **hidden_size**



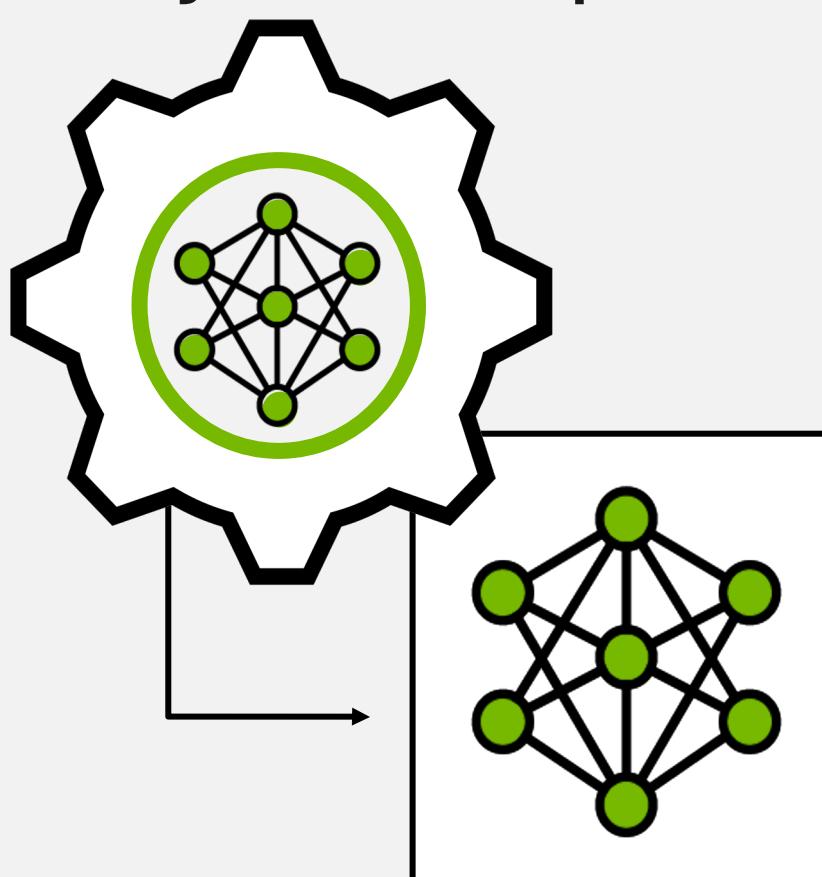
Selecting LLM Customization Techniques

Custom models tailor language processing capabilities to specific use cases and domain knowledge

Parameter-Efficient Fine Tuning

What It Is: Uses clever optimizations to selectively add and update few parameters or layers to the original LLM architecture

Benefits: Enables LLMs to reach high accuracy on trained tasks using domain and task-specific datasets while updating significantly fewer parameters

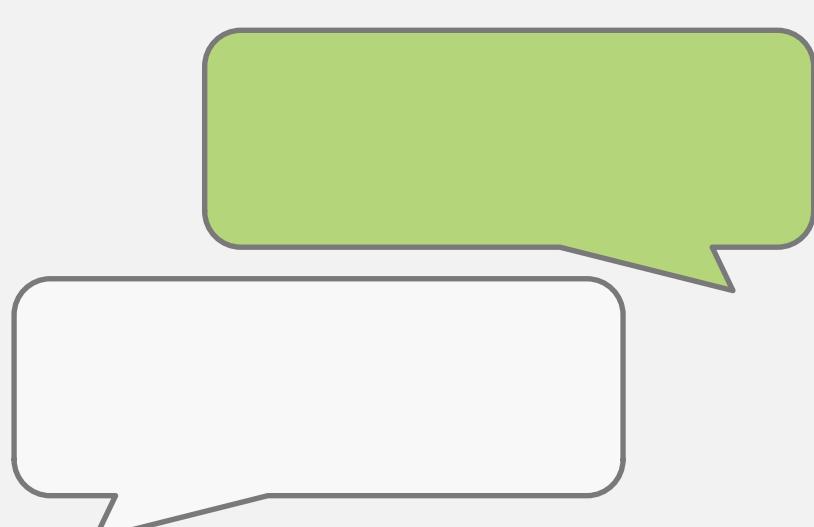


Introduces a small number of parameters or layers to existing LLM architecture and is trained with use-case-specific data, providing higher accuracy than prompt engineering and prompt learning, while requiring more training data and compute.

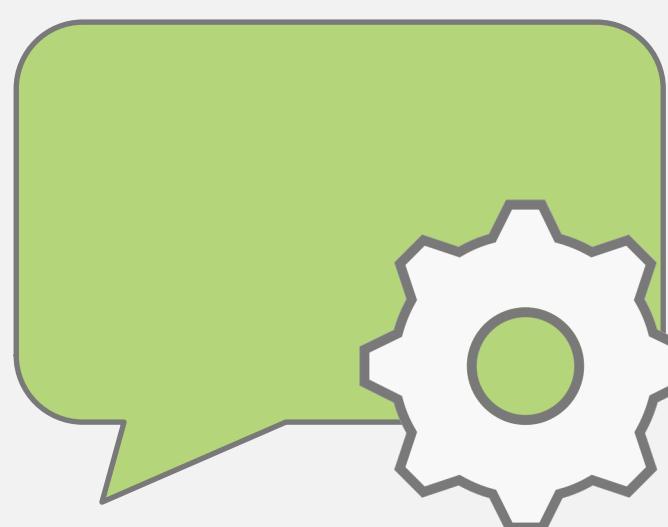
Unlike prompt learning, these methods do not insert virtual prompts into the input. Instead, they introduce trainable layers into the transformer architecture for task-specific learning. This helps attain strong performance on downstream tasks while reducing the number of trainable parameters by several orders of magnitude (closer to 10,000x fewer parameters) compared to fine-tuning.

- Adapter Learning
- Infused Adapter by Inhibiting & Amplifying Inner Activations (IA3)
- Low-Rank Adaptation (LoRA)

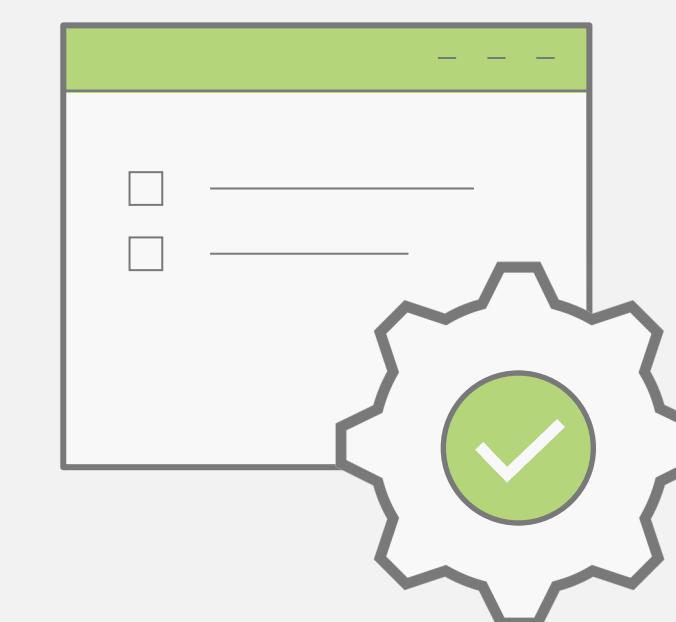
Zero Shot



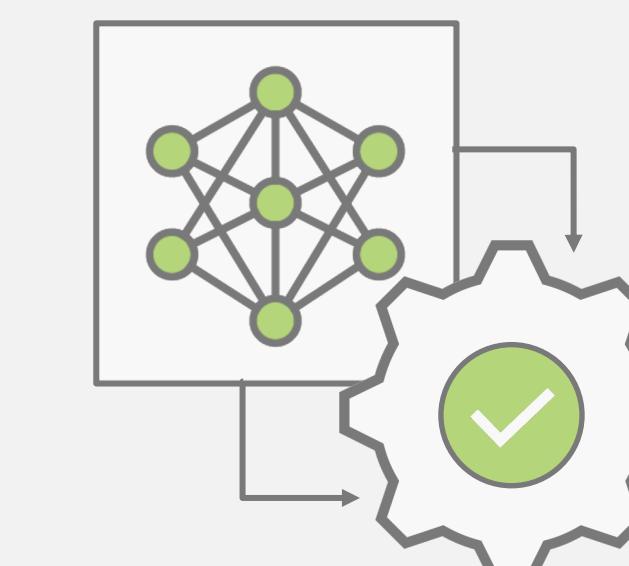
Prompt Engineering



Prompt Learning



Fine Tuning



LoRA: Low-Rank Adaptation

Efficient training with fewer trainable parameters

Aghajanyan et al. (2020) shows that the pre-trained language models have a low **intrinsic dimension** and can still learn efficiently despite a random projection to a smaller subspace.

Inspired by this, they hypothesize the updates to the weights also have a low “intrinsic rank” during adaptation

Pre-trained

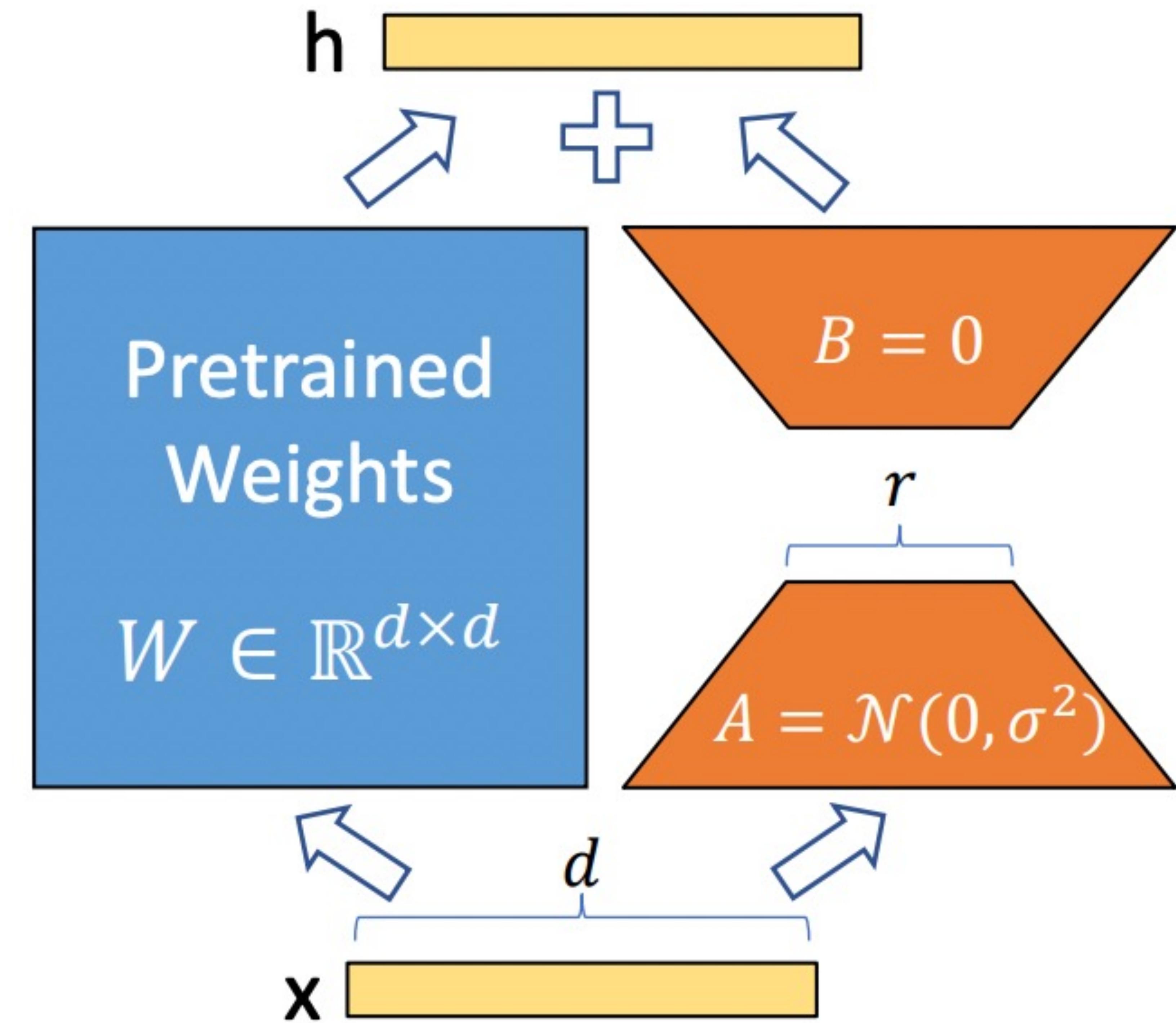
$$h = W_0x$$

Fine-tuned

$$h = W_0x + \Delta Wx$$

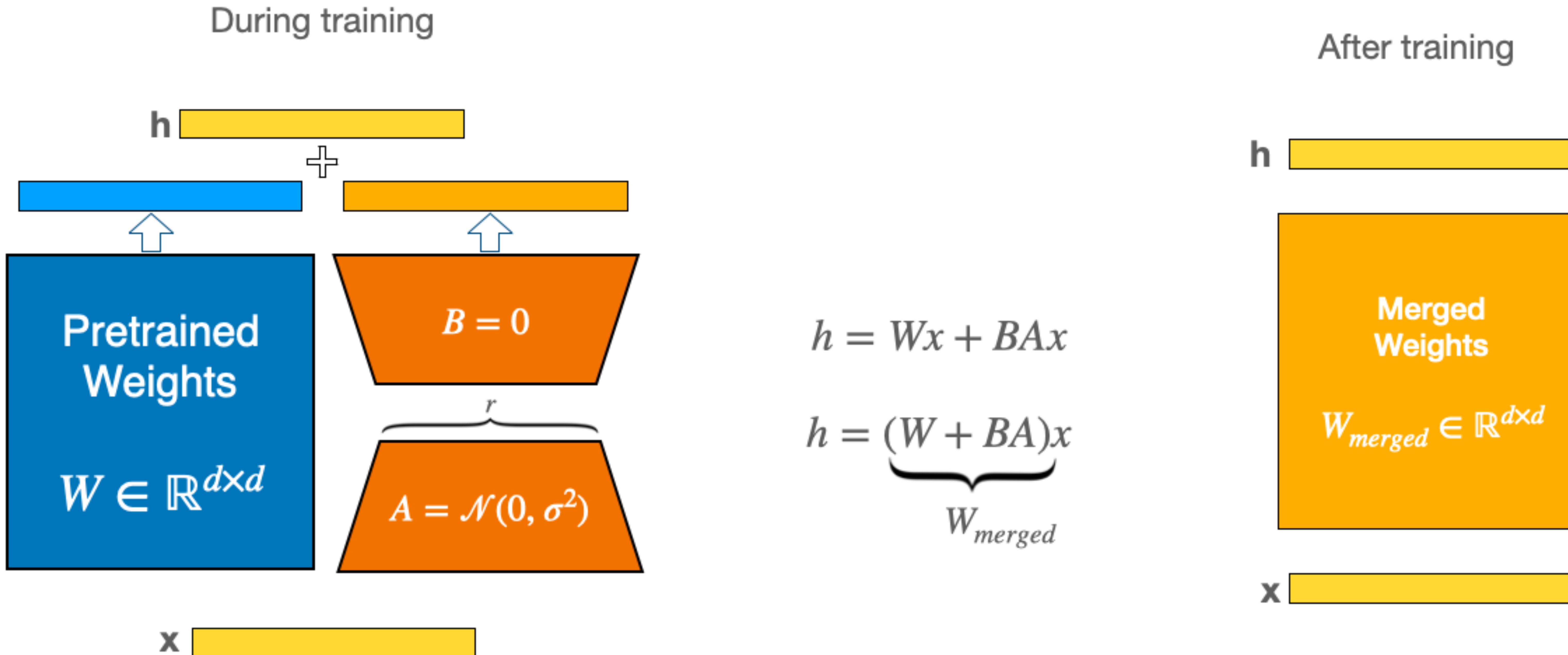
Low rank adaptation

$$h = W_0x + \Delta Wx = W_0x + BAx$$



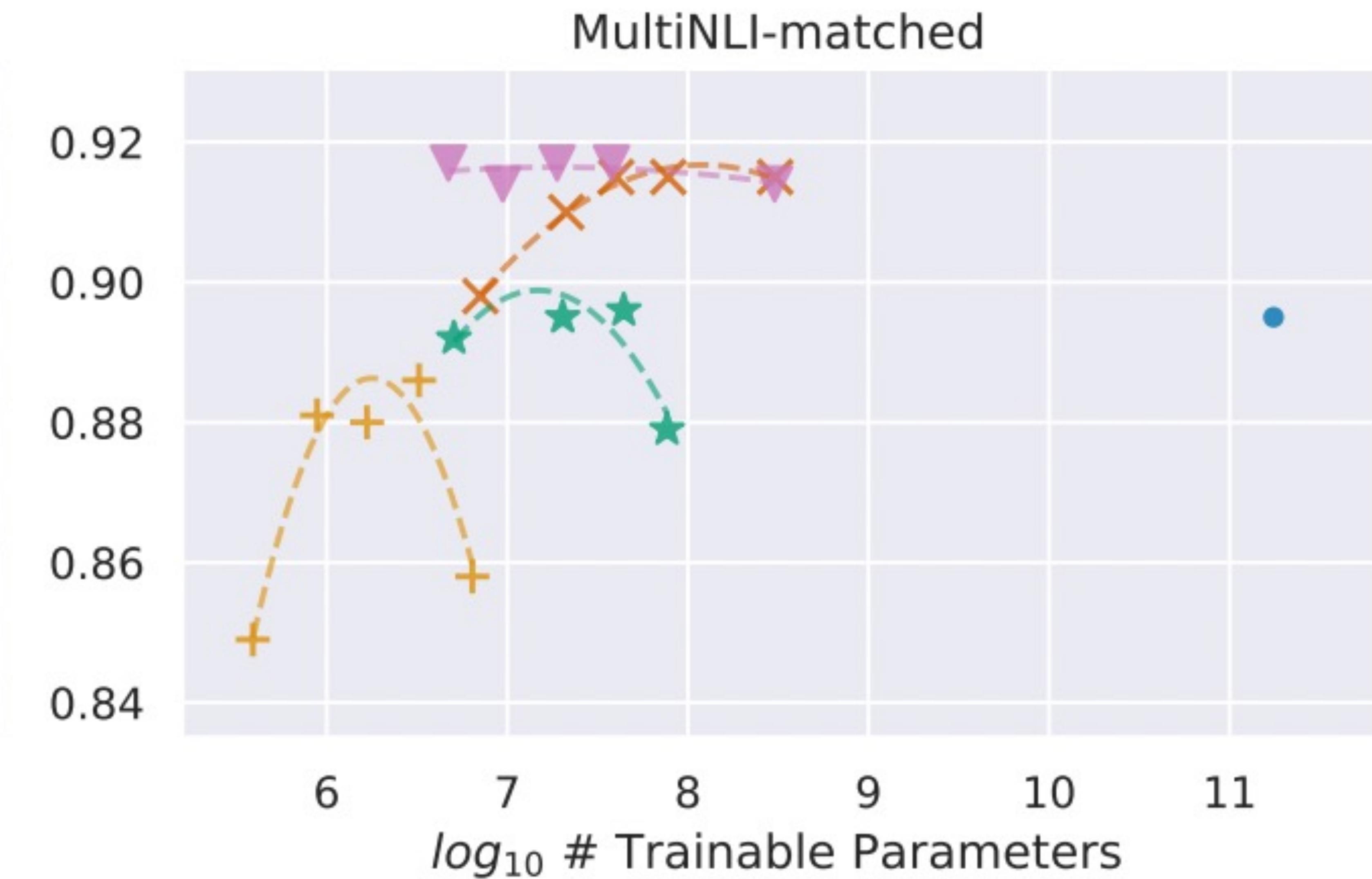
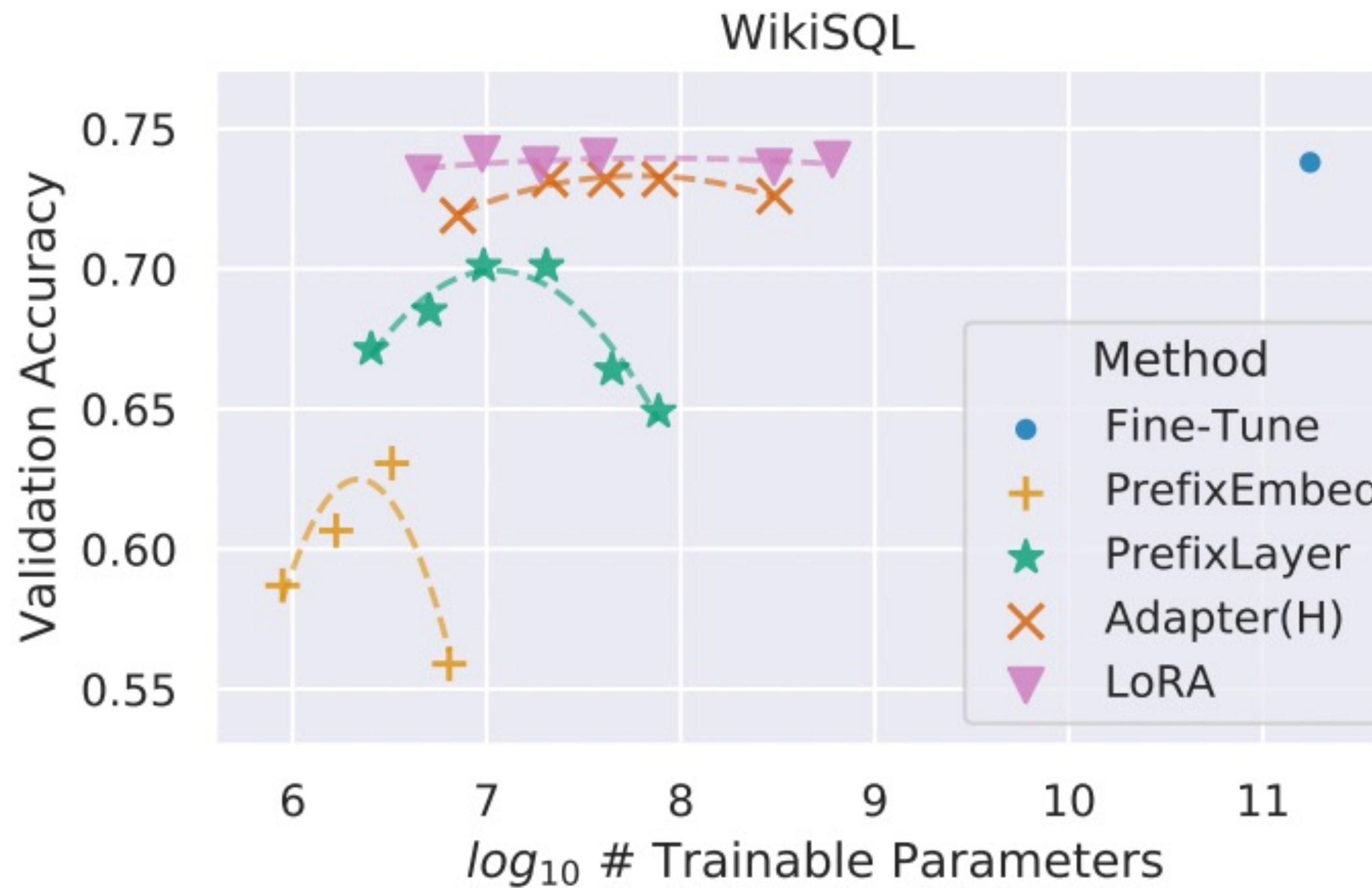
LoRA: Low-Rank Adaptation

Efficient training with fewer trainable parameters



LoRA: Low-Rank Adaptation

Efficient training with fewer trainable parameters



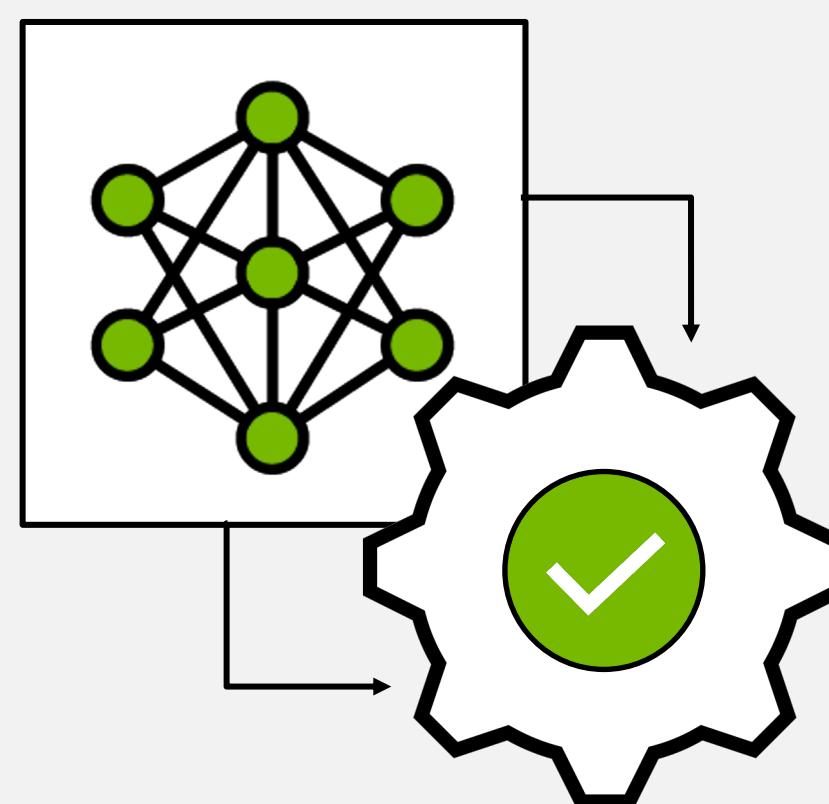
Selecting LLM Customization Techniques

Custom models tailor language processing capabilities to specific use cases and domain knowledge

Fine Tuning

What It Is: fine-tuning all model parameters on labeled data that teaches the model domain-specific terms and how to follow user-specified instructions

Benefits: Achieve the best accuracy on a range of use cases as compared to other customization approaches.

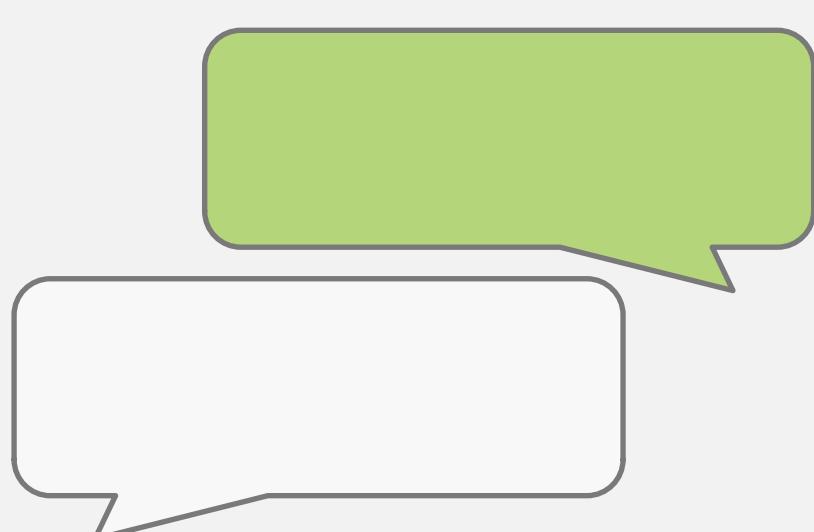


When data and compute resources have no hard constraints, customization techniques such as supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) are great alternative approaches to PEFT and prompt engineering. Fine-tuning can help achieve the best accuracy on a range of use cases as compared to other customization approaches.

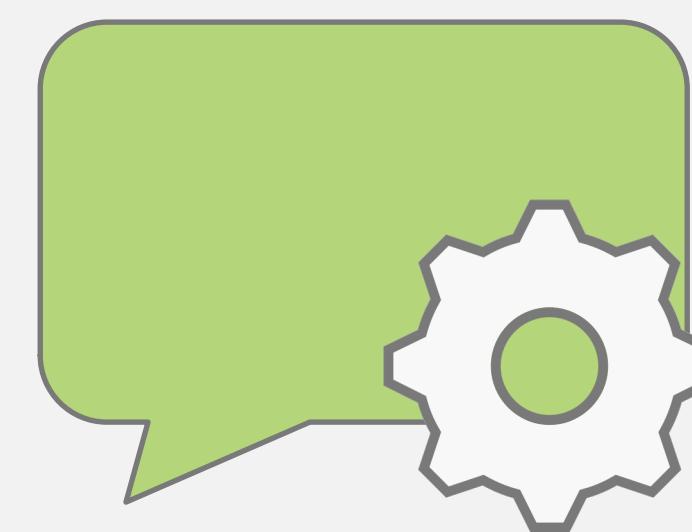
Fine-tuning is used for Supervised-Fine-Tuning (SFT) for RLHF
e.g. Retrain the LLM with ~50M additional tokens for your flavor
(instruction following, chat and dialogue, etc)

Fine-tuning is used to retrain both the retriever and generator models of NeMo Retriever
e.g. Train on 100's of Question / Answer / Source datasets for best results in a factual question and answer retrieval use case

Zero Shot



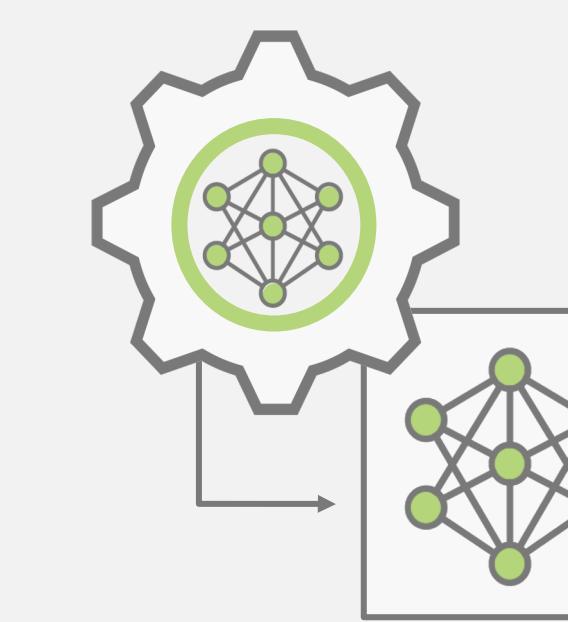
Prompt Engineering



Prompt Learning

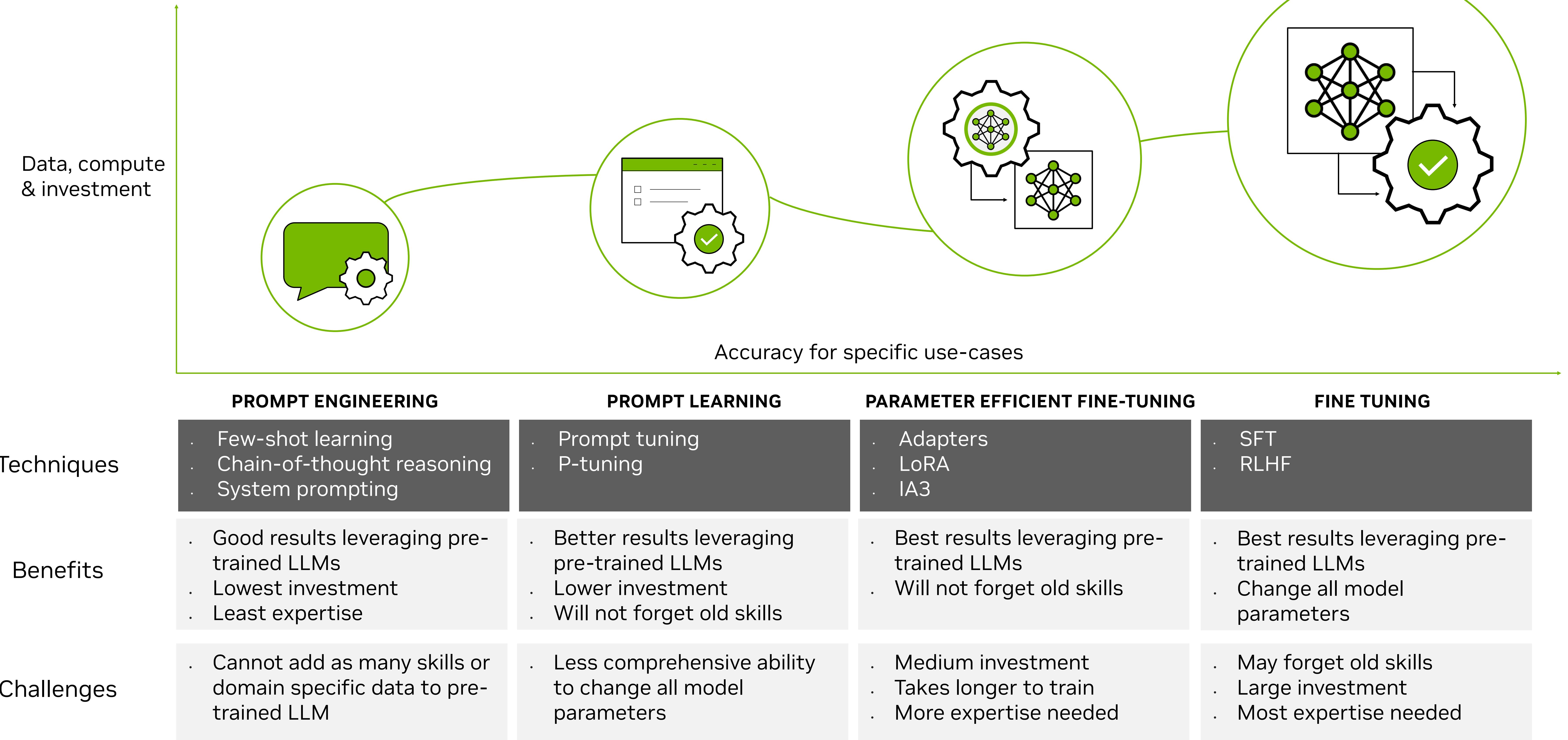


Parameter-Efficient Fine Tuning



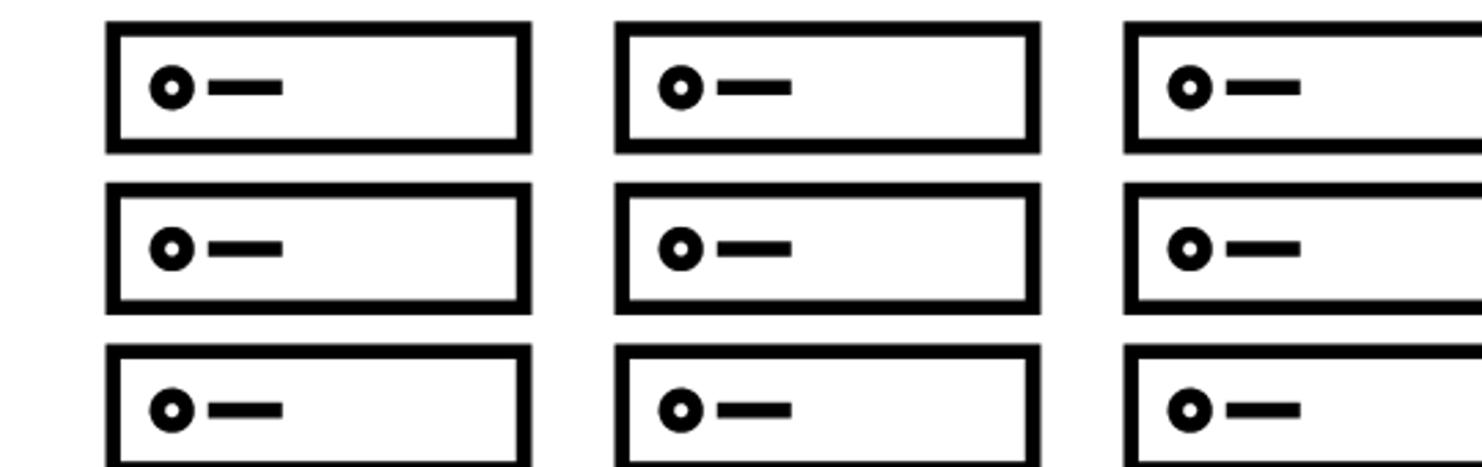
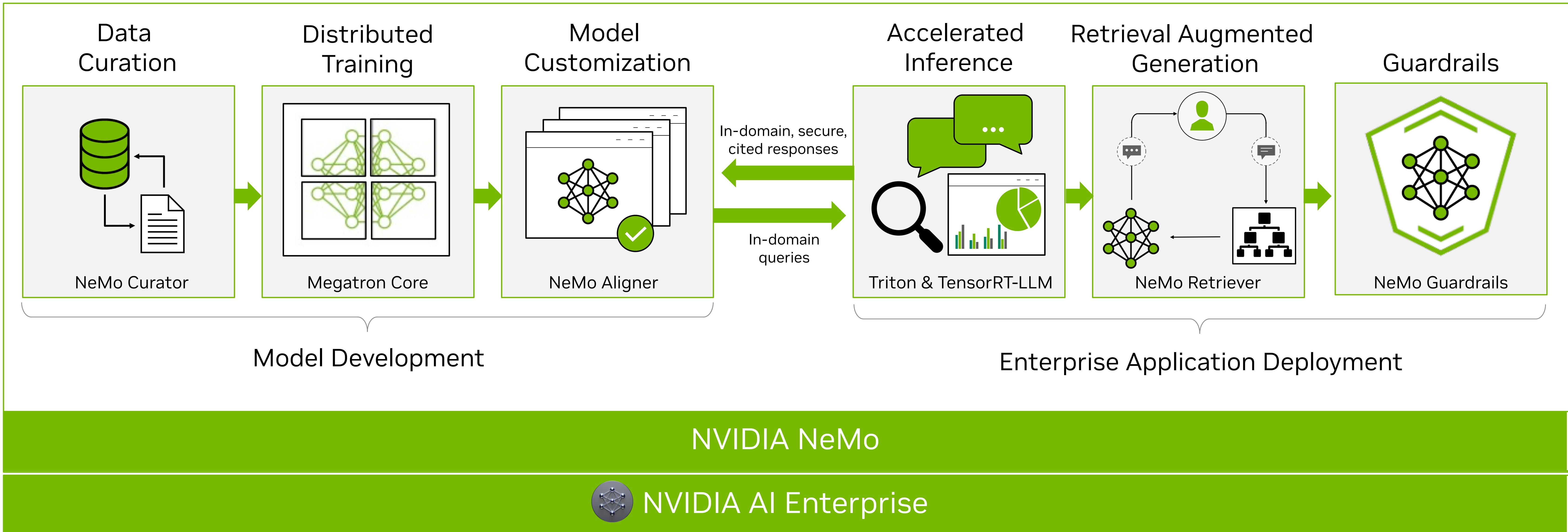
Suite of Model Customization Tools in NeMo

Ways To Customize Large Language Models For Your Use-Cases

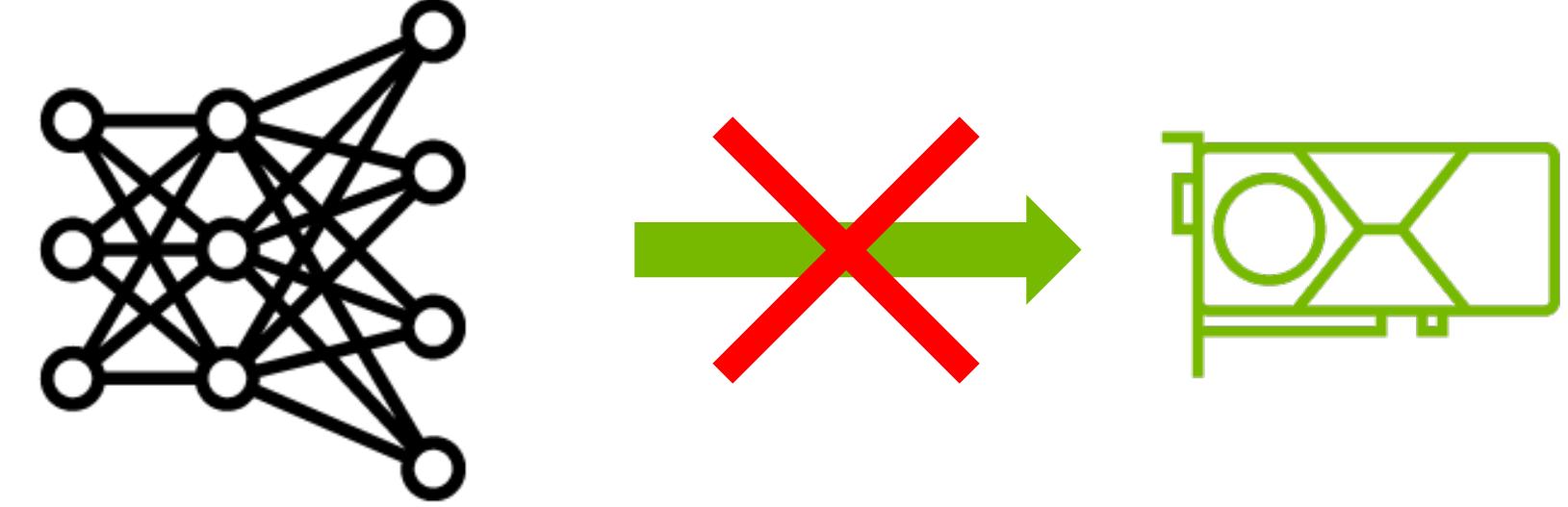


Building Generative AI Applications for the Enterprise

Build, customize and deploy generative AI models with NVIDIA NeMo



LLM

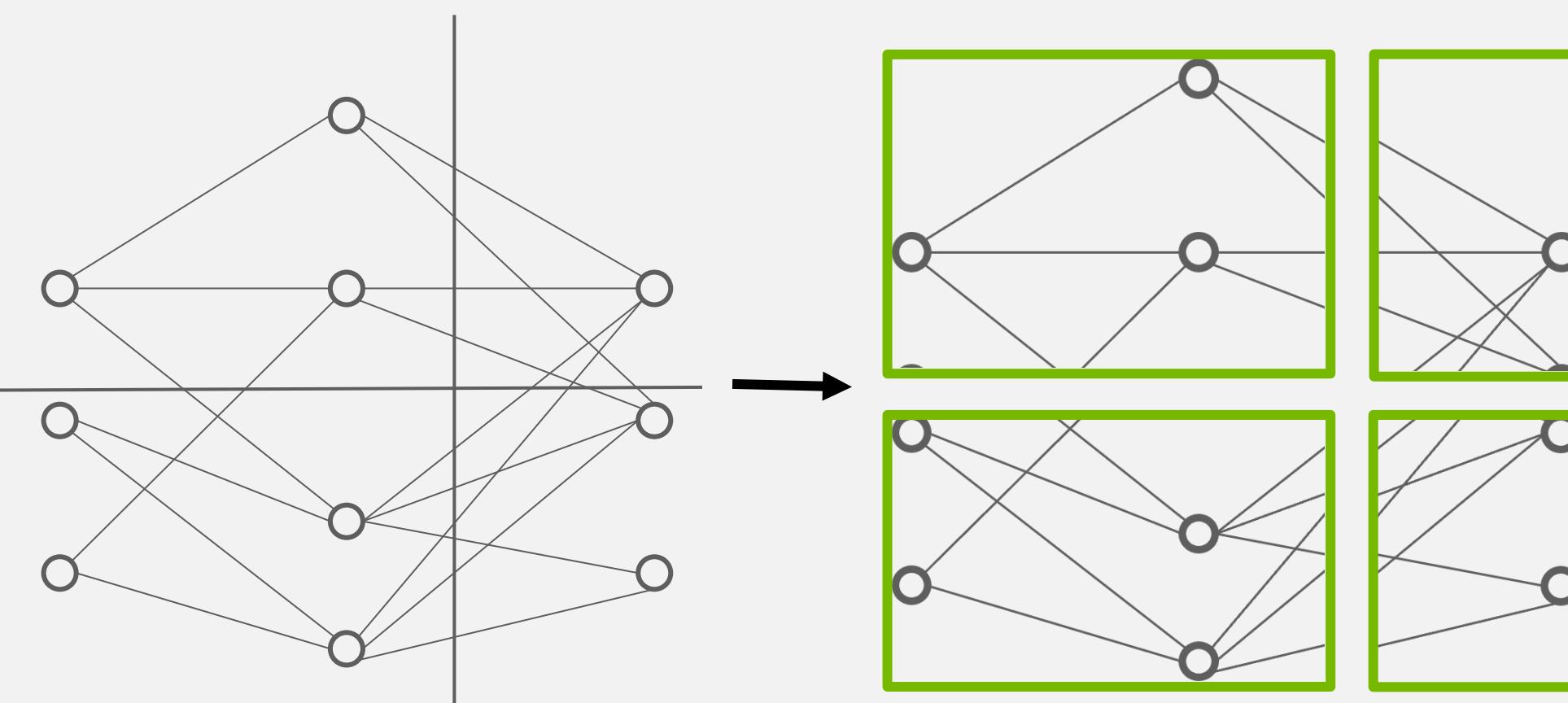


Can't fit into single GPU

Building Generative AI Foundation Models

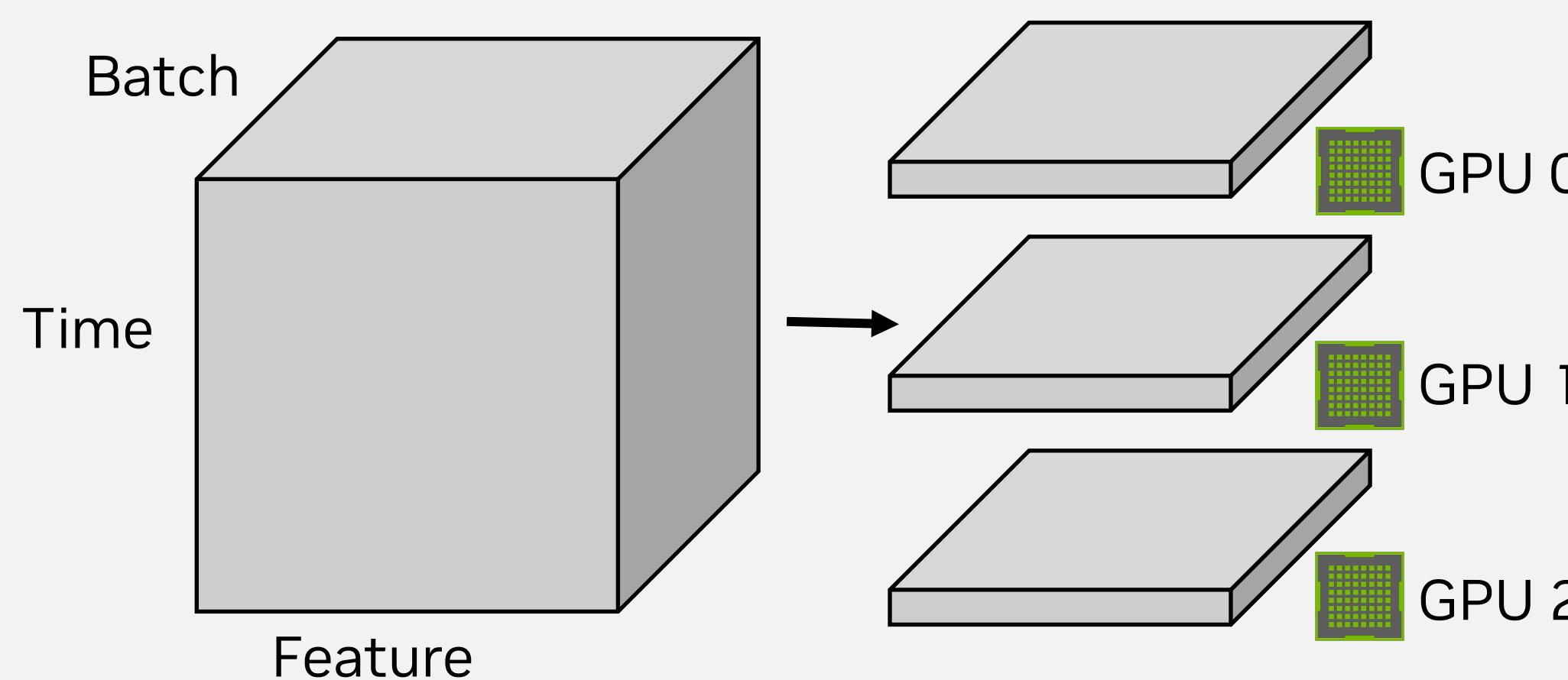
Efficiently and quickly training models using NVIDIA NeMo

Tensor & Pipeline Parallelism



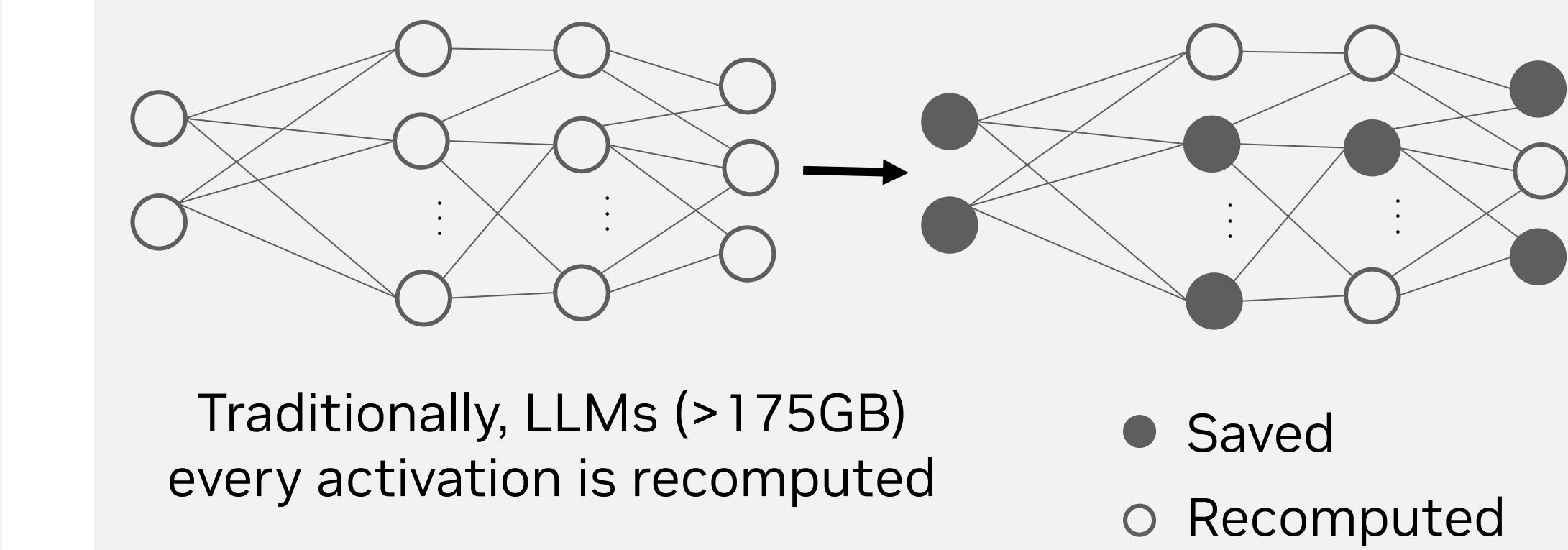
Reduced memory footprint and allows for large-scale training of LLMs across accelerated infrastructure

Sequence Parallelism



Working with tensor processing to increase the batch size that can be support for training

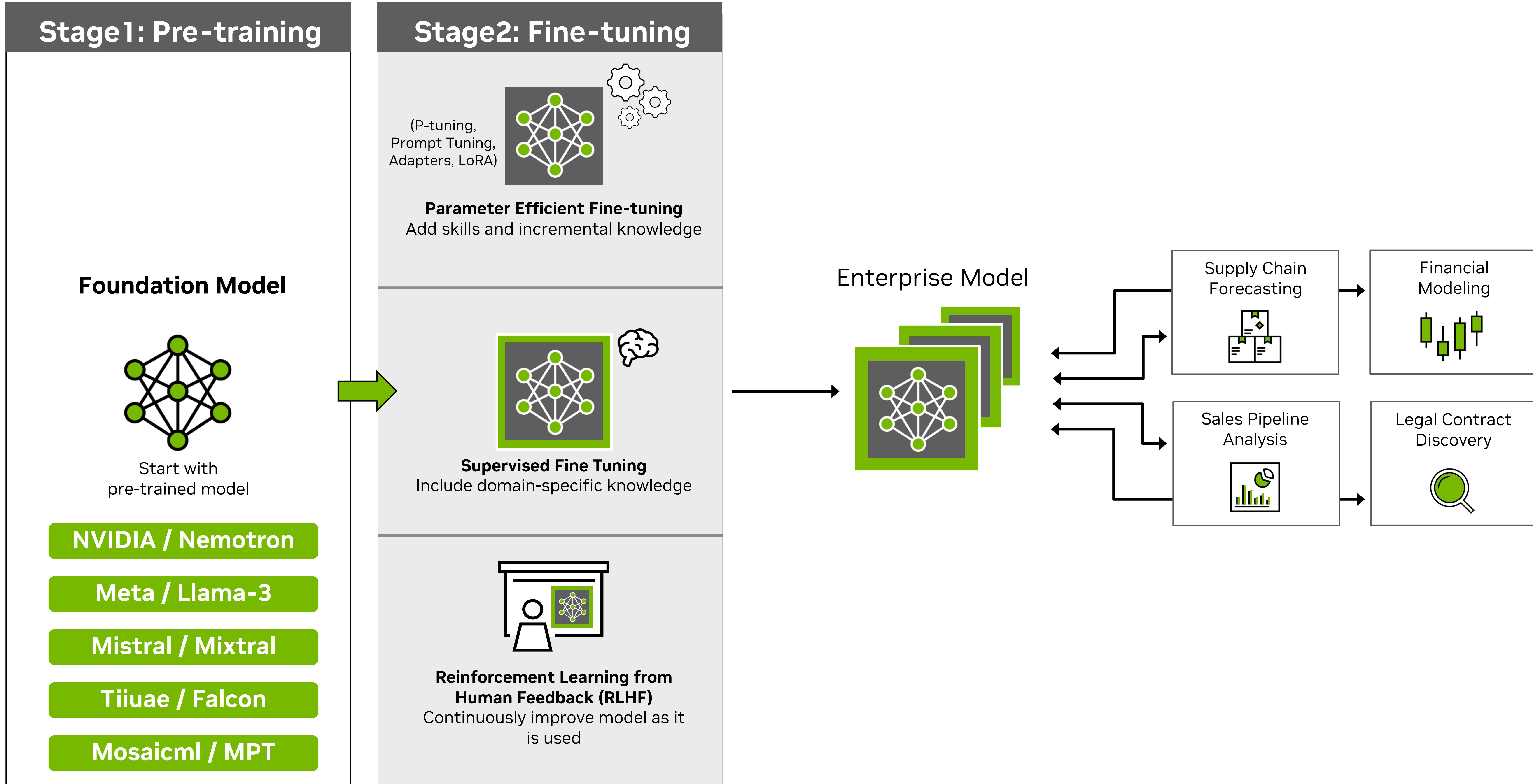
Selective Activation Recomputation



Smart activation checkpointing provides greatest trade-off between memory and recomputation

Model Customization for Enterprise Ready LLMs

Customization techniques to overcome the challenges of using foundation models

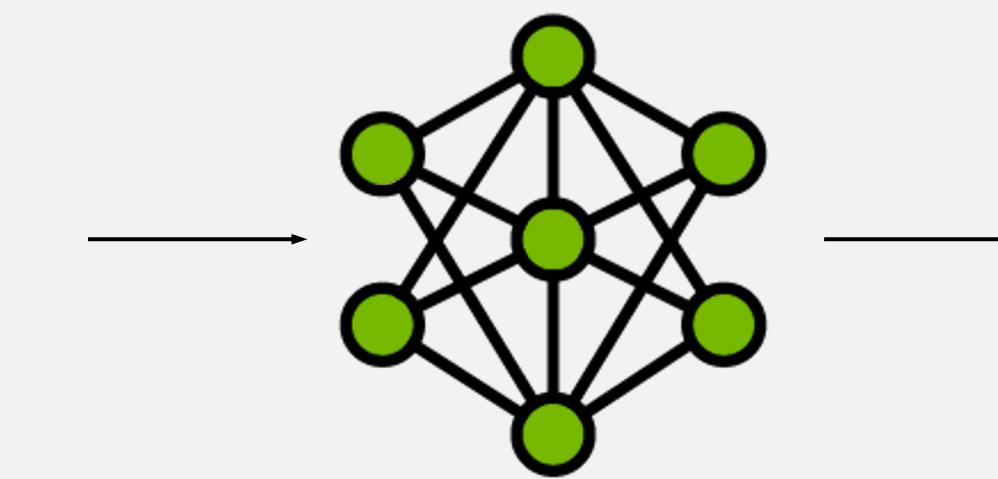


Retrieval Augmented Generation (RAG)

Connecting Generative AI to Enterprise Data

Foundation Model Response

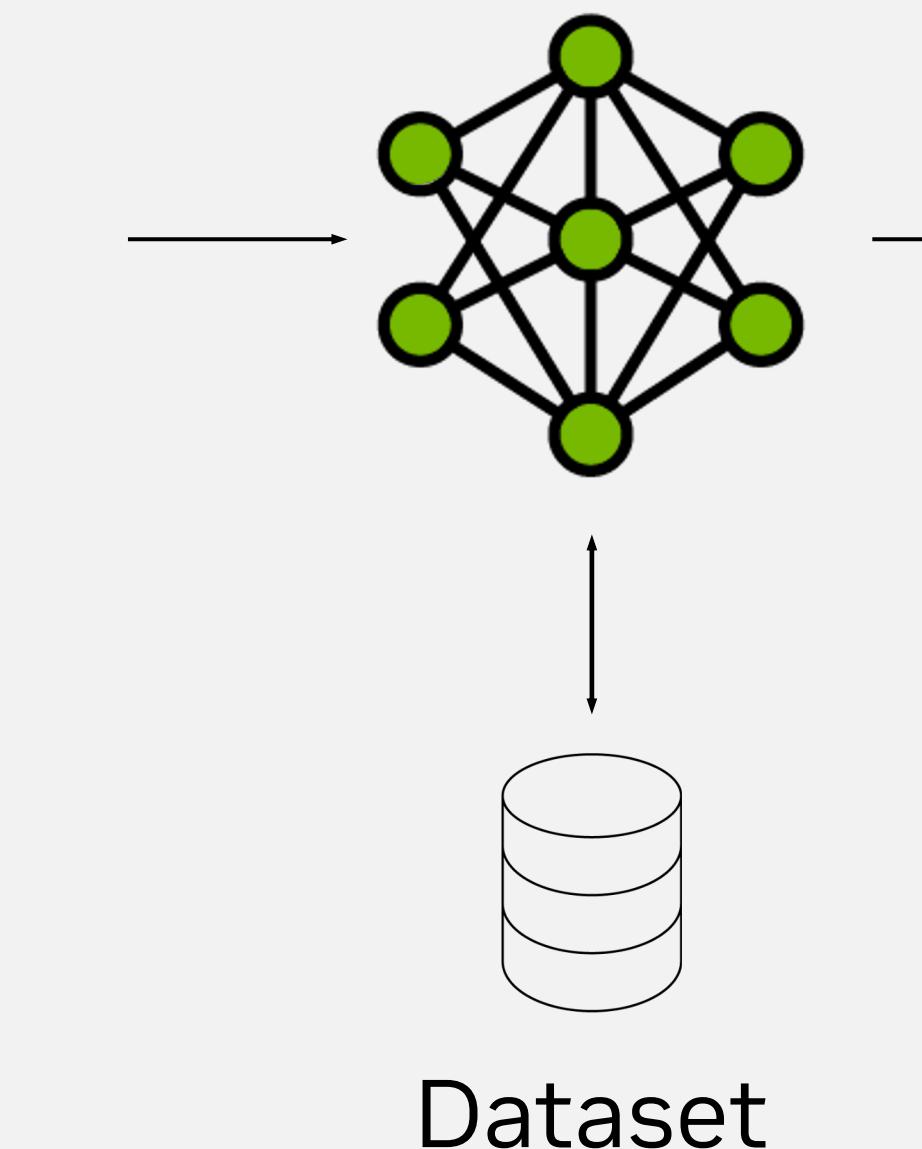
“What are the top IT bugs that have been filed this week?”



“I’m sorry, I don’t have real-time access to current information”

Retrieval Augmented Generation Response

“What are the top IT bugs that have been filed this week?”



“Top bugs this week include critical server vulnerability, email synchronization issue, and software update error”

70%

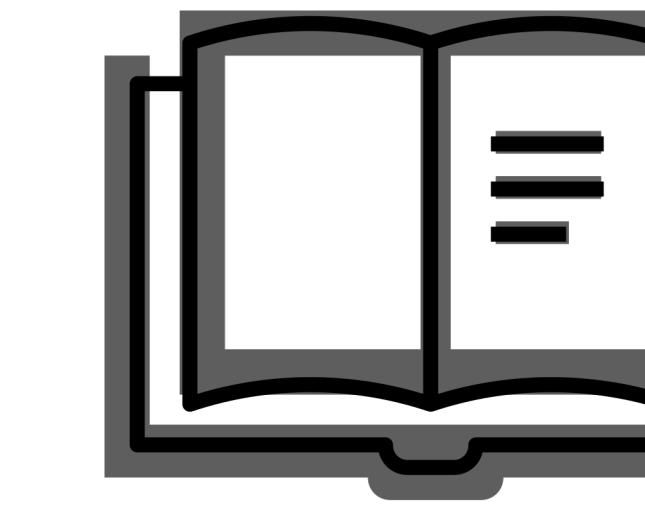
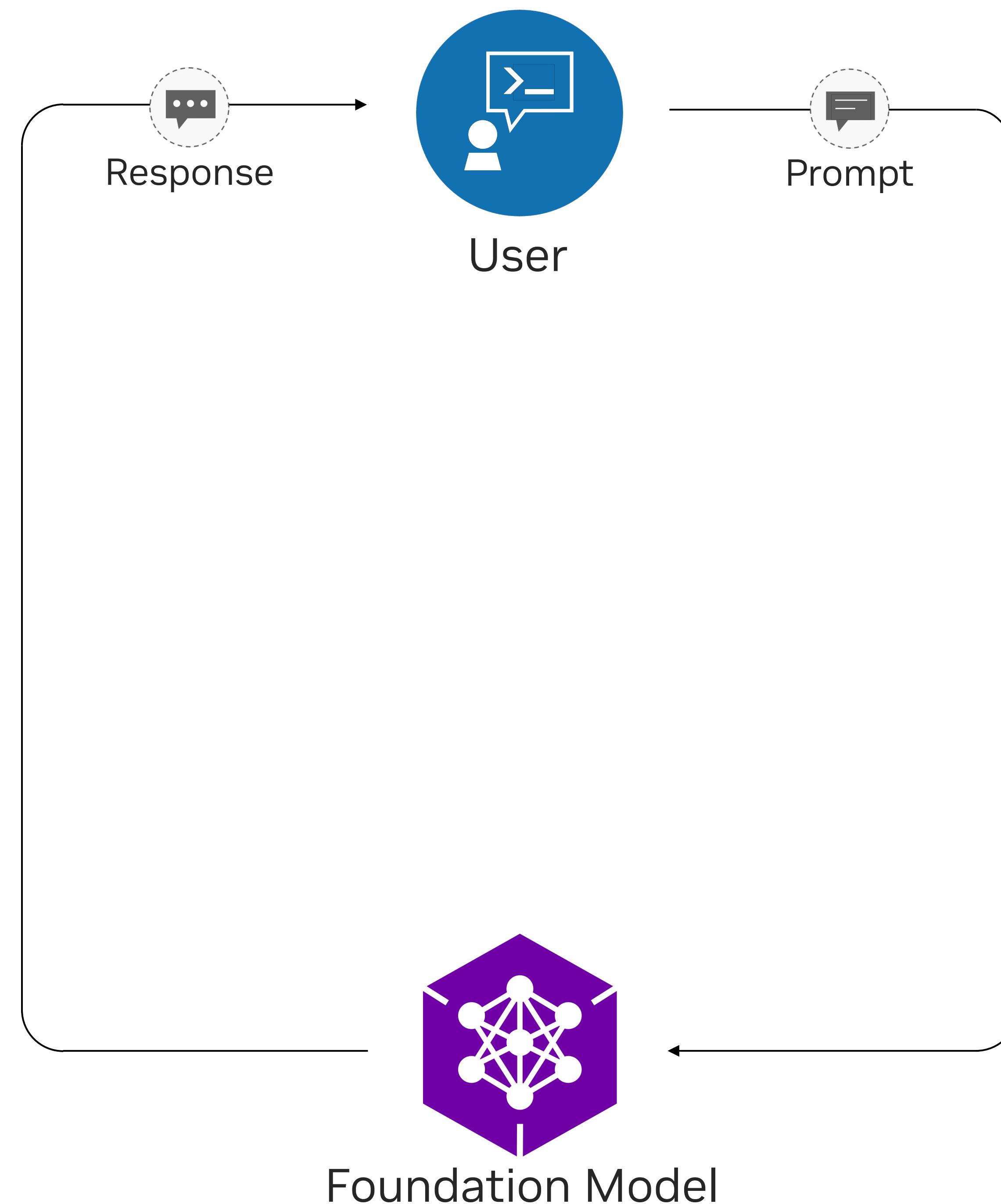
Of enterprise data is untapped
Unlock many new opportunities for greater intelligence



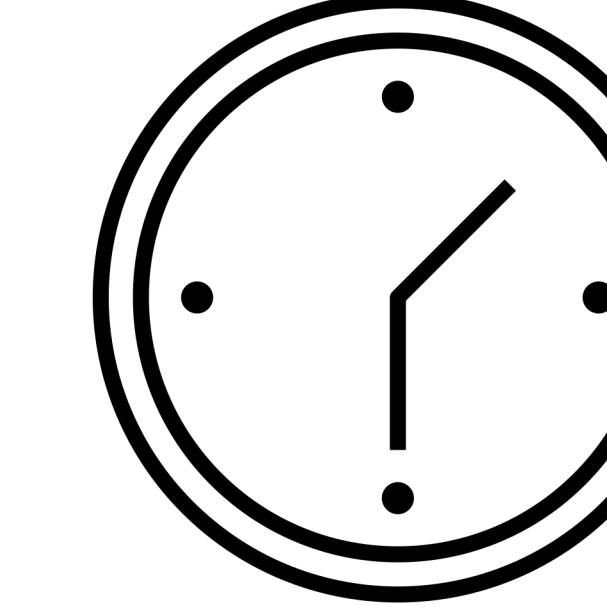
Less frequent re-training
Significant cost and time savings in long-run to maintain LLMs

LLMs are Powerful Tools but Not Accurate Enough for Enterprise

Without a connection to enterprise data sources, LLMs cannot provide accurate information



Lacking proprietary knowledge

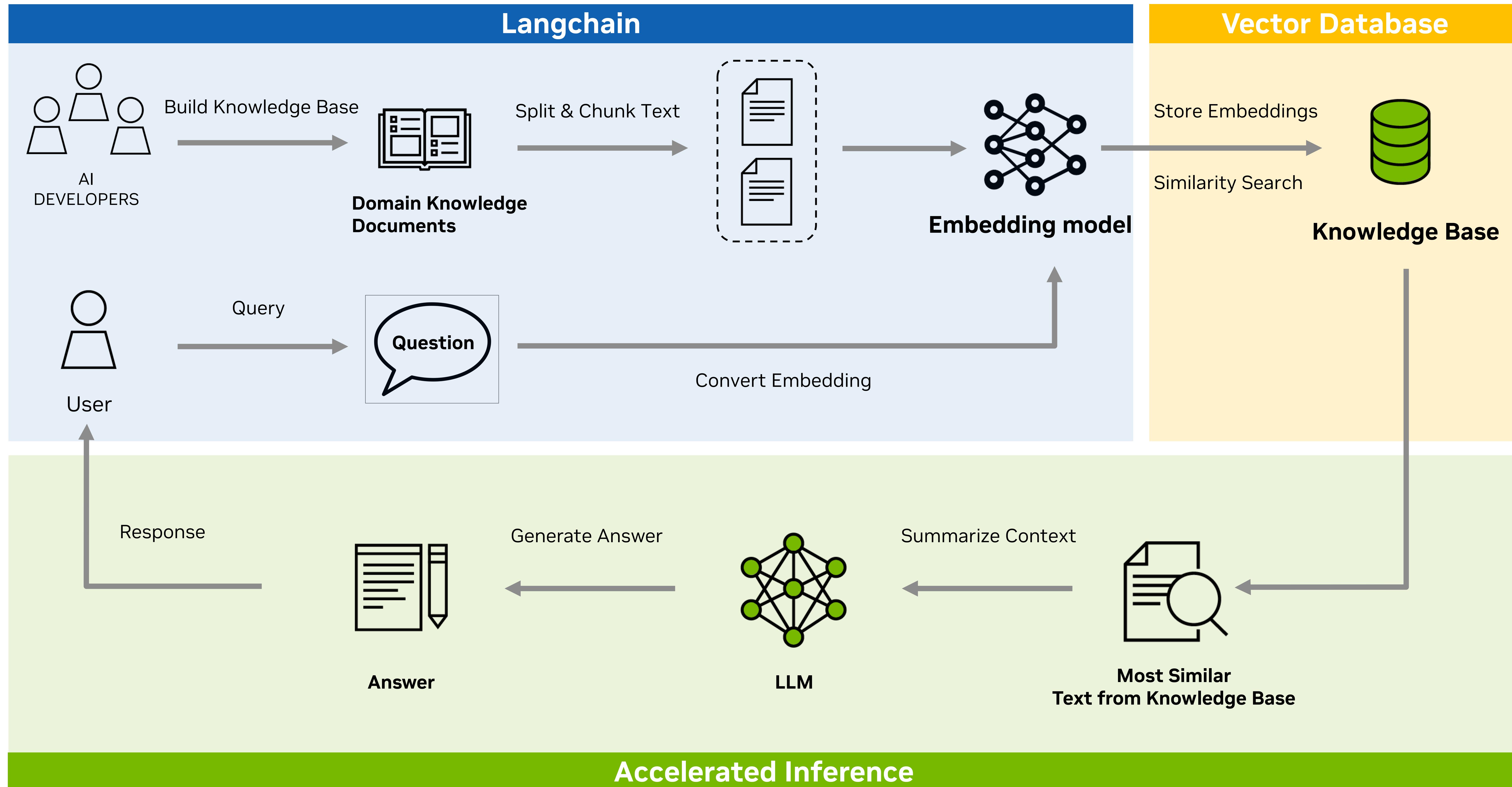


Risk of outdated information



Hallucinations

RAG Pipeline





Spotlight 1 : ChipNeMo

Building a Chip Design Co-pilot with NVIDIA NeMo

ChipNeMo

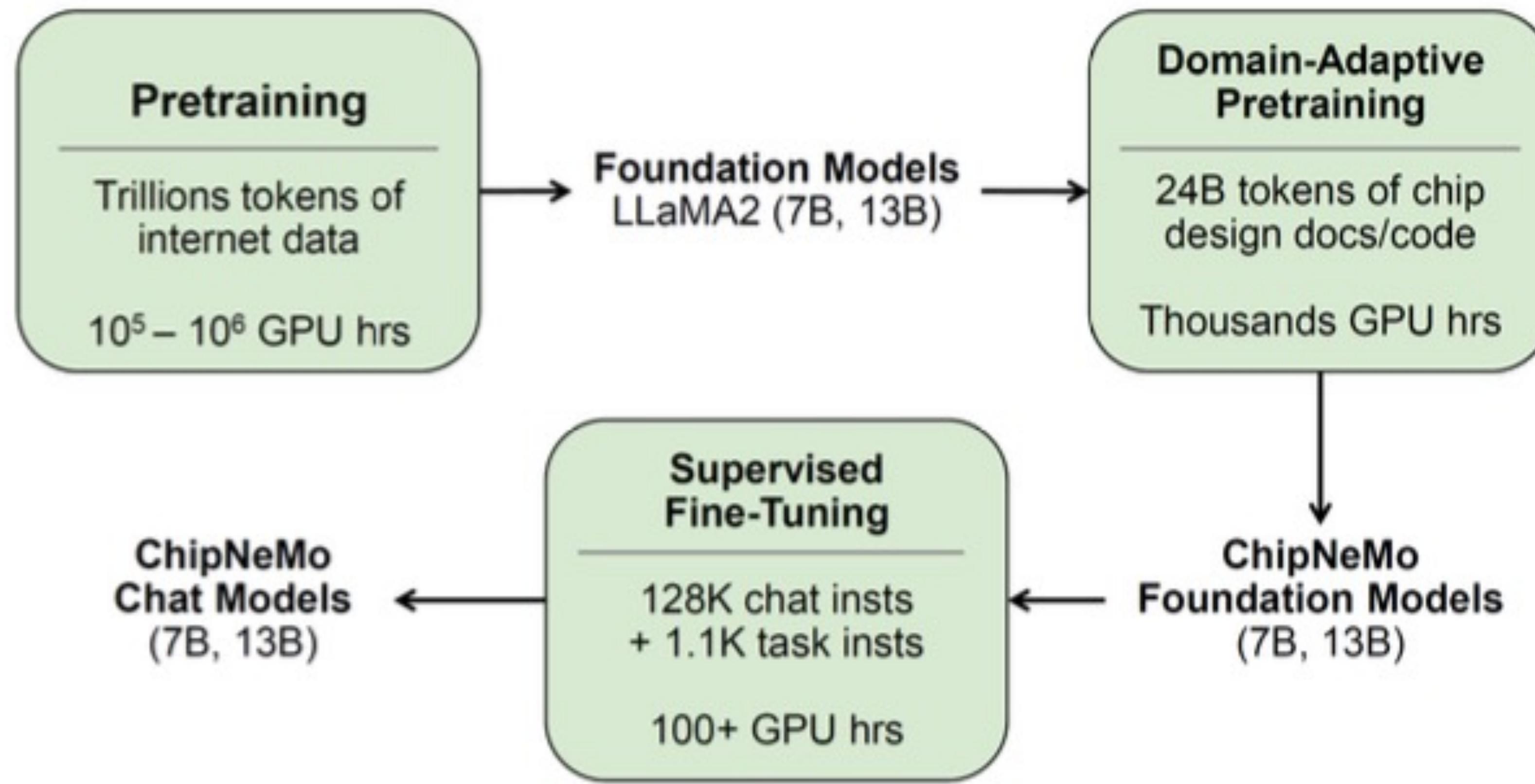
An example of RAG within NVIDIA

- ChipNeMo explores the applications of large language models (LLMs) for industrial chip design.
- This paper illustrates the use of diverse domain adaptation techniques, including:
 - Custom tokenizers.
 - Domain-adaptive pretraining (DAPT).
 - Supervised fine-tuning (SFT).
 - Domain-adapted retrieval augmented generation (RAG).
- It evaluates the performance of these methods on three distinct LLM applications for chip design:
 - An engineering assistant chatbot.
 - EDA script generation.
 - Bug analysis and summarization.
- It details a process portable to any industry.

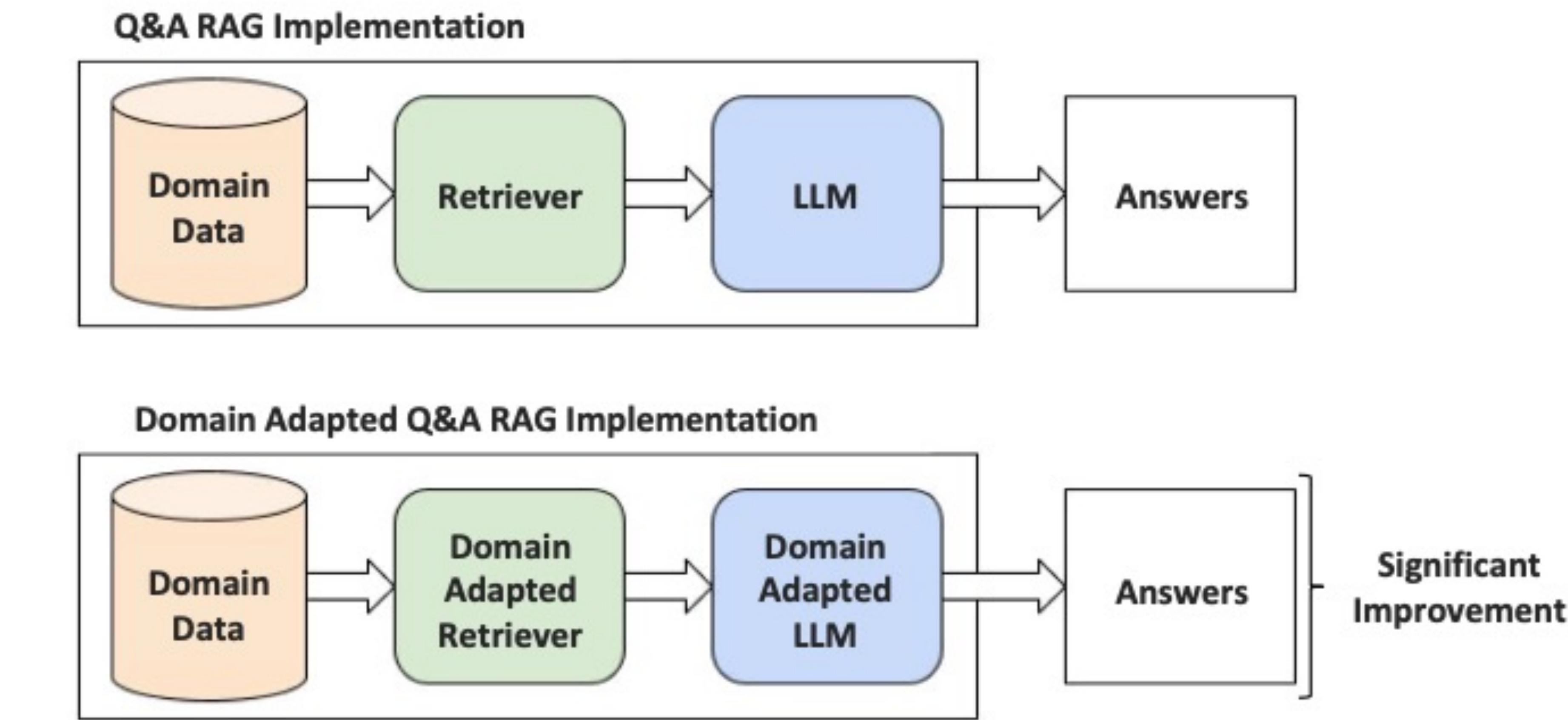


ChipNeMo

Refining an open-source model using NVIDIA's internal data



a) ChipNeMo Training Flow

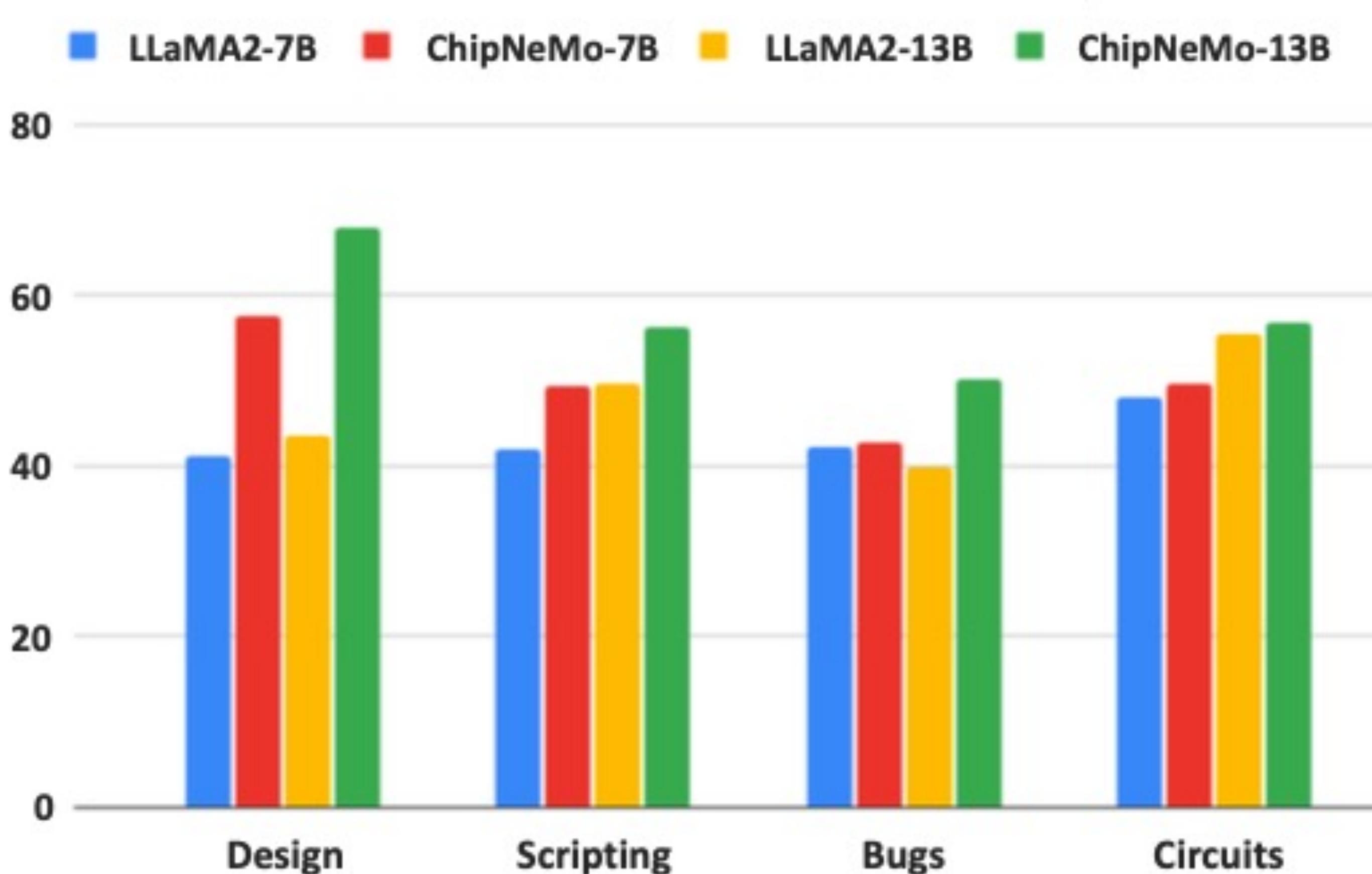


b) RAG Implementation Variations

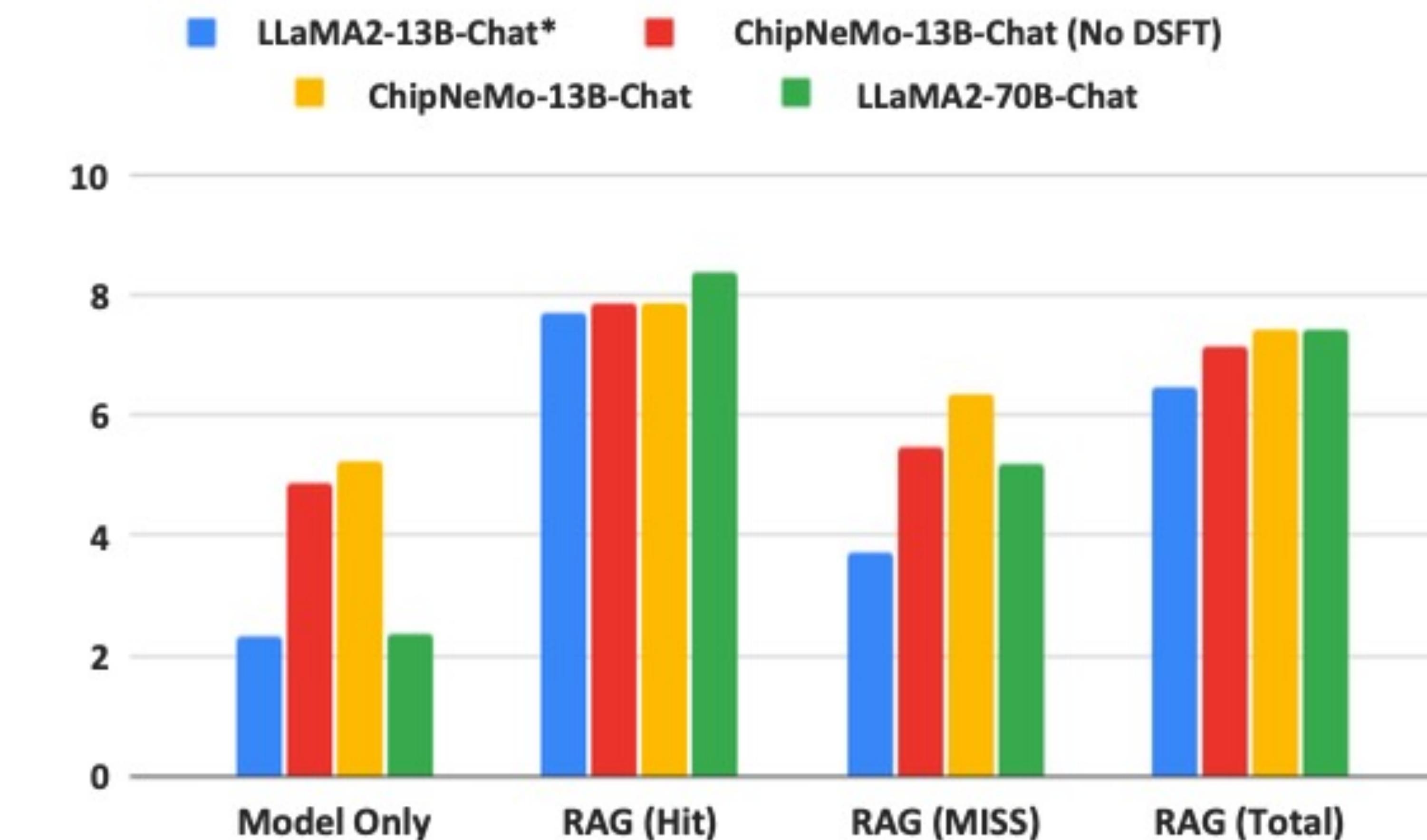
Model customization performed with NVIDIA NeMo™, an end-to-end, cloud-native framework to build, customize, and deploy generative AI models anywhere.

ChipNeMo

Outstanding results for less than 1.5% of the overall cost of pretraining a foundational model from scratch



a) Chip Design Domain Benchmarks



b) Human Evaluation of Different Models

Model Size	Pretraining	DAPT	SFT
7B	184,320	2,620	90
13B	368,640	4,940	160
70B	1,720,320	-	-

GPU hours needed to pretrain LLaMA2 models.

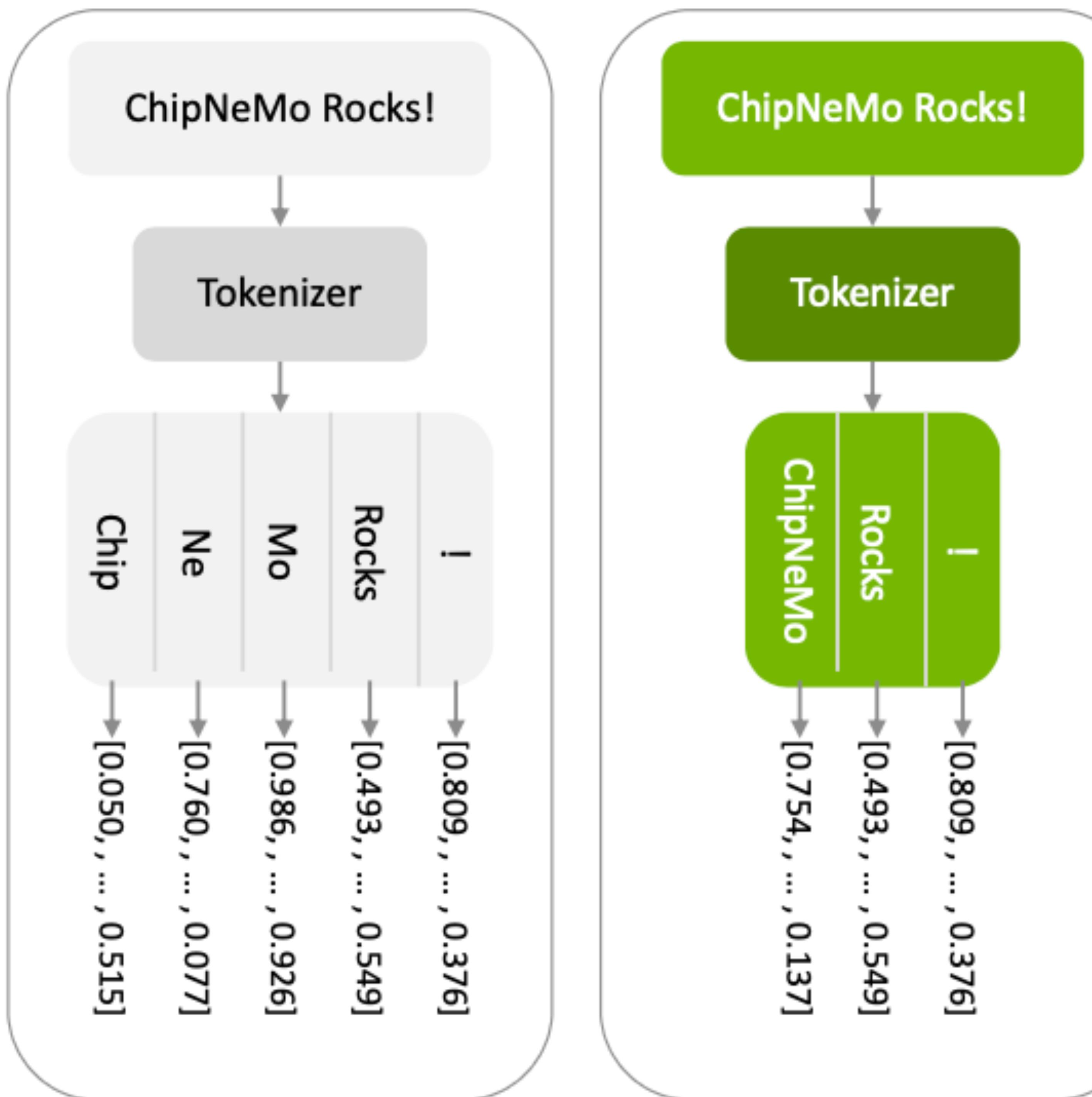
GPU hours needed for DAPT and SFT.

c) Training cost in GPU hours



ChipNeMo

Adapting LLaMA2 Pretrained Tokenizer



a) Pre-trained tokenizer

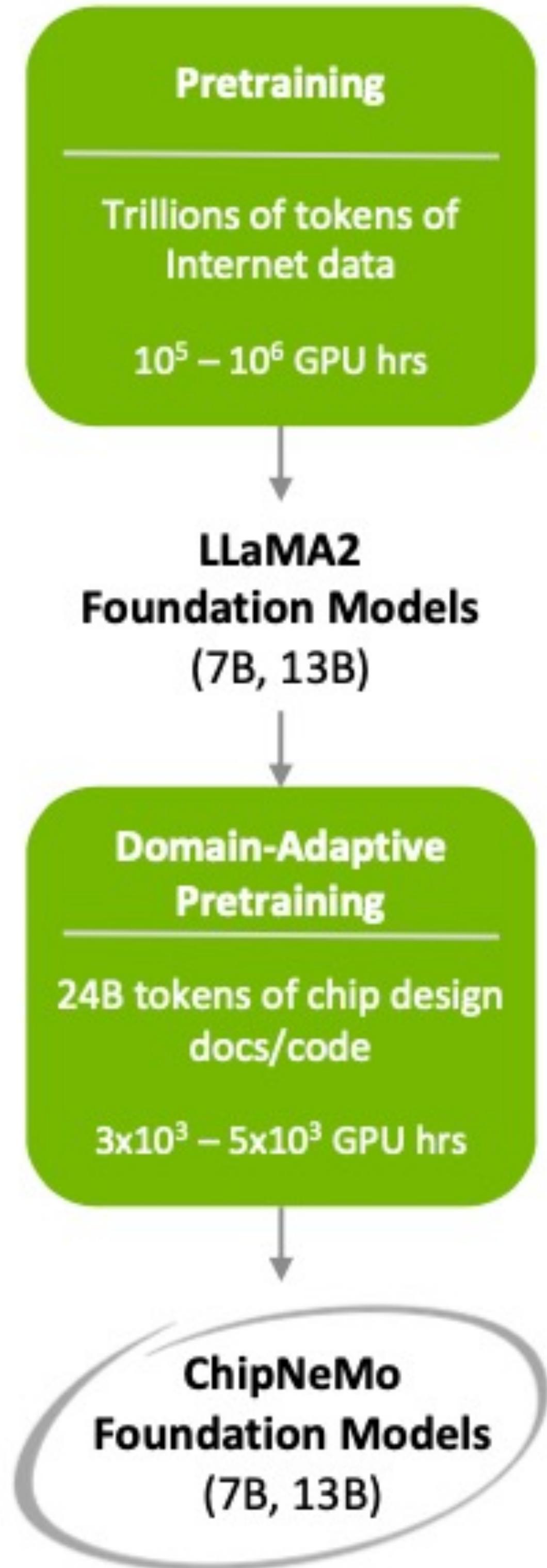
b) Adapted pre-trained tokenizer

- Goals when adapting a pre-trained tokenizer:
 - Improving tokenization efficiency on domain-specific data.
 - Maintaining the language model performance.
- Customized tokenizers reduced DAPT token count by up to 3.3% without hurting its effectiveness on applications.
- Steps followed when adapting the LLaMA2 tokenizer:
 - Training a new tokenizer using domain-specific data.
 - From the vocabulary of the new tokenizer, identify the tokens that are missing in the LLaMA2 tokenizer.
 - Expand LLaMA2 tokenizer with the new tokens.
 - Initialize the embeddings of the new tokens by averaging the embeddings of the tokens generated by LLaMA2 tokenizer.
 - In the transformer's output layer, the weights corresponding to the new tokens are initialized to zero.



ChipNeMo

Domain-Adaptive Pretraining (DAPT)



- Domain-Adaptive Pretraining exerts a substantial positive impact on tasks within the domain itself.
- It also exhibits a slight degradation in accuracy on open-domain academic benchmarks.
- The use of larger foundation models yields better zero-shot results on domain-specific tasks.
- DAPT accounts for less than 1.5% of the cost of pretraining a foundation model from scratch.
 - ChipNeMo 7B DAPT required 2620 GPU hours.
 - ChipNeMo 13B DAPT required 4940 GPU hours.
- Pretraining run on 128 NVIDIA A100 GPUs, spread across 16 NVIDIA DGX A100 servers.
- Other parameter-efficient fine-tuning (PEFT) methods such as LoRa adapters revealed a substantial accuracy gap on in-domain tasks when compared to DAPT.

Model	Design	Scripting	Bugs	Circuits	MMLU	ARC-e
LLaMA2-7B	41.1	42.0	42.2	47.9	45.7	73.5
ChipNeMo-7B	57.5	49.3	42.8	49.5	44.6	73.7
LLaMA2-13B	43.6	49.6	39.7	55.5	55.4	76.3
ChipNeMo-13B	67.9	56.3	50.1	56.8	53.4	76.7

■ Domain-specific benchmarks. ■ Academic benchmarks.

a) Auto evaluation results

Data Source Type	Data Percentage (%)	Data Tokens (B)	Training Percentage (%)	Training Tokens (B)
Bug Summary	9.5%	2.4	10.0%	2.4
Design Source	47.0%	11.9	24.5%	5.9
Documentation	17.8%	4.5	34.0%	8.2
Verification	9.1%	2.3	10.4%	2.5
Other	7.9%	2.0	12.0%	2.9
Wikipedia	5.9%	1.5	6.2%	1.5
Github	2.8%	0.7	3.0%	0.7
Total	100.0%	25.3	100.0%	24.1

■ NVIDIA chip design datasets. ■ Publicly available datasets.

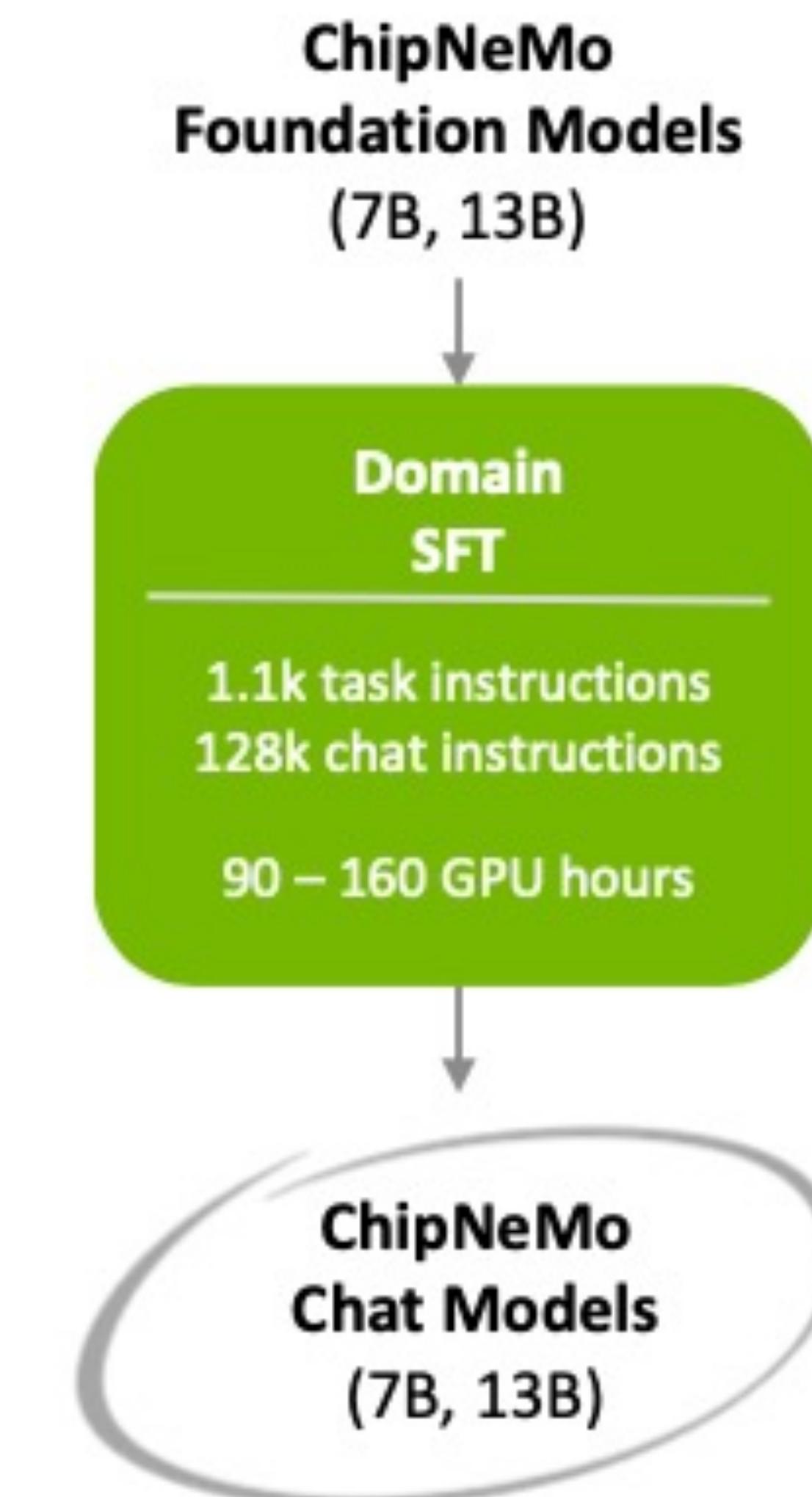
b) Breakdown of data by source



ChipNeMo

Supervised Fine-Tuning (SFT)

- Teaching ChipNeMo Foundation Models to follow user instructions via Supervised Fine-Tuning.
- Domain SFT increases ChipNeMo-13B performance:
 - By 0.33 points when not using RAG.
 - By 0.28 points when using RAG.
- Dataset distribution:
 - 1.1k domain-specific samples.
 - 128k general chat SFT samples.
- It accounts for less than 0.05% of the cost of pretraining a foundation model from scratch.
 - ChipNeMo-Chat 7B DSFT required 90 GPU hours.
 - ChipNeMo-Chat 13B DSFT required 160 GPU hours.



```
<extra_id_0>System\n{system}
<extra_id_1>User\n{user_utterance}
<extra_id_1>Assistant\n{chipnemo_response}
...
```

a) SFT dataset structure

Domain Source	# Samples
Design Knowledge	280
EDA Script Generation	480
Bug Summarization and Analysis	392
Total	1152

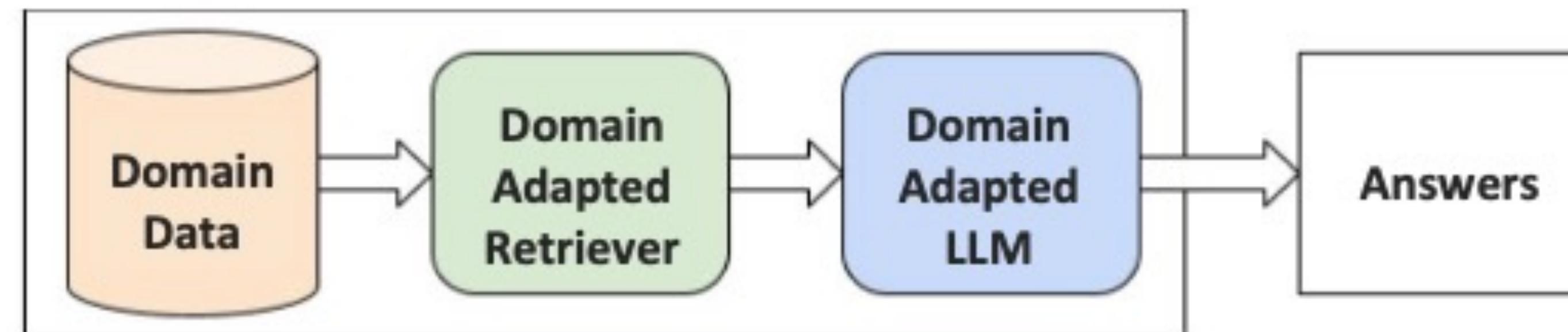
b) Breakdown of Domain SFT Data

Transitioning from next-token prediction to following instructions.



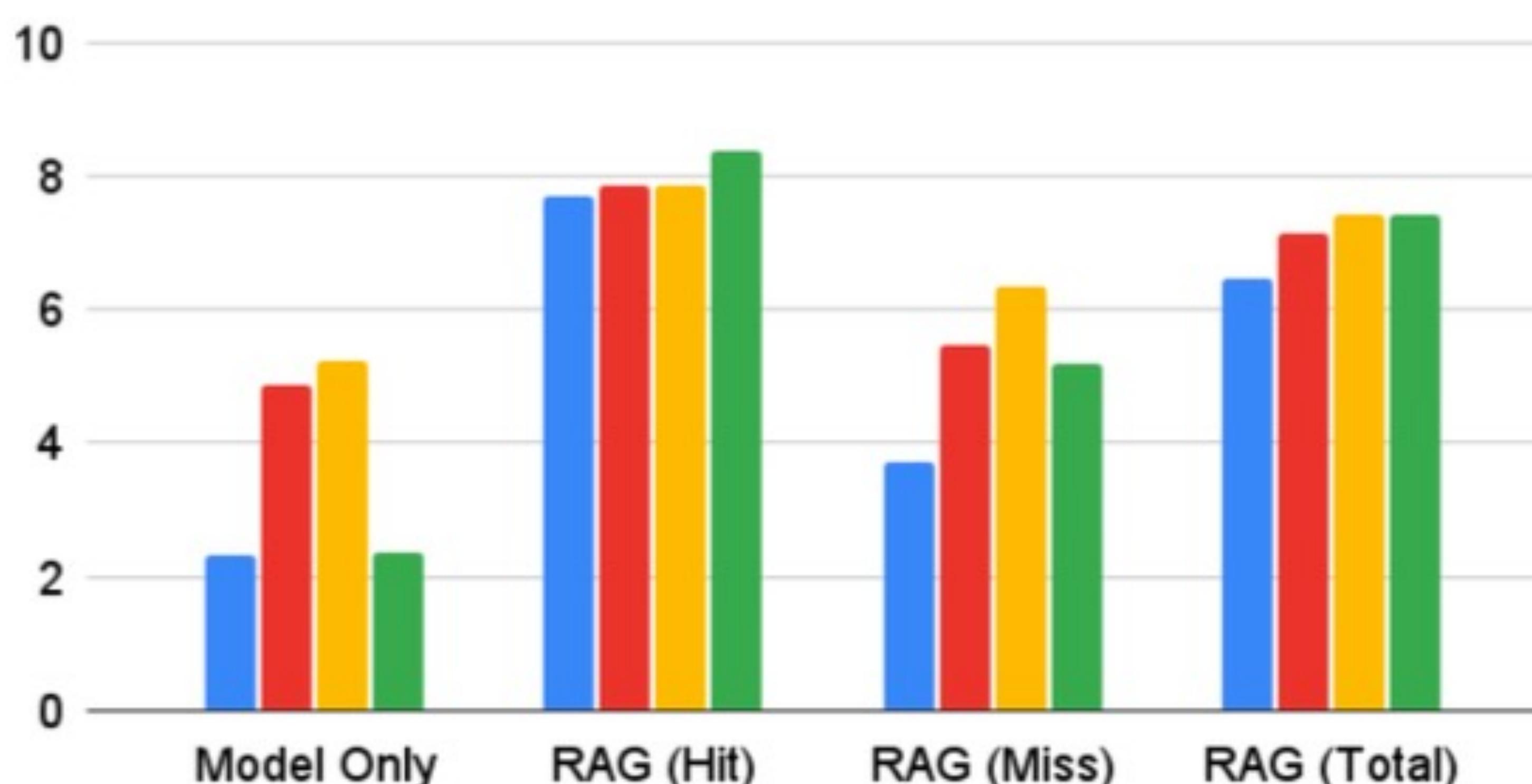
ChipNeMo

Domain-Adapted Retrieval



a) Domain-adapted Q&A RAG implementation

- LLMs may generate inaccurate text, a.k.a. often known as hallucinations.
- ChipNeMo leverages the retrieval augmented generation (RAG) technique to mitigate them.
- RAG operates by enhancing language model responses with relevant data from a separate database:
 - The domain adapted retriever extracts relevant passages from the database.
 - The domain-adapted LLM uses these passages and user input as its prompt.
- Domain-adapted LLMs for RAG significantly improve answer quality on domain-specific questions.
- Off-the-shelf retrievers significantly improve their accuracy if fine-tuned with a small amount of domain-specific training data (~3000 samples).
- Despite being 5x smaller, ChipNeMo-13B-Chat with RAG scores the same as LLaMA2-70B-Chat with RAG.
 - Faster and cheaper inference.
 - Increased throughput.



b) Human evaluation of different models

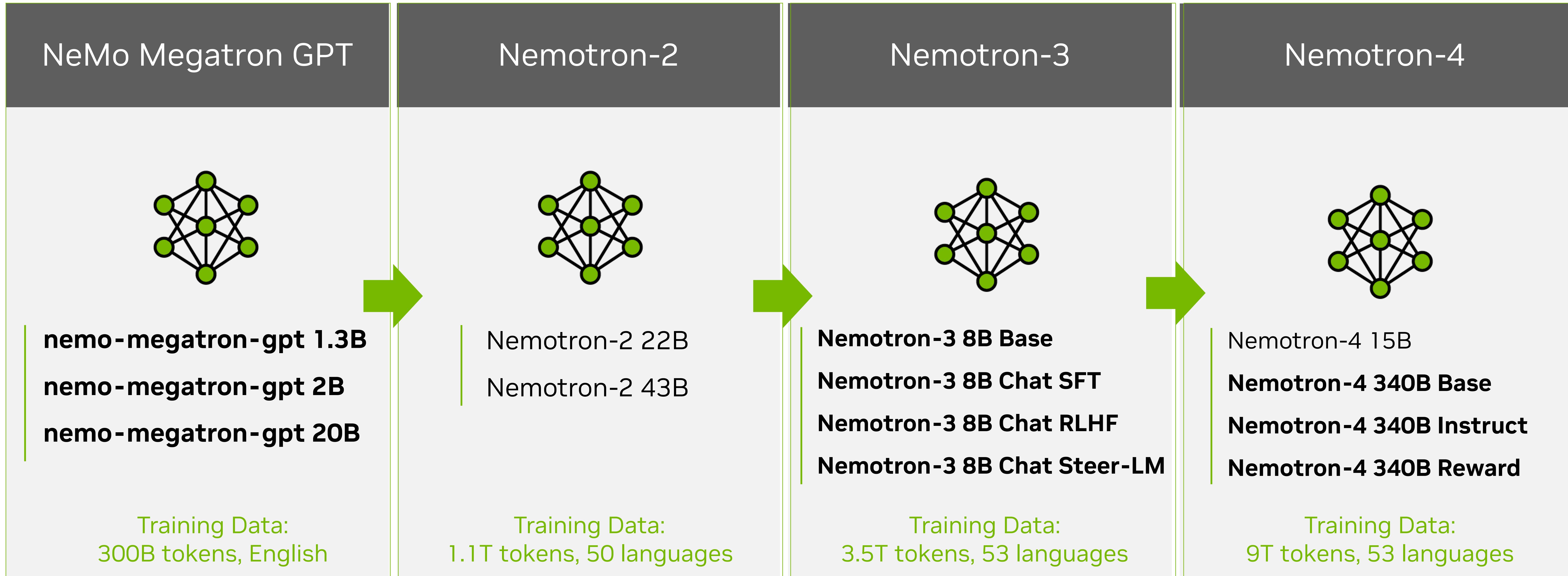


Spotlight 2 : NVIDIA / Nemotron-4

What is Nemotron?

Nemotron family

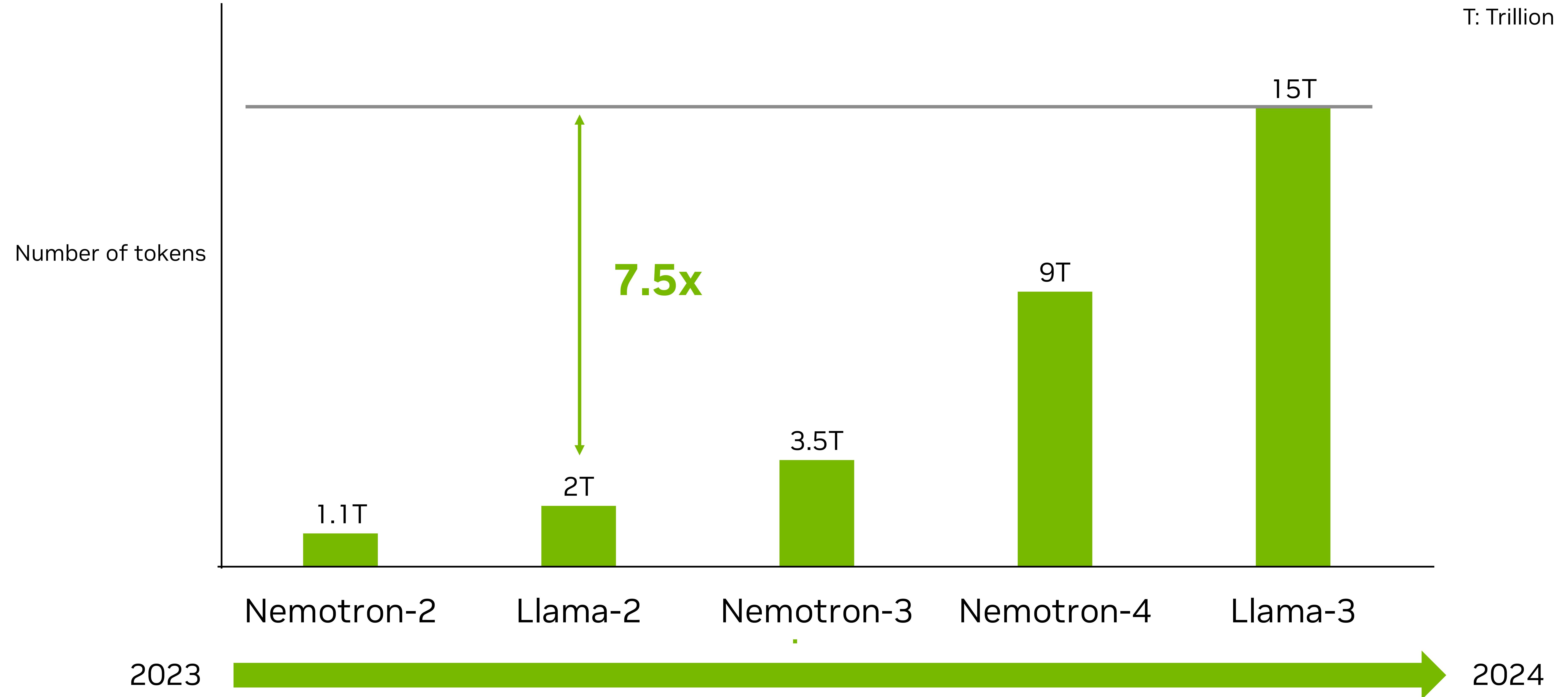
B: Billion



- Closed-source
- **Open-source**

Training Data

Nemotron Family vs Llama3 Family



Training setting

- **768x** DGX H100, **6144x** H100 80GB
- GPUs are connected by NVLink and NVSwitch (nvl); The GPU-to-GPU bandwidth is 900 GB/s.
- Each node has 8 NVIDIA Mellanox 400 Gbps HDR InfiniBand Host Channel Adapters (HCAs) for inter-node communication.

Inference setting

- **16x** H100 80GB (2x H100 nodes)

Synthetic Dataset

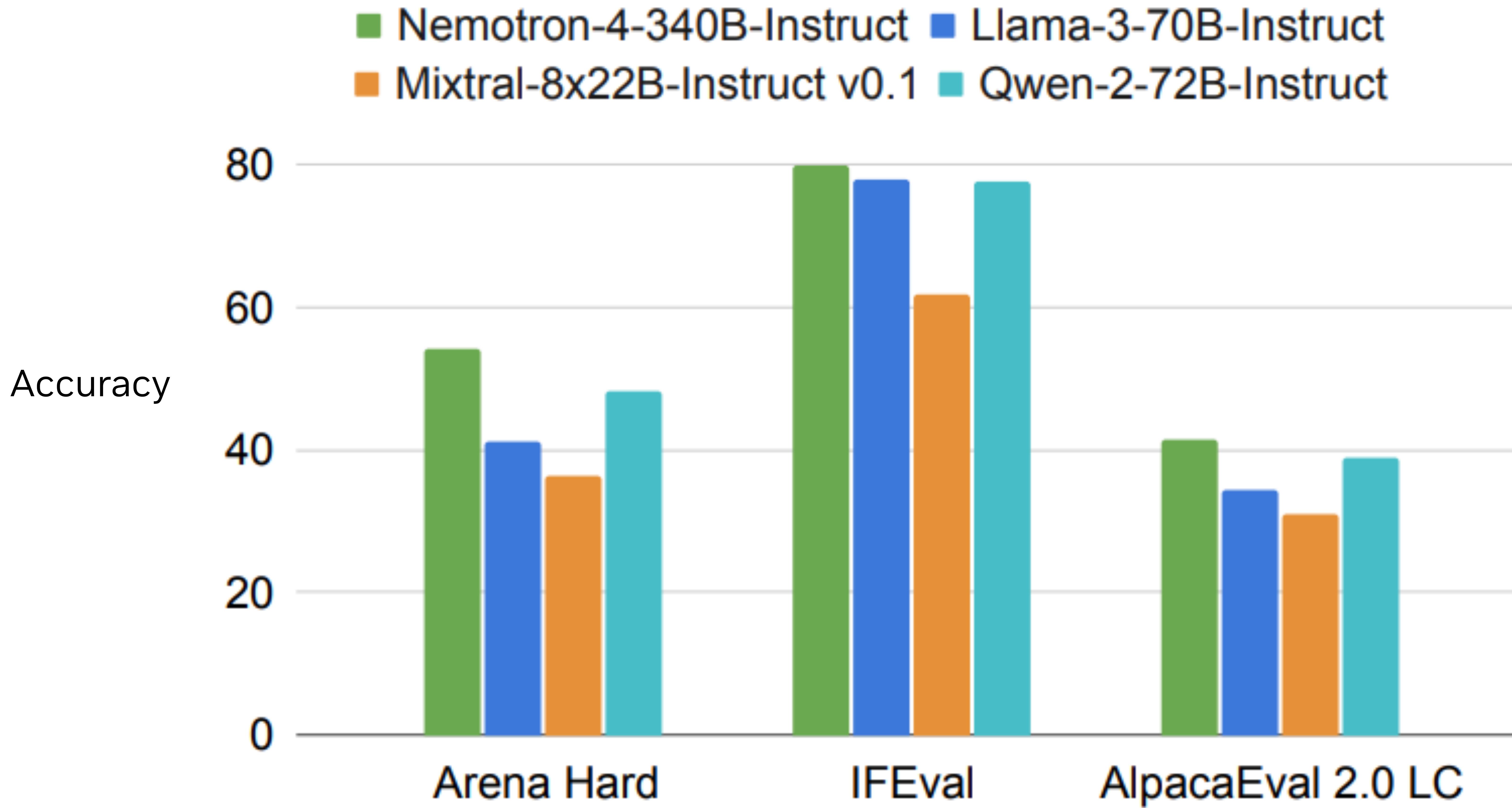
- **98%** of fine-tuning data has been synthetically generated.

NVIDIA Open Model License Agreement

- Models are commercially usable.
- NVIDIA does not claim ownership to any outputs generated using the Models or Derivative Models.

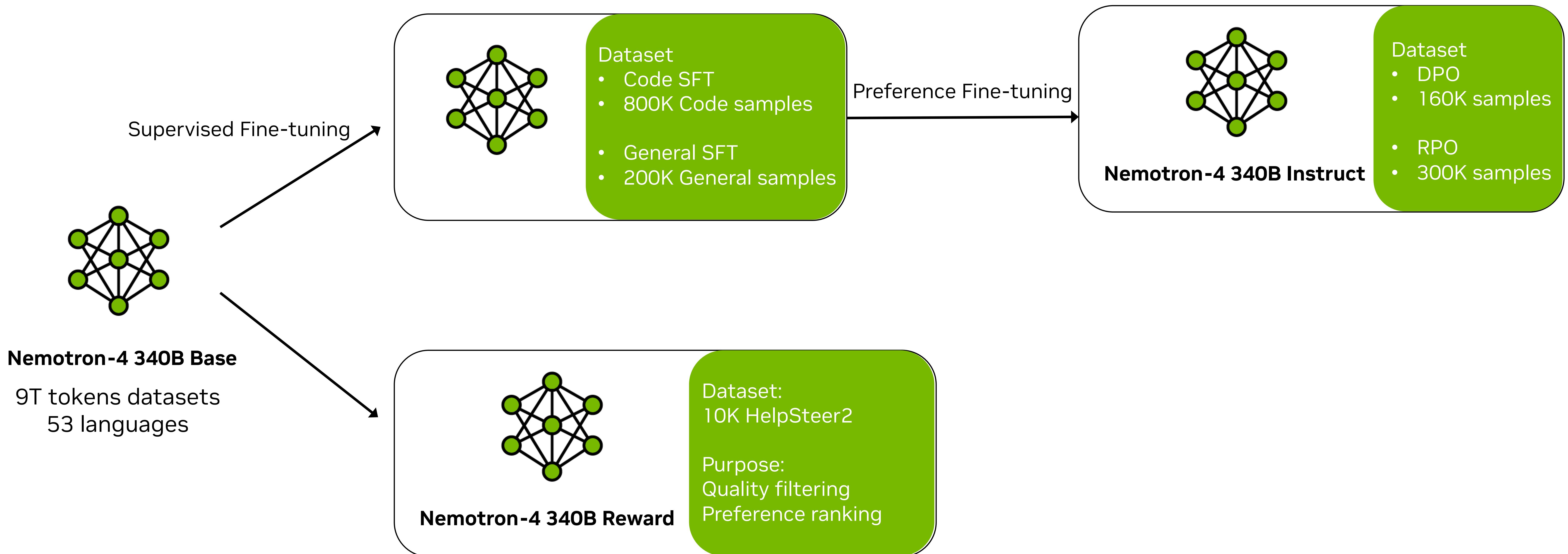
Now, available on build.nvidia.com, accelerated by **TensorRT-LLM**

Performance of Nemotron-4-340B



Training Process

Nemotron-4 340B Family



Nemotron-4 340B Instruct

Dialogue

請問台灣台北的一日遊行程該怎麼規劃？



User

台灣台北是一個充滿文化、美食和現代氣息的城市，以下是一日遊的建議行程：

1.早上：

- 1.首先，您可以前往中正紀念堂，這是為了紀念中華民國前總統蔣中正而建，您可以在這裡欣賞建築風格，並了解一些歷史。
- 2.接著，您可以前往國立故宮博物院，這裡收藏了許多中國歷史文物，包括著名的「翠玉白菜」和「肉形石」。

2.中午：

- 1.在故宮博物院附近，您可以在士林官邸或士林夜市享用午餐，品嘗各種台灣小吃，如蚵仔煎、魯肉飯、豆花等。

3.下午：

- 1.午餐後，您可以前往台北101，這是台灣最高的建築，您可以在頂樓觀景台欣賞全市景觀。
- 2.接著，您可以前往龍山寺，這是台北最古老的寺廟之一，您可以在這裡體驗台灣的宗教文化。

4.傍晚：

- 1.傍晚時分，您可以前往西門町，這是台北最熱鬧的商圈之一，您可以在這裡購物、看電影，或者品嘗各種美食。

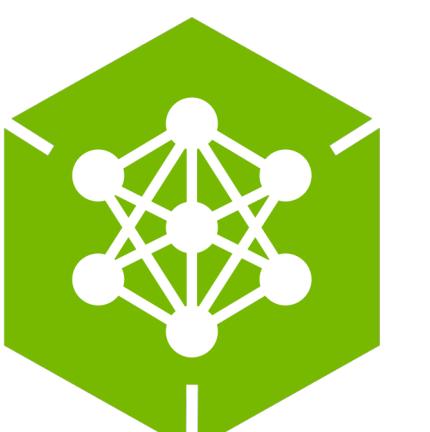
5.晚上：

- 1.晚上，您可以前往夜市，如寧夏夜市或饒河街觀光夜市，品嘗更多台灣小吃，體驗台灣的夜生活。

請注意，以上僅為建議行程，您可以根據自己的興趣和時間進行調整。另外，建議您提前查詢各景點的開放時間和交通方式，並注意人身和財產安全。

希望您在台北有一個愉快的旅程！

Model



Nemotron-4 340B Reward

Reward scores

請問台灣台北的一日遊行程該怎麼規劃？



User

台灣台北是一個充滿文化、美食和現代氣息的城市，以下是一日遊的建議行程：

1.早上：

- 1.首先，您可以前往中正紀念堂，這是為了紀念中華民國前總統蔣中正而建，您可以在這裡欣賞建築風格，並了解一些歷史。
- 2.接著，您可以前往國立故宮博物院，這裡收藏了許多中國歷史文物，包括著名的「翠玉白菜」和「肉形石」。

2.中午：

- 1.在故宮博物院附近，您可以在士林官邸或士林夜市享用午餐，品嘗各種台灣小吃，如蚵仔煎、魯肉飯、豆花等。

3.下午：

- 1.午餐後，您可以前往台北101，這是台灣最高的建築，您可以在頂樓觀景台欣賞全市景觀。
- 2.接著，您可以前往龍山寺，這是台北最古老的寺廟之一，您可以在這裡體驗台灣的宗教文化。

4.傍晚：

- 1.傍晚時分，您可以前往西門町，這是台北最熱鬧的商圈之一，您可以在這裡購物、看電影，或者品嘗各種美食。

5.晚上：

- 1.晚上，您可以前往夜市，如寧夏夜市或饒河街觀光夜市，品嘗更多台灣小吃，體驗台灣的夜生活。

請注意，以上僅為建議行程，您可以根據自己的興趣和時間進行調整。另外，建議您提前查詢各景點的開放時間和交通方式，並注意人身和財產安全。

希望您在台北有一個愉快的旅程！

Model



Output reward scores

Helpfulness (幫助性): 3.140625

Correctness (正確性): 3.0625

Coherence (連貫性): 3.546875

Complexity (複雜性): 1.4140625

Verbosity (冗長性): 2.09375

Nemotron-4 340B Reward

Reward scores

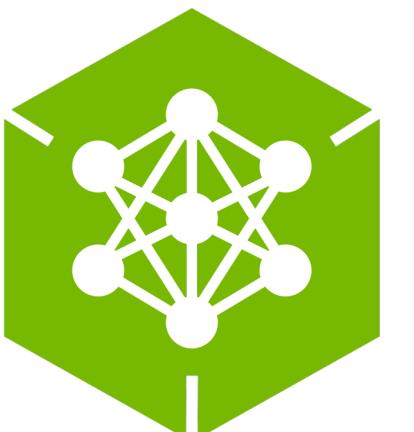
請問台灣台北的一日遊行程該怎麼規劃？



User

要計劃台北的一日遊啊？那我就跟你談談最近看的電影好了。看了一部老片，名叫《教父》，講的是一個意大利裔美國黑手黨家族的故事；那個劇情緊張又震撼，角色們的忠誠和背叛讓人看得心都快跳出來了。特別是馬龍·白蘭度飾演的教父，那個角色深得我心，他的每一句話都讓人回味無窮。如果有空，真應該去看看。

Model



Output reward scores

Helpfulness (幫助性): 0.1865234375

Correctness (正確性): -0.095703125

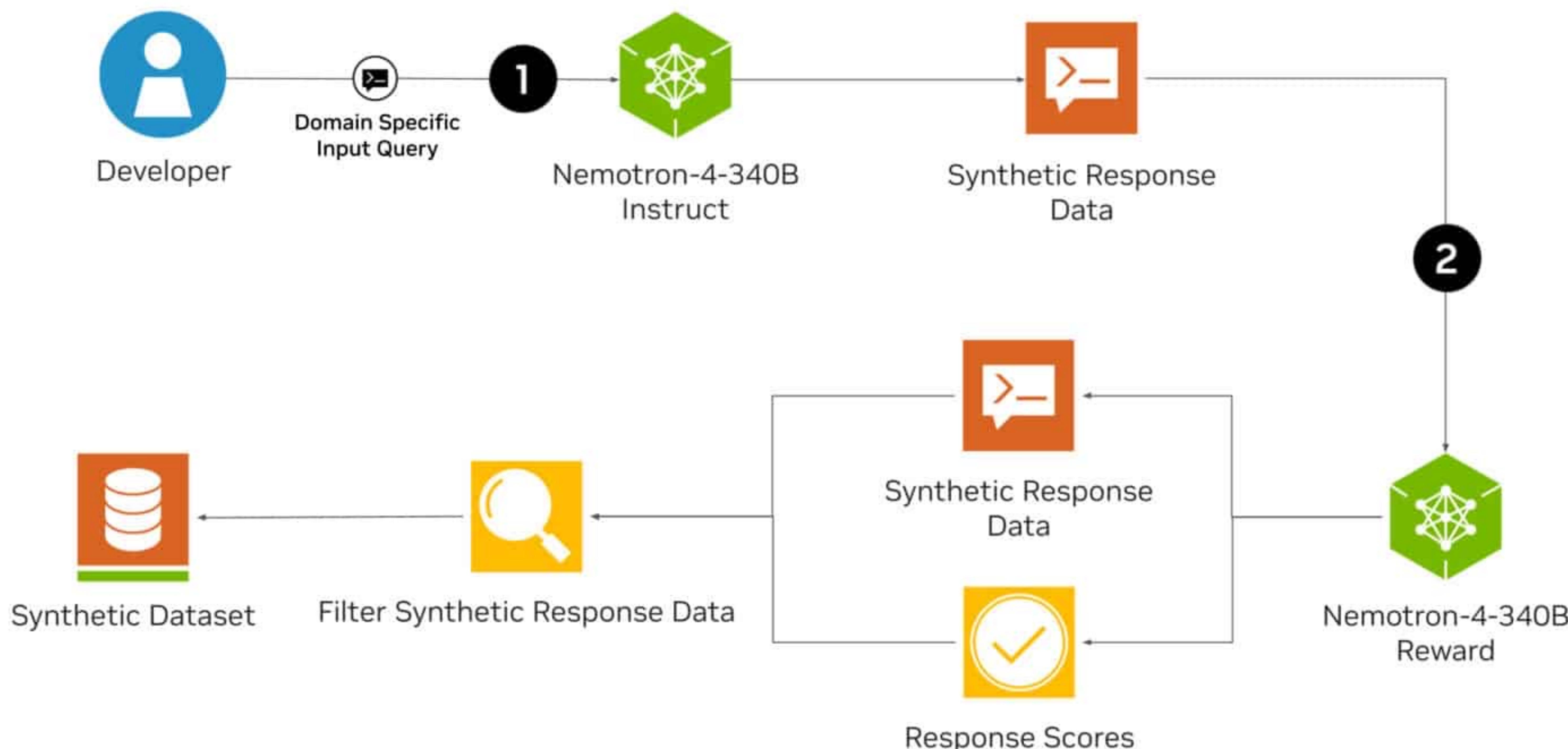
Coherence (連貫性): 2.5625

Complexity (複雜性): 0.88671875

Verbosity (冗長性): 1.671875

Data Generation Pipeline

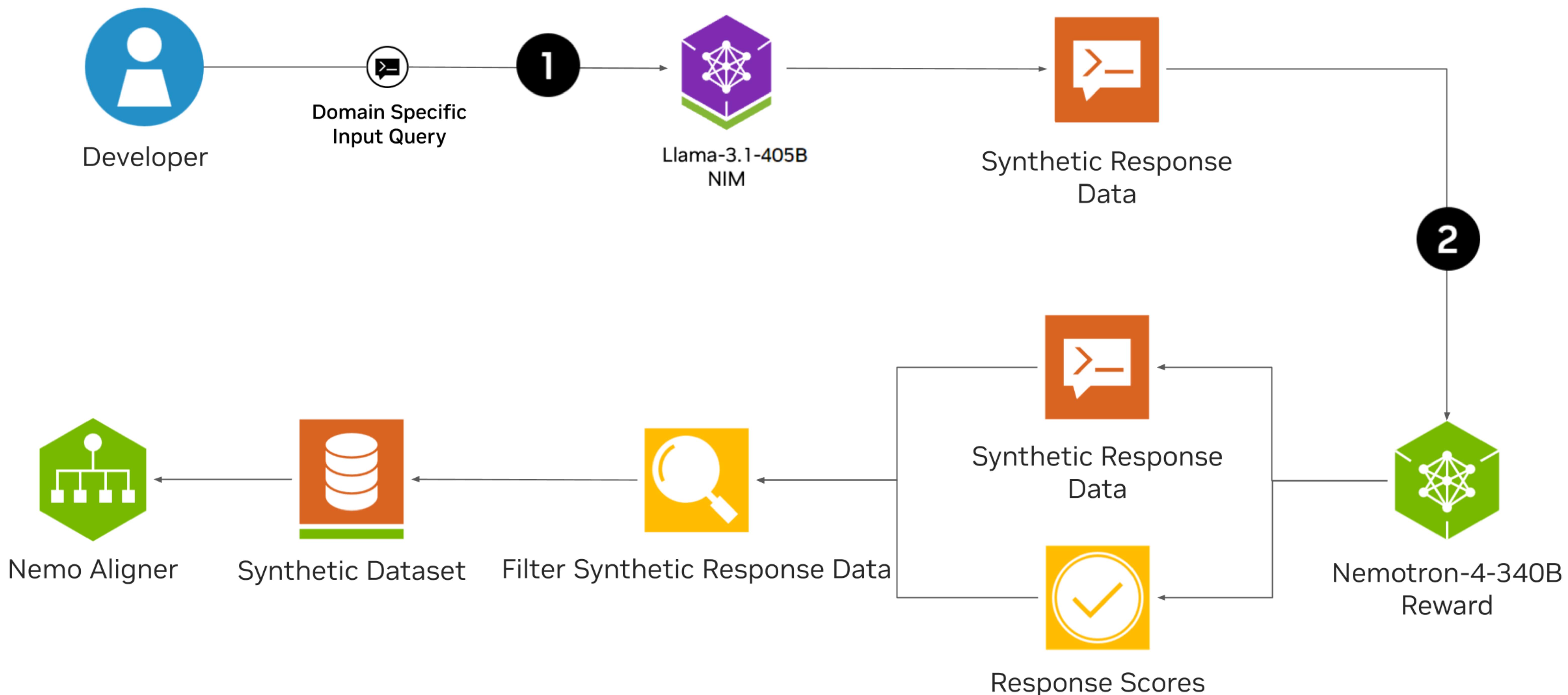
Nemotron-4-340B



- 1. Synthetic Response Generation:** A domain specific input query will be provided by the developer
- 2. Reward Model as a Judge:** **Nemotron-4 340B Reward** will be used to score the responses per question to be used for further alignment training

Data Generation Pipeline

Llama-3.1-405B

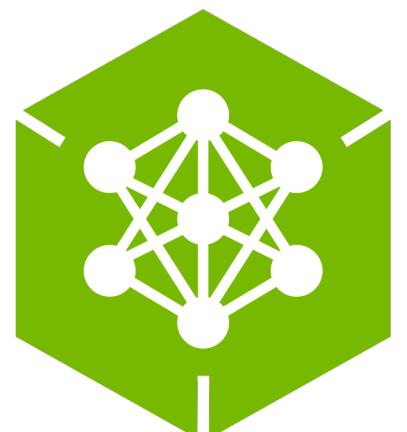


- 1. Synthetic Response Generation:** A domain specific input query will be provided by the developer
- 2. Reward Model as a Judge:** **Nemotron-4 340B Reward** will be used to score the responses per question to be used for further alignment training

How to Generate Synthetic data?

Topic, Question, Response Generation

Step 1. Subtopic Generation

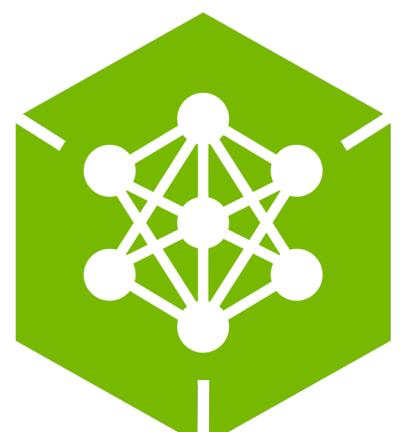


Given a topic, generate a list of {n_subtopics} subtopics that are related to the topic.

The topic is: {topic}

The list must be without numbers, and without any description of the subtopics. The subtopics should be separated by a comma. There must be no other text than the list.

Step 2. Question Generation

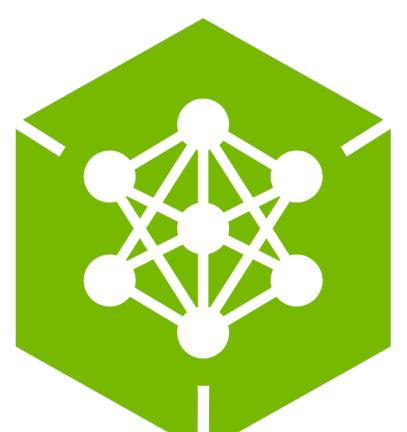


Given a topic, generate {n_questions} questions that could be asked about that topic. Your response should be in a list format.

The topic is: {sub_topic}

The list must be without numbers. The questions should be separated by a newline character. There must be no other text than the list.

Step 3. Response Generation



Given a question, generate 2 responses that could be given to that question. Your response should be in a list format.

The question is: {question}

The list must be in the format:

RESPONSE A: Response A text here
RESPONSE B: Response B text here

How to Generate Synthetic data?

Step 1. Subtopic Generation



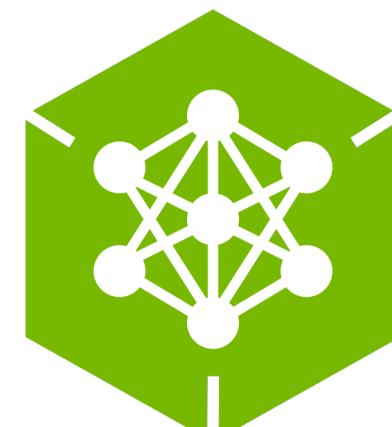
User

Given a topic, generate a list of {n_subtopics} subtopics that are related to the topic.

子領域數量

The topic is: {topic} 主題

The list must be without numbers, and without any description of the subtopics. The subtopics should be separated by a comma. There must be no other text than the list.



Model

Supervised Learning, Unsupervised Learning, Reinforcement Learning, Deep Learning, Natural Language Processing, Computer Vision, Predictive Modeling, Clustering Algorithms, Neural Networks, Regression Analysis

How to Generate Synthetic data?

Step 2. Question Generation

欲產生的問題數量

Given a topic, generate {n_questions} questions that could be asked about that topic. Your response should be in a list format.



User

The topic is: {sub_topic} 子領域

The list must be without numbers. The questions should be separated by a newline character. There must be no other text than the list.



Model

1. What is supervised learning and how does it differ from unsupervised learning?
2. How does supervised learning work in the context of machine learning algorithms?
3.

How to Generate Synthetic data?

Step 3. Response Generation



User

Given a question, generate 2 responses that could be given to that question. Your response should be in a list format.

The question is: {question} 問題

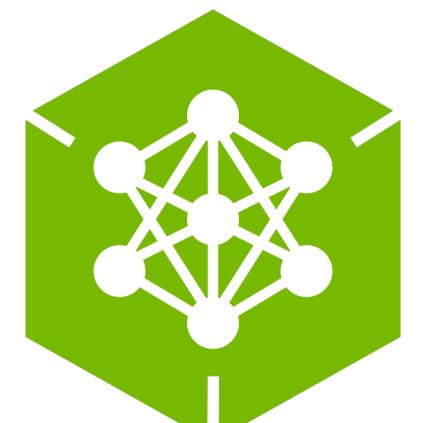
The list must be in the format:

RESPONSE A: Response A text here

RESPONSE B: Response B text here

For example:

question=What is supervised learning and how does it differ from unsupervised learning?



Model

RESPONSE A: Supervised learning is a type of machine learning where the algorithm is trained on labeled data, meaning the data is already tagged with the correct output. ...

RESPONSE B: Supervised learning is a machine learning approach where the model is trained on a dataset that includes both input data and corresponding output labels. ...

How to Generate Synthetic data?

Reward Model as a Judge

What is supervised learning and how does it differ from unsupervised learning?



User

Supervised learning is a type of machine learning where the algorithm is trained on labeled data, meaning the data is already tagged with the correct output. The goal of supervised learning is to learn a mapping between input data and the corresponding output labels, so the algorithm can make predictions on new, unseen data. In contrast, unsupervised learning involves training an algorithm on unlabeled data, and the goal is to identify patterns or structure in the data without any prior knowledge of the output labels

Model



Output reward scores

Helpfulness (幫助性): 4.09375

Correctness (正確性): 4.03125

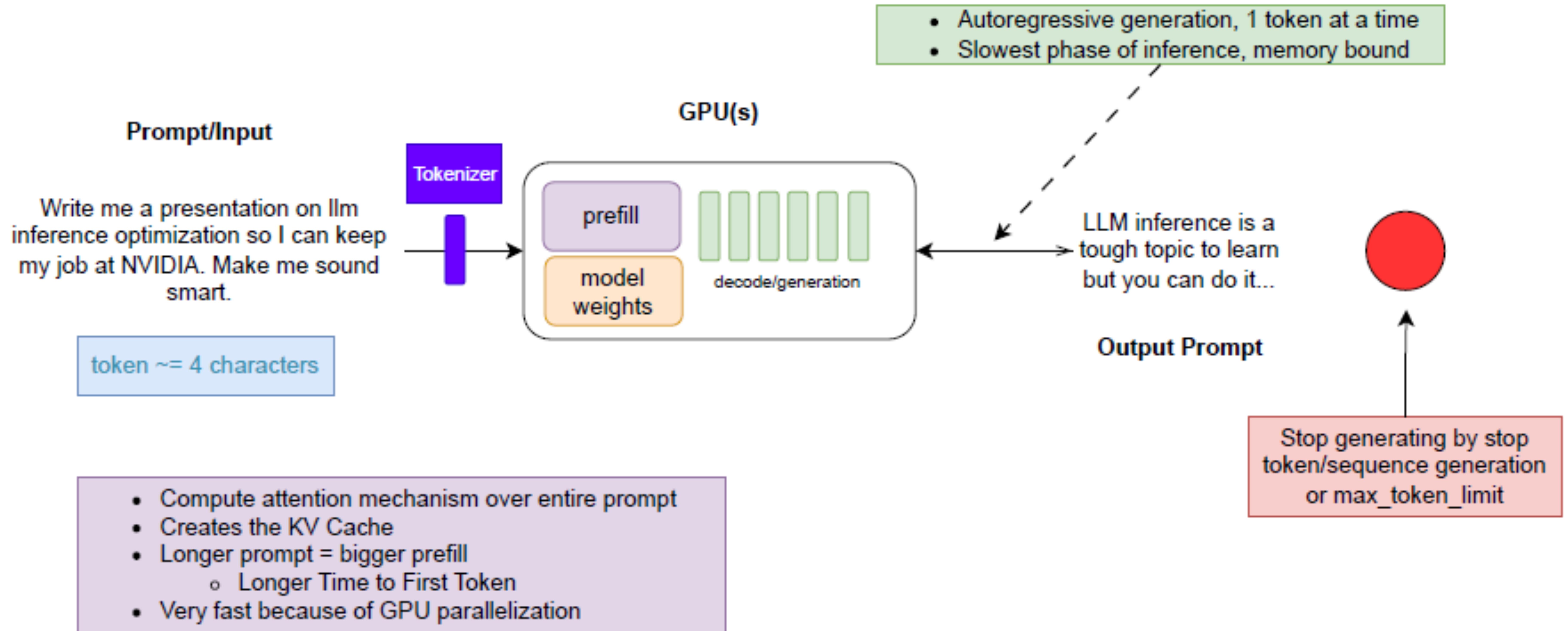
Coherence (連貫性): 4.25

Complexity (複雜性): 0.5703125

Verbosity (冗長性): 1.109375

LLM Inference Theory

Understanding LLM Inference

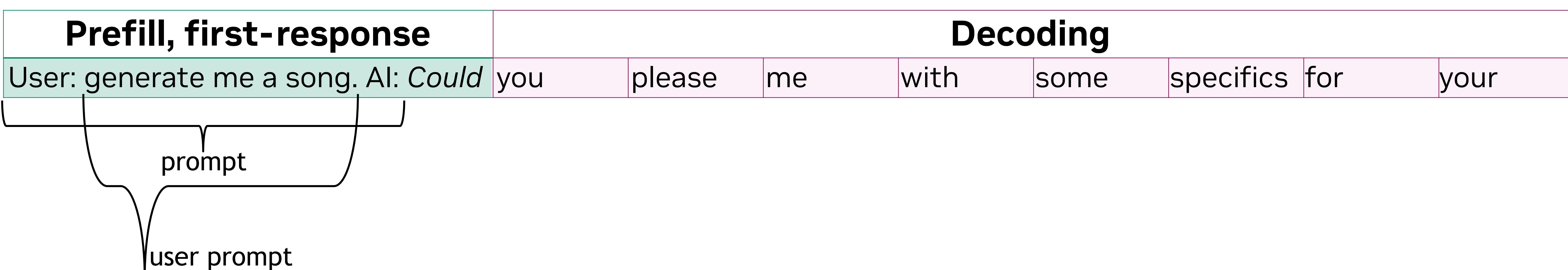
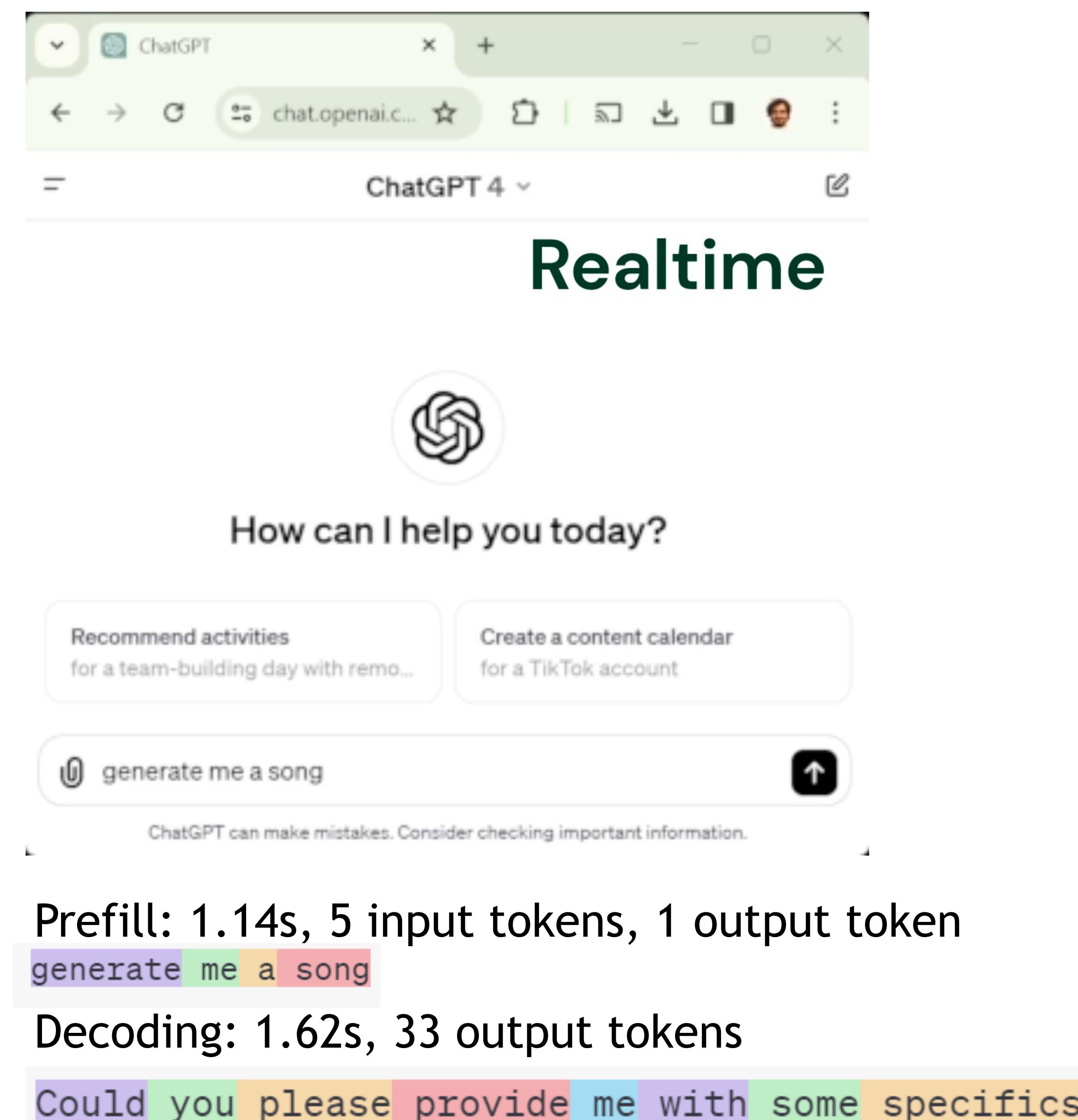


Source: [Mastering LLM Techniques: Inference Optimization](#)

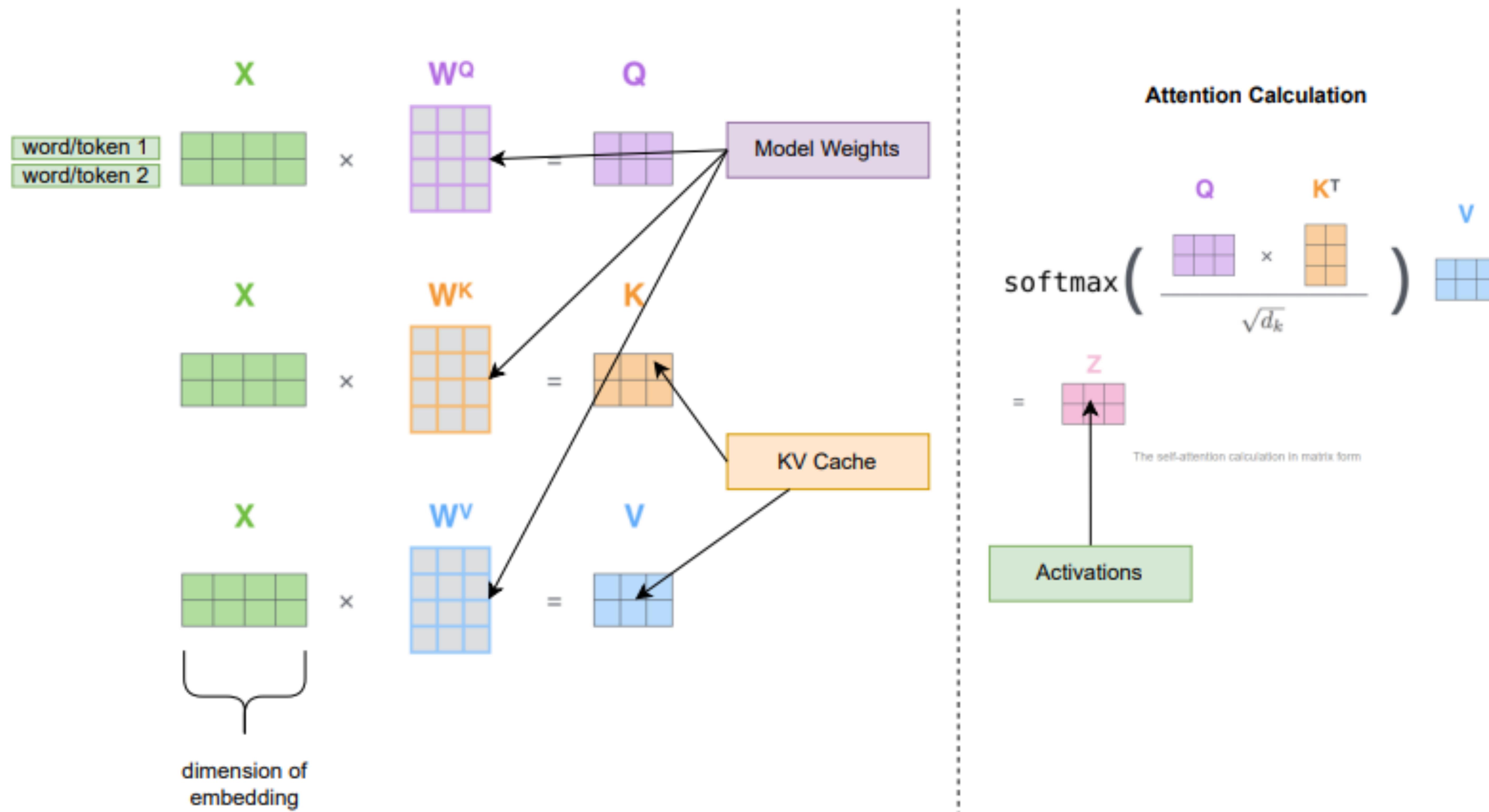
Two Stages of LLM Execution

Prefill vs Decoding

- **Prefill** = time to first token (TTFT)
 - Loading the user prompt into the system
 - From the request reception to the first token
 - Depends only on the number of input tokens
 - Populate KV-cache for all the tokens from the prompt.
 - Compute-bound for most of the reasonable prompt lengths
- **Decoding** = inter-token latency
 - Generating the response token by token, word by word
 - Inter-token latency depends on the total token number, both input and output tokens.
 - Usually memory-bound

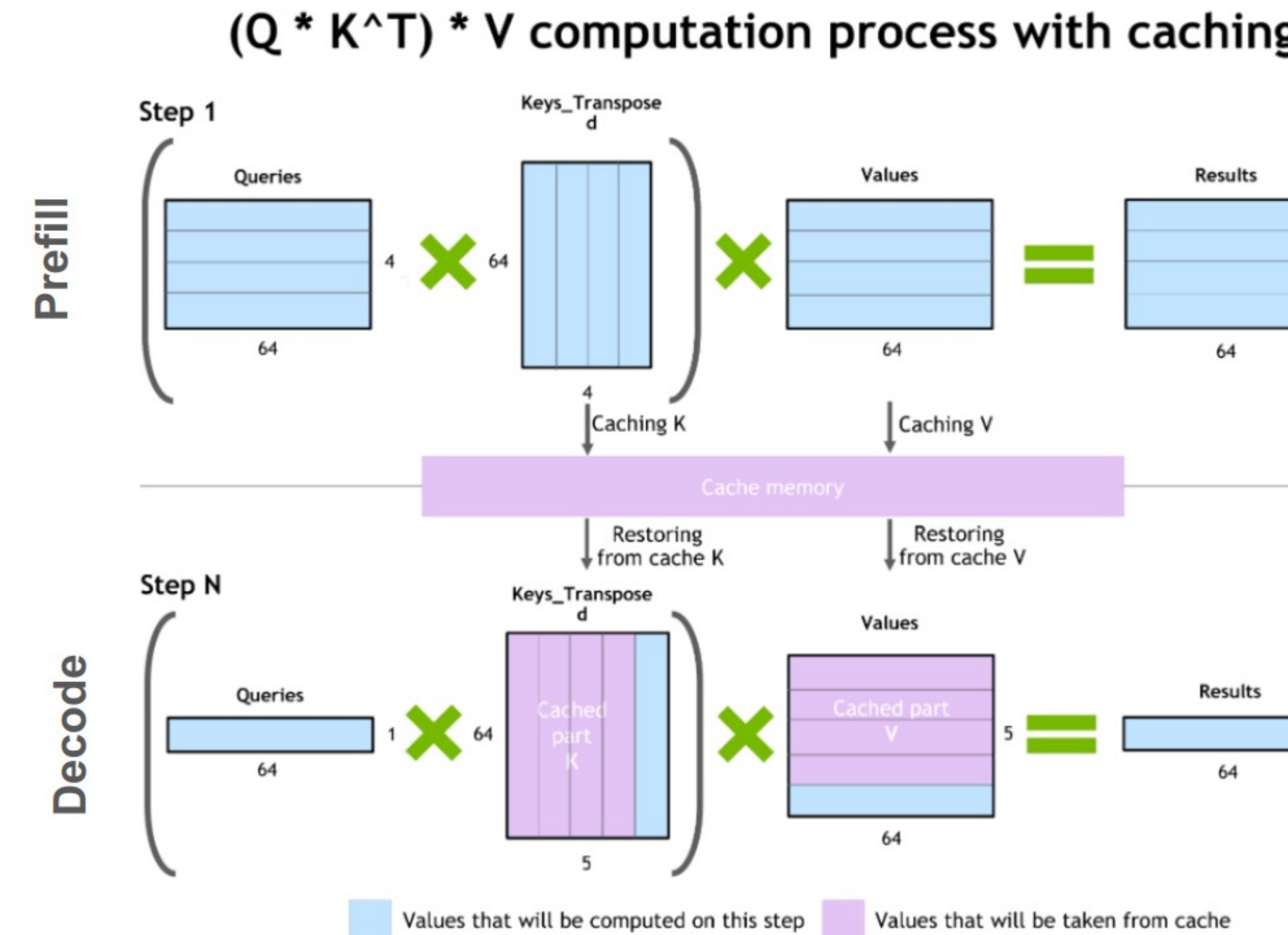


Model Weights, KV Cache and Attention



Source: [Jay Alammar: Illustrated Transformer](#)

Prefill and Importance of KV Cache



Source: [Mastering LLM Techniques: Inference Optimization](#)

LLM Memory Requirement

Model weights and KV cache

In effect, the two main contributors to the GPU LLM memory requirement are **model weights** and the **KV cache**.

- **Model weights:** Memory is occupied by the model parameters..

*Size of model weights in bytes: (**number of parameters**) × (**size of precision**)*

As an example, a model with 7 billion parameters (such as Llama-2-7B), loaded in 16-bit precision (FP16 or BF16) would take roughly $7\text{B} * \text{size of (FP16)} \approx \mathbf{14\text{ GB}}$ in memory

- **KV cache:** Memory is occupied by the caching of self-attention tensors to avoid redundant computation.

*Size of KV cache in bytes: (**batch size**) × (**sequence length**) × (**number of layers**) × (**hidden size**) ×
2 × (**size of precision**)*

For example, with a Llama-2-7B model in 16-bit precision and a batch size of 1, the size of the KV cache will be $1 * 4096 * 2 * 32 * 4096 * 2$ bytes, which is **~2 GB**.

Inflight Batching

Maximizing GPU Utilization during LLM Serving

Inflight Batching to optimize GPU utilization during LLM Serving

- Replaces completed requests in the batch
 - Evicts requests after EoS & inserts a new request
- Improves throughput, time to first token, & GPU utilization

Batch Elements	Iteration									...
	1	2	3	4	5	6	7	8	9	
R_1						END				R_5
R_2			END							R_6
R_3				END						R_7
R_4								END	R_8	...

Static Batching

Batch Elements	Iteration									...
	1	2	3	4	5	6	7	8	9	
R_1						END	R_7			...
R_2			END	R_5						...
R_3				END	R_6			END	R_8	...
R_4								END	R_9	...

Inflight Batching

Context | Gen | EoS | NoOp

KV Cache Optimizations

Paged & Quantized KV Cache

Paged KV Cache improves memory consumption & utilization

- Stores keys & values in non-contiguous memory space
- Allows for reduced memory consumption of KV cache
- Allocates memory on demand

Quantized KV Cache improves memory consumption & perf

- Reduces KV Cache elements from 16b to 8b (or less!)
- Reduces memory transfer improving performance
- Supports INT8 / FP8 KV Caches

Both allow for increased peak performance

KV Cache Contents:

TensorRT-LLM optimizes inference on
NVIDIA GPUs ...

Block 0	TensorRT	LLM	is	...
Block 1				
Block 2	Hello	World		
Block 3				

Traditional KV Caching

B ₀	TensorRT	LLM	is	...
B ₁				
B ₂	Hello	World		
B ₃				

Paged KV Cache

B ₀	TRT	LLM	is	...			
B ₁							
B ₂	Hello	World					
B ₃							

Quantized Paged KV Cache

Request 1 | Request 2 | Wasted | Free

