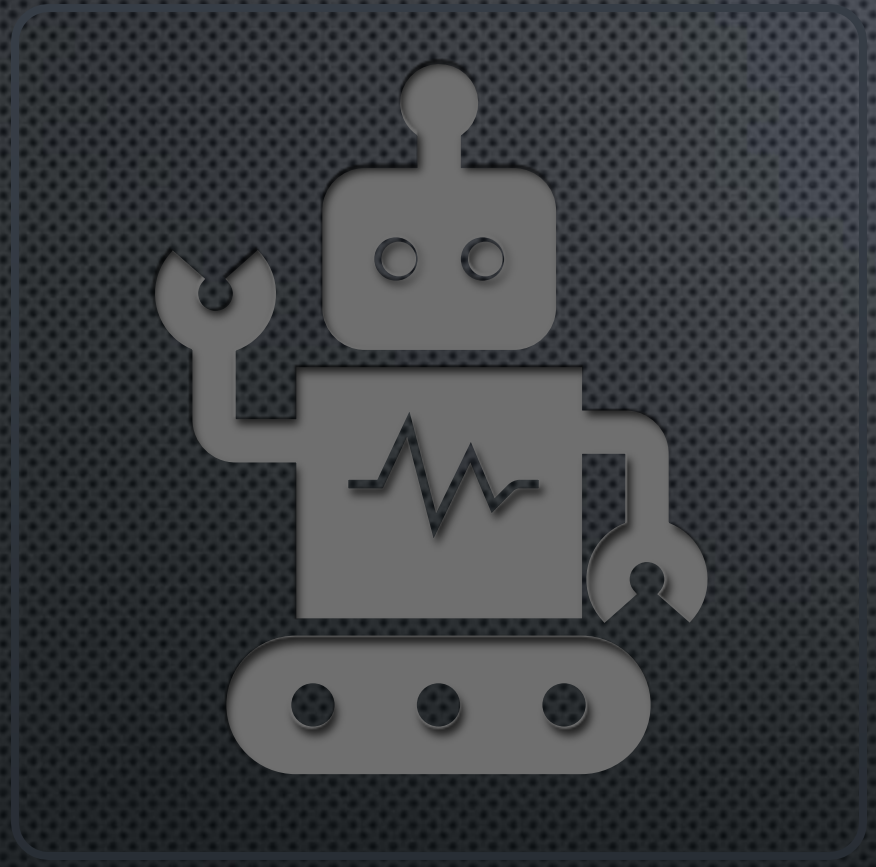


INTRODUCTION TO ADVANCED LARGE LANGUAGE MODELS

A COMPREHENSIVE TUTORIAL ON TECHNIQUES, ARCHITECTURES, AND PRACTICAL APPLICATIONS

PART III

Giorgio Roffo, PhD

Explore and Connect

LinkedIn: Giorgio Roffo - Linkedin
ResearchGate: Work Done
Google Scholar: My Publications
GitHub: giorgioroffo

Special thanks to Shelbee Eigenbrode, Antje Barth, and Mike Chambers for their work on "Generative AI with Large Language Models" (Coursera, Amazon AWS, 2023), which significantly informed and inspired the material used in these slides.

WE ARE GOING TO TALK ABOUT…

1. **Beyond Basic LLMs**: Advances Toward Sophisticated Applications
   - Retrieval-Augmented Generation (RAG)
   - Program-Aided Language Models (PAL)
   - Integration of Reasoning and Action (ReAct)
   - Architectural Designs for LLM Applications (ChatGPT, Gemini, etc..)

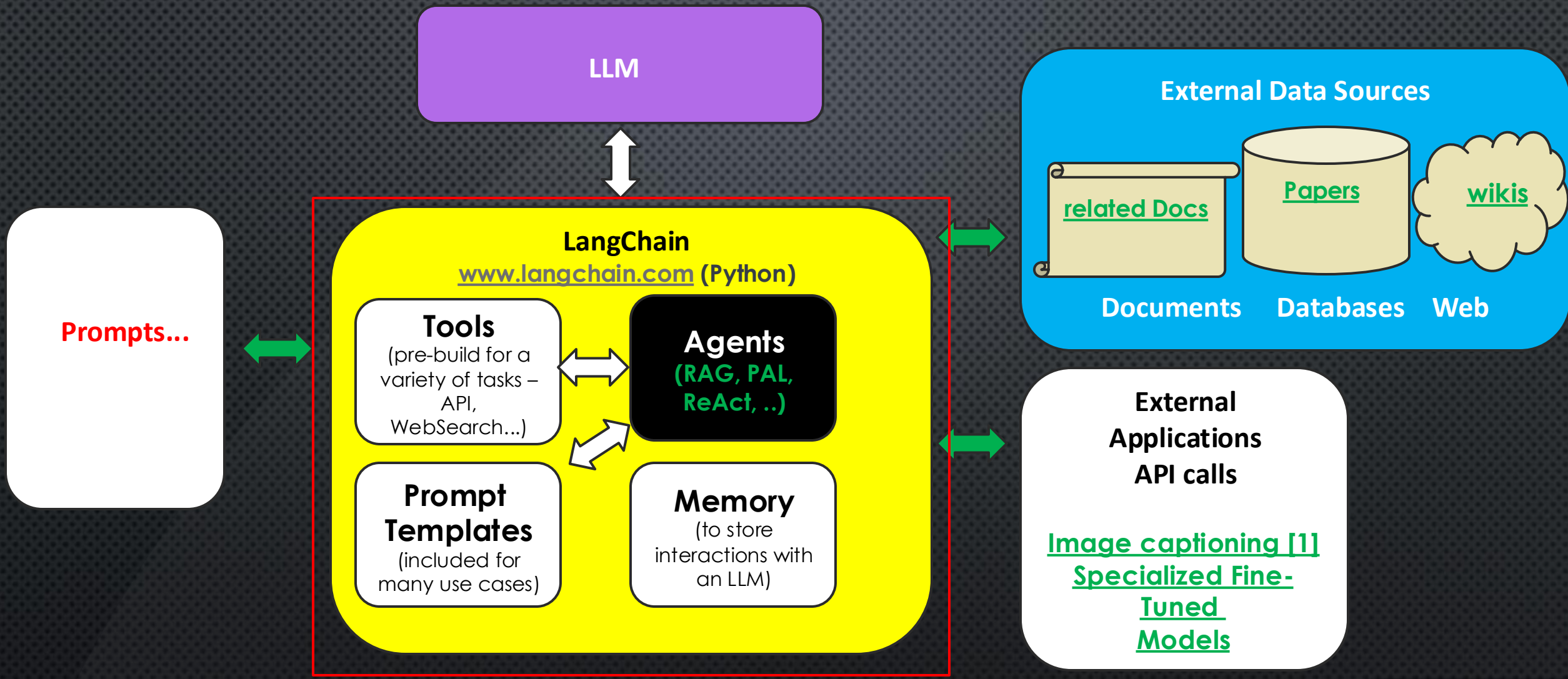
Part I

2. **Introduction: Transformers**
3. **Selecting LLM Architectures**
4. **LLM Training Resources**: GPU Memory Requirements
   - Models: BERT-L (340M), GPT-2 (1.5B), LLaMA-2 (7-13-70B), GPT-3 (175B), PaLM (540B)
   - GPU RAM for Models:
     - 1B parameters: 24GB (32-bit precision)
     - 175B parameters: 4200GB (32-bit precision)
     - 500B parameters: 12000GB (32-bit precision)

Part II

5. **Datasets & Benchmarks**: Criteria for Selecting Evaluation Metrics and Datasets
6. **Fine-Tuning Strategies**: Addressing Specific Tasks and Preventing Overfitting
7. **RLHF**: Enhancing LLMs Through Human Interaction
8. **ReST (Google)**: Reinforced Self-Training for Language Modeling
9. MED-Gemini

Part III



[1] Scaling Up Vision-Language Pre-Training for Image Captioning *Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, Lijuan Wang*; Proceedings of the **IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022**, pp. 17980-17989

WE ARE GOING TO TALK ABOUT…

1. **Beyond Basic LLMs**: Advances Toward Sophisticated Applications
   - Retrieval-Augmented Generation (RAG)
   - Program-Aided Language Models (PAL)
   - Integration of Reasoning and Action (ReAct)
   - Architectural Designs for LLM Applications (ChatGPT, Gemini, etc..)

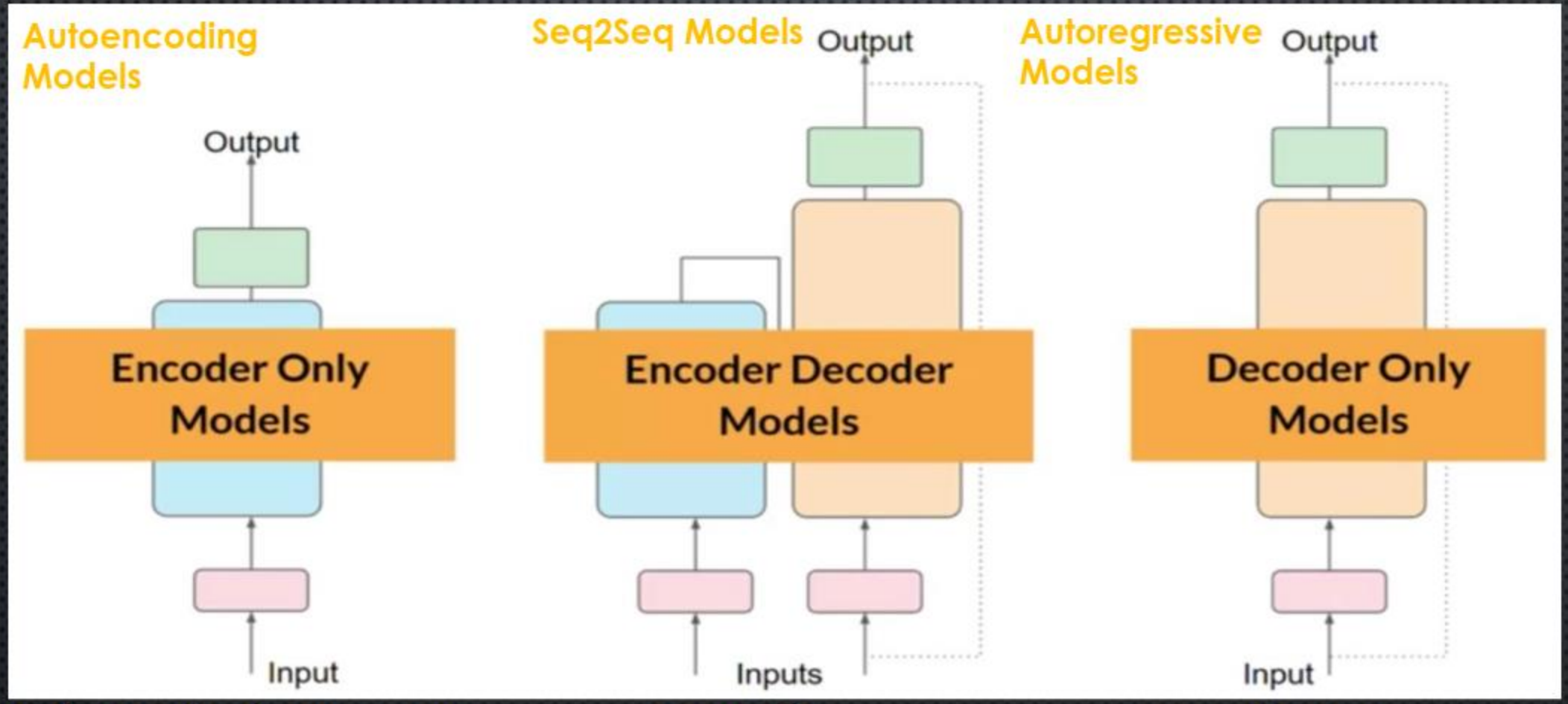
Part I

2. **Introduction: Transformers**
3. **Selecting LLM Architectures**
4. **LLM Training Resources**: GPU Memory Requirements
   - Models: BERT-L (340M), GPT-2 (1.5B), LLaMA-2 (7-13-70B), GPT-3 (175B), PaLM (540B)
   - GPU RAM for Models:
     - 1B parameters: 24GB (32-bit precision)
     - 175B parameters: 4200GB (32-bit precision)
     - 500B parameters: 12000GB (32-bit precision)

Part II

5. **Datasets & Benchmarks**: Criteria for Selecting Evaluation Metrics and Datasets
6. **Fine-Tuning Strategies**: Addressing Specific Tasks and Preventing Overfitting
7. **RLHF**: Enhancing LLMs Through Human Interaction
8. **ReST (Google)**: Reinforced Self-Training for Language Modeling
9. MED-Gemini

Part III



understanding input context

Main Task: Sentiment Analysis, word class.
Models: BERT, ROBERTA

Conversion

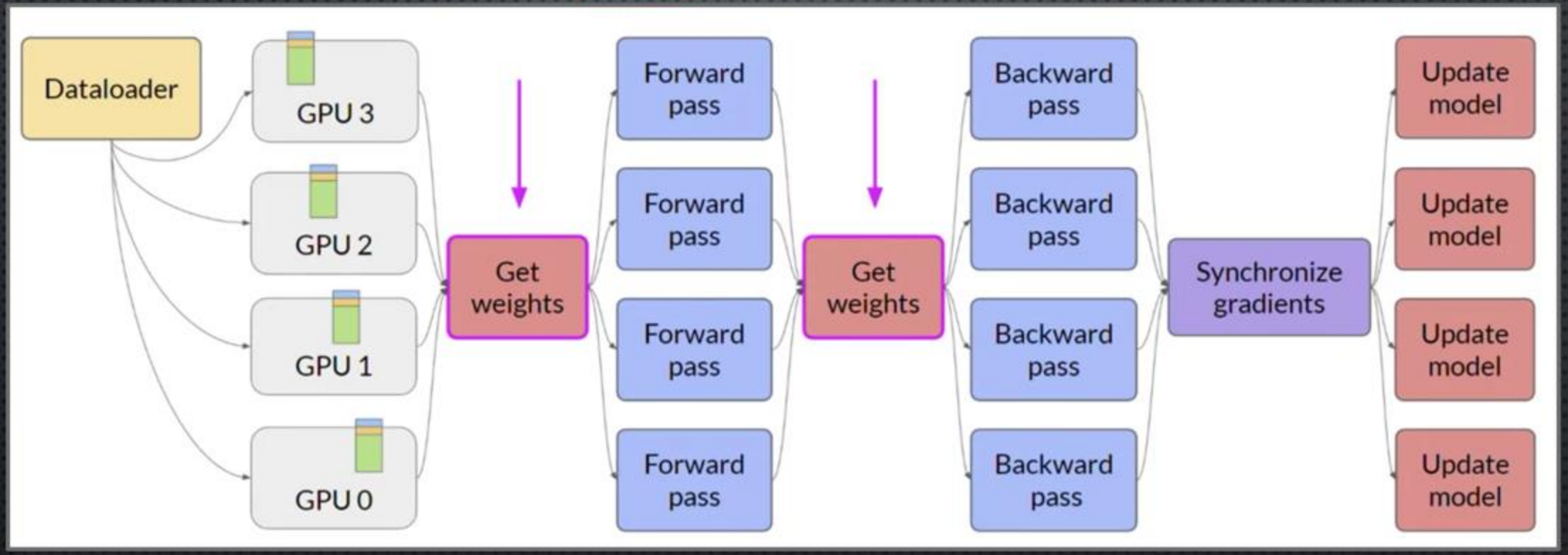
Main Task: Translation, Summarization, Q/A
Models: BART, T5

Generating text

Main Task: Generalize all tasks.
Models: GPT, BLOOM, Jurassic, LLaMA

Efficient multi-GPU compute strategies

1. Quantization. Reduces memory to store and train models
2. Distributed Data Parallel (DDP)
3. Fully Sharded Data Parallel (FSDP) -> PyTorch 2023



[9] Rajbhandari S, Rasley J, Ruwase O, He Y. **Zero: Memory optimizations toward training trillion parameter models.** In SC20: International Conference for High Performance Computing, Networking, Storage and Analysis 2020 Nov 9 (pp. 1-16). IEEE.

[10] Zhao Y, Gu A, Varma R, Luo L, Huang CC, Xu M, Wright L, Shojanazeri H, Ott M, Shleifer S, Desmaison A. **Pytorch FSDP: experiences on scaling fully sharded data parallel.** arXiv preprint arXiv:2304.11277. **2023** Apr 21.

MedQA Original: https://paperswithcode.com/dataset/medqa-usmle

Question	A 27-year-old male presents to urgent care complaining of pain with urination. He reports that the pain started 3 days ago. He has never experienced these symptoms before. He *denies gross hematuria or pelvic pain*. He is sexually active with his girlfriend, and they consistently use condoms. When asked about recent travel, he admits to recently returning from a boys' trip" in Cancun where he had *unprotected sex* 1 night with a girl he met at a bar. The patients medical history includes type I diabetes that is controlled with an insulin pump. His mother has rheumatoid arthritis. The patients temperature is 99 F (37.2 C), blood pressure is 112/74 mmHg, and pulse is 81/min. On physical examination, there are no lesions of the penis or other body rashes. No costovertebral tenderness is appreciated. A urinalysis reveals no blood, glucose, ketones, or proteins but is *positive for leukocyte esterase*. A urine microscopic evaluation shows a *moderate number of white blood cells* but no casts or crystals. A urine culture is negative. Which of the following is the most likely cause for the patient's symptoms?
Options	**A: Chlamydia trachomatis**, B: Systemic lupus erythematosus, C: Mycobacterium tuberculosis, D: Treponema pallidum
Evidence	At least one-third of male patients with *C. trachomatis* urethral infection have *no evident signs or symptoms of urethritis*. ... Such patients generally have *pyuria* ..., a *positive leukocyte esterase test*, ...
Question	A 57-year-old man presents to his primary care physician with a 2-month history of *right upper and lower extremity weakness*. He noticed the weakness when he started falling far more frequently while running errands. Since then, he has had *increasing difficulty* with walking and lifting objects. His past medical history is significant only for well-controlled hypertension, but he says that some members of his *family have had musculoskeletal problems*. His right upper extremity shows *forearm atrophy* and *depressed reflexes* while his right lower extremity is *hypertonic with a positive Babinski sign*. Which of the following is most likely associated with the cause of this patients symptoms?
Options	A: HLA-B8 haplotype, B: HLA-DR2 haplotype, **C: Mutation in SOD1**, D: Mutation in SMN1, E: Viral infection
Evidence	1. The manifestations of ALS ... *insidiously* developing *asymmetric weakness*, usually first evident distally in one of the limbs. 2. ... *hyperactivity of the muscle-stretch reflexes (tendon jerks)* and, often, *spastic resistance to passive movements* ... 3. *Familial ALS (FALS)*... clinically indistinguishable from sporadic ALS... Genetic studies have identified mutations in multiple genes, including cytosolic enzyme *SOD1*...

Table 1: Two examples of MEDQA. The correct answer among options is marked in bold font. Key words in the question and evidence text to help answer the questions are highlighted in italic font. Evidence for both examples are from the textbook "Harrison's Principles of Internal Medicine".

Capabilities of Gemini Models in Medicine https://arxiv.org/html/2404.18416v1

- **MedQA-RS** (Reasoning and Search)
- **MedQA-R (**Reasoning)

MedQA-R (Reasoning), which extends MedQA with synthetically generated reasoning explanations, or "Chain-of-Thoughts" (CoTs), and MedQA-RS (Reasoning and Search), which extends MedQA-R with instructions to use web search results as additional context to improve answer accuracy.

*[25] Saab K, Tu T, Weng WH, et Al.. **Capabilities of gemini models in medicine. Google, DeepMind** et al. arXiv preprint arXiv:2404.18416. 2024*

ROUGE (Recall-Oriented Understudy for Gisting Evaluation):

$$\text{ROUGE-L Recall:} = \frac{\text{LCS(Gen, Ref)}}{\text{unigrams in reference}}$$

$$\text{ROUGE-L Precision:} = \frac{\text{LCS(Gen, Ref)}}{\text{unigrams in output}}$$

$$\text{ROUGE-L F1:} = 2\,\frac{\text{precision x recall}}{\text{precision + recall}}$$

Example ROUGE-L

$$\text{ROUGE-2 Recall:} = \frac{\text{bigram matches}}{\text{bigrams in reference}}$$

$$\text{ROUGE-2 Precision:} = \frac{\text{bigram matches}}{\text{bigrams in output}}$$

$$\text{ROUGE-2 F1:} = 2\,\frac{\text{precision x recall}}{\text{precision + recall}}$$

Example ROUGE-2

BLEU Score (Bilingual Evaluation Understudy):

BLEU is a metric for evaluating a machine-translated text by comparing it to one or more reference translations.
The core idea is to calculate the precision of n-grams (contiguous sequence of n items from a given sample of text or speech) in the candidate translation that also appear in the reference translation.

WE ARE GOING TO TALK ABOUT…

1. **Beyond Basic LLMs**: Advances Toward Sophisticated Applications
    - Retrieval-Augmented Generation (RAG)
    - Program-Aided Language Models (PAL)
    - Integration of Reasoning and Action (ReAct)
    - Architectural Designs for LLM Applications (ChatGPT, Gemini, etc..)
2. **Introduction: Transformers**
3. **Selecting LLM Architectures**
4. **LLM Training Resources**: GPU Memory Requirements
    - Models: BERT-L (340M), GPT-2 (1.5B), LLaMA-2 (7-13-70B), GPT-3 (175B), PaLM (540B)
    - GPU RAM for Models:
        - 1B parameters: 24GB (32-bit precision)
        - 175B parameters: 4200GB (32-bit precision)
        - 500B parameters: 12000GB (32-bit precision)
5. **Datasets & Benchmarks**: Criteria for Selecting Evaluation Metrics and Datasets
6. **Fine-Tuning Strategies**: Addressing Specific Tasks and Preventing Overfitting
7. **RLHF**: Enhancing LLMs Through Human Interaction
8. **ReST (Google)**: Reinforced Self-Training for Language Modeling
9. MED-Gemini

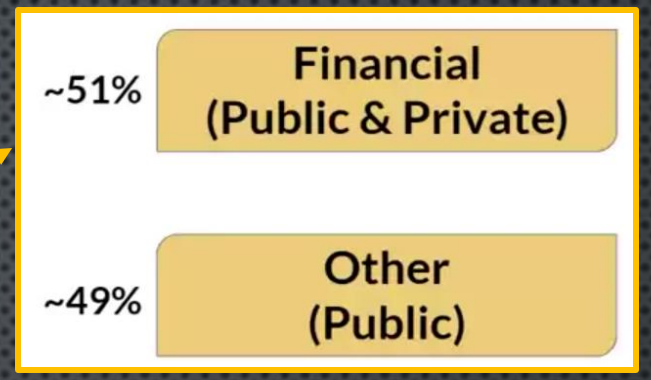
Domain adaptation

BloombergGPT: A Large Language Model for Finance

Shijie Wu[1,*], Ozan İrsoy[1,*], Steven Lu[1,*], Vadim Dabravolski[1], Mark Dredze[1,2], Sebastian Gehrmann[1], Prabhanjan Kambadur[1], David Rosenberg[1], Gideon Mann[1]
[1] Bloomberg, New York, NY USA
[2] Computer Science, Johns Hopkins University, Baltimore, MD USA
gmann16@bloomberg.net

Abstract

The use of NLP in the realm of financial technology is broad and complex, with applications ranging from sentiment analysis and named entity recognition to question answering. Large Language Models (LLMs) have been shown to be effective on a variety of tasks; however, no LLM specialized for the financial domain has been reported in literature. In this work, we present BLOOMBERGGPT, a 50 billion parameter language model that is trained on a wide range of financial data. We construct a 363 billion token dataset based on Bloomberg's extensive data sources, perhaps the largest domain-specific dataset yet, augmented with 345 billion tokens from general purpose datasets. We validate BLOOMBERGGPT on standard LLM benchmarks, open financial benchmarks, and a suite of internal benchmarks that most accurately reflect our intended usage. Our mixed dataset training leads to a model that outperforms existing models on financial tasks by significant margins without sacrificing performance on general LLM benchmarks. Additionally, we explain our modeling choices, training process, and evaluation methodology. As a next step, we plan to release training logs (Chronicles) detailing our experience in training BLOOMBERGGPT.



OR:

Medical language

After a strenuous workout, the patient experienced severe myalgia that lasted for several days.

After the biopsy, the doctor confirmed that the tumor was malignant and recommended immediate treatment.

Sig: 1 tab po qid pc & hs

- *Uncommon words to describe medical conditions and procedures.*
- *Infrequent terms in common training sets*

[9] Wu S, Irsoy O, Lu S, Dabravolski V, Dredze M, Gehrmann S, Kambadur P, Rosenberg D, Mann G. **Bloomberggpt: A large language model for finance**. arXiv preprint arXiv:2303.17564. 2023 Mar 30.

LLM fine-tuning Strategies:

- **Instruction Fine-Tuning (full fine-tuning updates model parameters)**
  - Each prompt/completion pair includes a specific "instruction" to the LLM.
    - *"Classify this review:" [TEXT] "Sentiment:"*
    - *"Summarize the following text:" [TEXT]*
    - *"Translate this sentence to: .." [TEXT]*
  - Public datasets are not formatted, there are many prompt-template libraries that can format the data in prompt instruction format for different tasks:
    - Classification / sentiment analysis
      - "Given the following review: [TEXT] predict the associated rating [ANSWER_CHOICES]"
    - Text Generation
      - "*Generate a [STAR_RATING]-star review (1 being lowest and 5 being highest) about this product [PRODUCT_TITLE]. || [BODY].*"
    - Text Summarization
      - "*Give a short sentence describing the following product review: [BODY]\ ||[REVIEW_HEADLINE].*"

```
"samsum": [
        ("{dialogue}\n\Briefly summarize that dialogue.", "{summary}"),
        ("Here is a dialogue:\n{dialogue}\n\nWrite a short summary!",
         "{summary}"),
        ("Dialogue:\n{dialogue}\n\nWhat is a summary of this dialogue?",
         "{summary}"),
        ("{dialogue}\n\nWhat was that dialogue about, in two sentences or less?",
         "{summary}"),
        ("Here is a dialogue:\n{dialogue}\n\nWhat were they talking about?",
         "{summary}"),
        ("Dialogue:\n{dialogue}\nWhat were the main points in that "
         "conversation?", "{summary}"),
        ("Dialogue:\n{dialogue}\nWhat was going on in that conversation?",
         "{summary}"),
]
```

LLM fine-tuning Strategies:

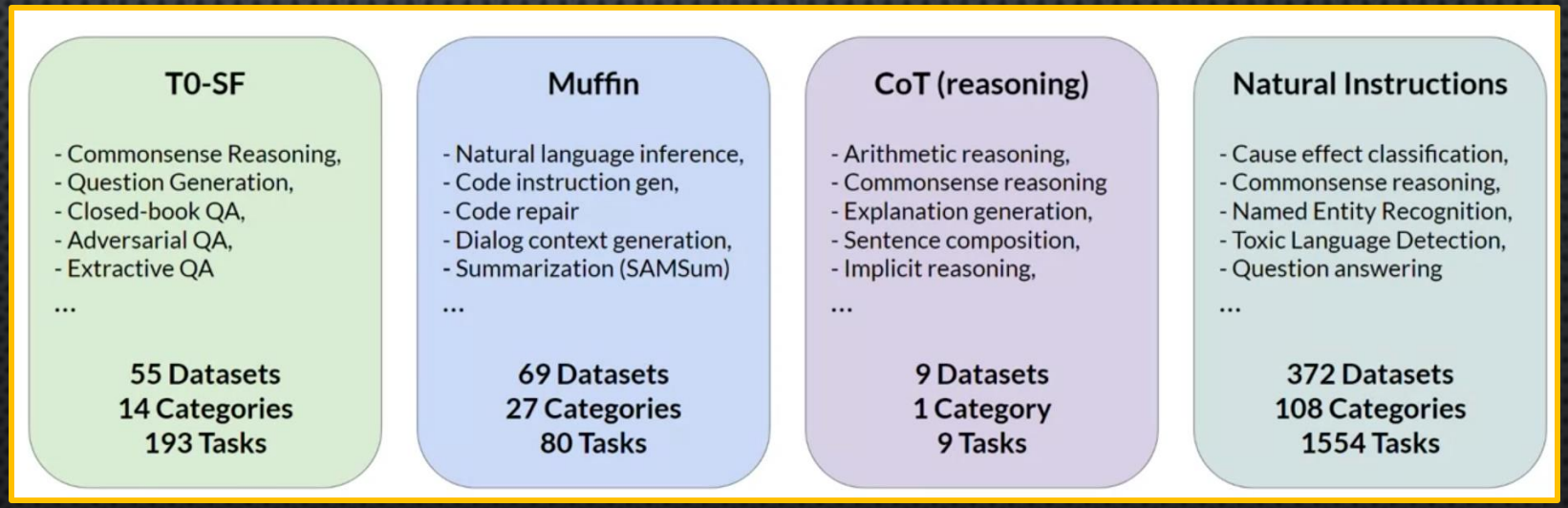
- Instruction Fine-Tuning (full fine-tuning updates model parameters)

- **Fine-tuning on a single task**
  - Improved performance on the new task.
  - Downside: **Catastrophic Forgetting.**
    - Catastrophic forgetting happens because the full fine-tuning process modifies the weights of the original LLM
    - The model forget in doing the other tasks even important and related to the new task.
      - *For example, improve the ability of a model to perform sentiment analysis on a review forget how to do named entity recognition.*

LLM fine-tuning Strategies:

- Instruction Fine-Tuning (full fine-tuning updates model parameters)
- Fine-tuning on a single task
- **Fine-Tuning on multiple tasks**
  - The dataset contains examples that instruct the model to carry out a variety of tasks.
  - Reduces the catastrophic forgetting
  - Drawback: it requires a lot of data (50K-100K examples)
  - Examples are **FLAN** models (**F**ine-tuned **LA**nguage **N**et)
    - FLAN-T5
    - FLAN-PaLM
    - …

FLAN-T5
it's been fine tuned on 473 datasets across 146 task categories

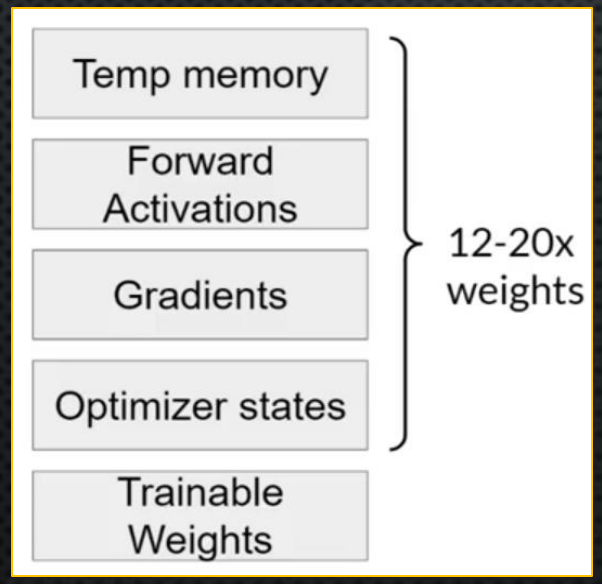


*[2] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. **2024**;25(70):1-53.*

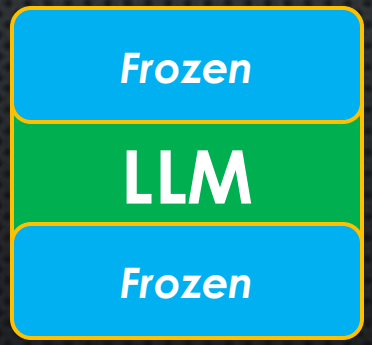
LLM fine-tuning Strategies:

- Instruction Fine-Tuning (full fine-tuning updates model parameters)
- Fine-tuning on a single task
- Fine-Tuning on multiple tasks
- **Parameter Efficient Fine Tuning (PEFT)**
  - **LoRA (Low Rank Adaption)**
  - **Adapters**
  - **Soft prompts**

Full fine-tuning



Parameter Efficient Fine Tuning (PEFT)



LLM with most layers frozen

Reduced memory requirements

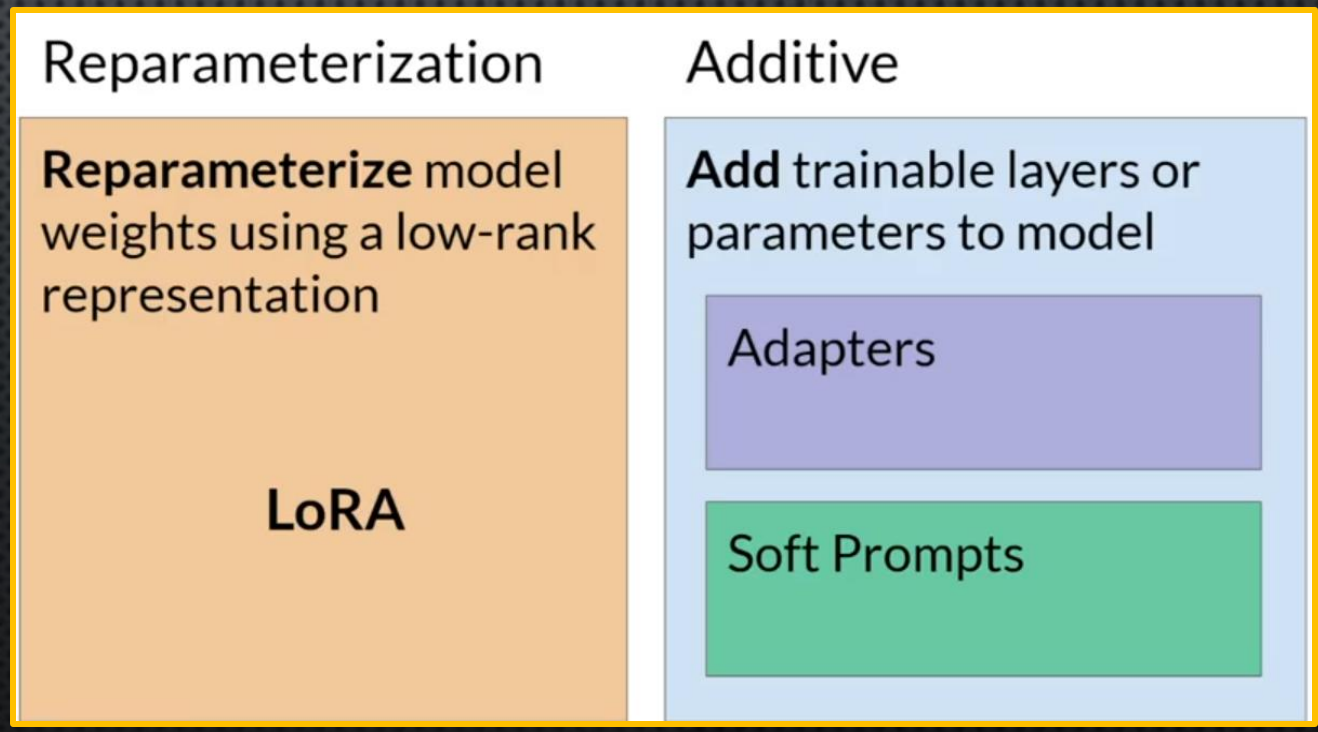
Small number of trainable layers

New trainable layers

LLM with additional layers for PEFT

LLM fine-tuning Strategies:

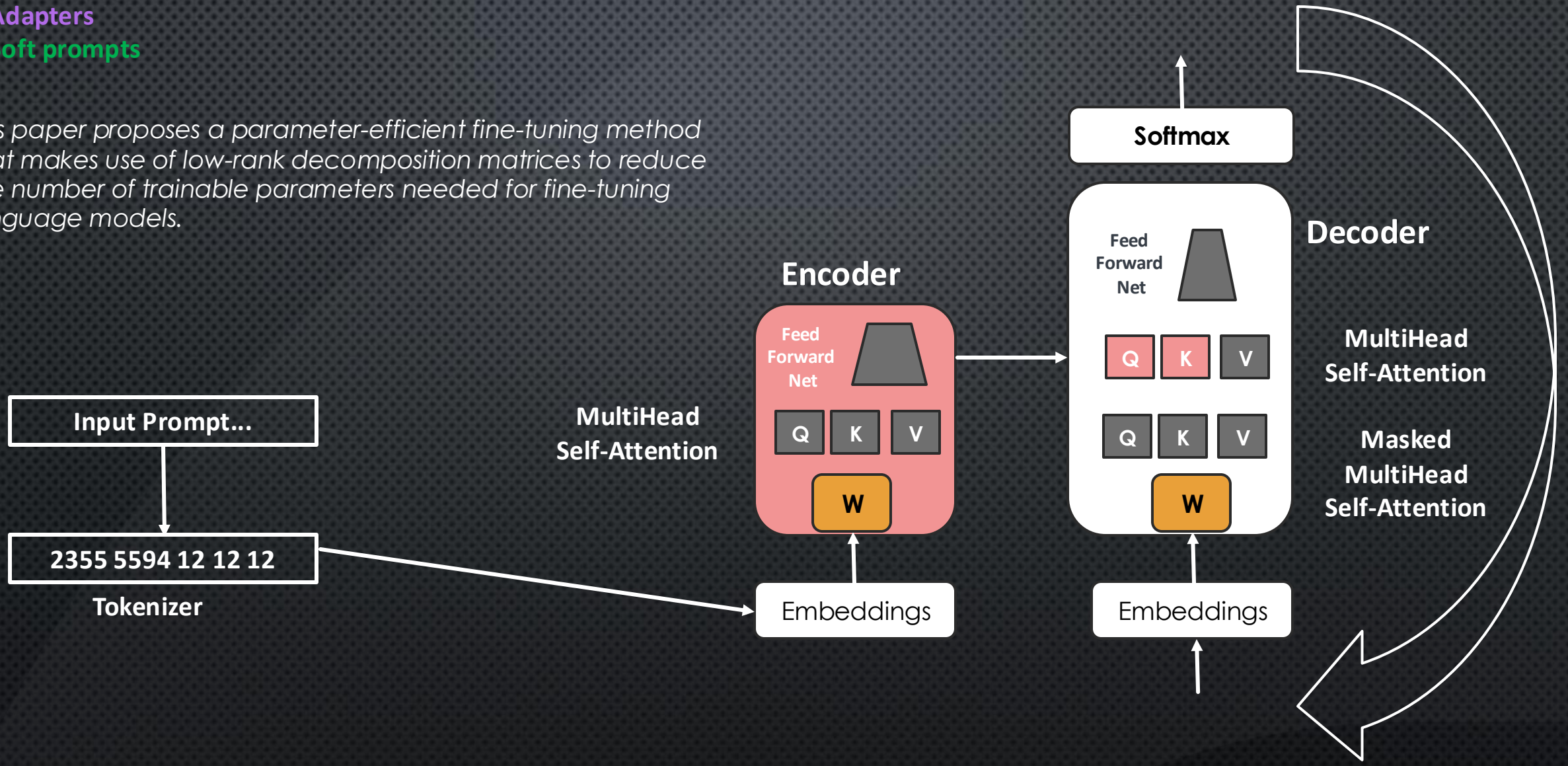
- Instruction Fine-Tuning (full fine-tuning updates model parameters)
- Fine-tuning on a single task
- Fine-Tuning on multiple tasks
- **Parameter Efficient Fine Tuning (PEFT)**
  - **LoRA (Low Rank Adaption) - Reparameterization Technique**
  - **Adapters**
  - **Soft prompts**



[11] Lialin V, Deshpande V, Rumshisky A. ***Scaling down to scale up: A guide to parameter-efficient fine-tuning****. arXiv preprint arXiv:2303.15647. 2023 Mar 28.*

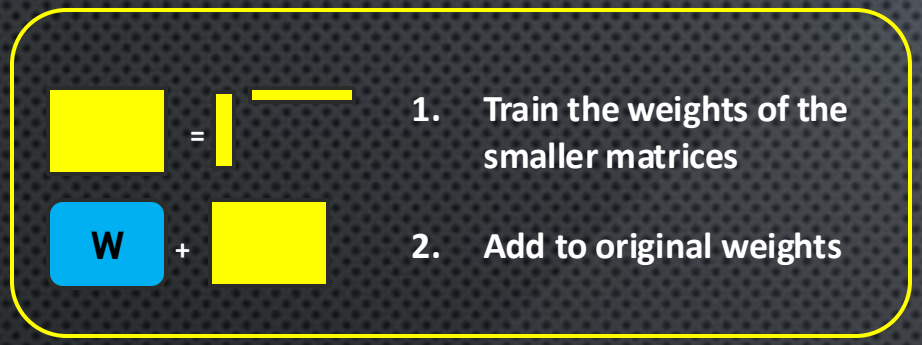
- **LoRA (Low Rank Adaption) - Reparameterization Technique**
- **Adapters**
- **Soft prompts**

This paper proposes a parameter-efficient fine-tuning method that makes use of low-rank decomposition matrices to reduce the number of trainable parameters needed for fine-tuning language models.



[11] Lialin V, Deshpande V, Rumshisky A. **Scaling down to scale up: A guide to parameter-efficient fine-tuning**. arXiv preprint arXiv:2303.15647. 2023.

- LoRA (Low Rank Adaption) - Reparameterization Technique
- Adapters
- Soft prompts



*[12] Hu EJ, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. LoRA: Low-Rank Adaptation of Large Language Models. InInternational Conference on Learning Representations 2021 Oct 6. (**Microsoft**)*

Section 6: Parameter Efficient Fine Tuning (PEFT)

- **LoRA (Low Rank Adaption) - Reparameterization Technique**
- **Adapters**
- **Soft prompts**

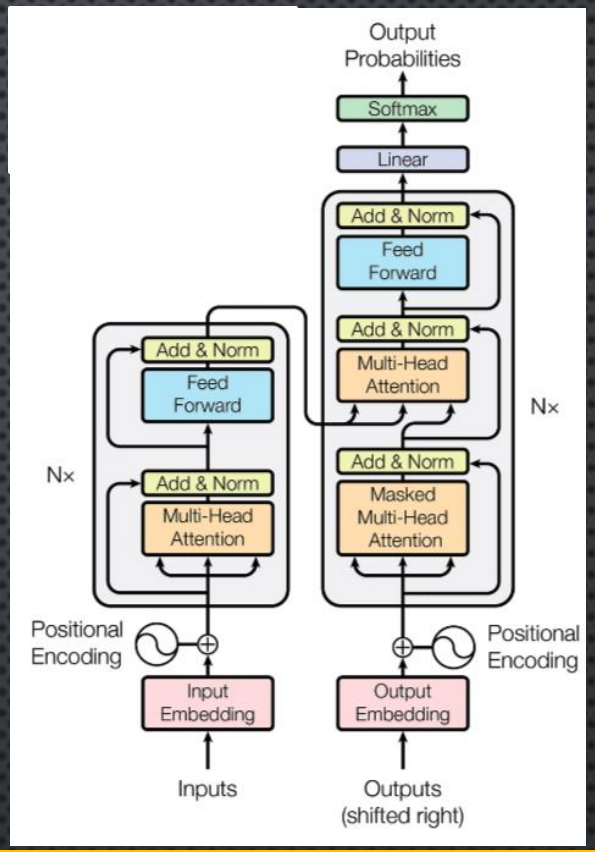
- **Transformer in Attention is all you need:**
  - Weights dimensions d x k = 512 x 64 = 32768 params

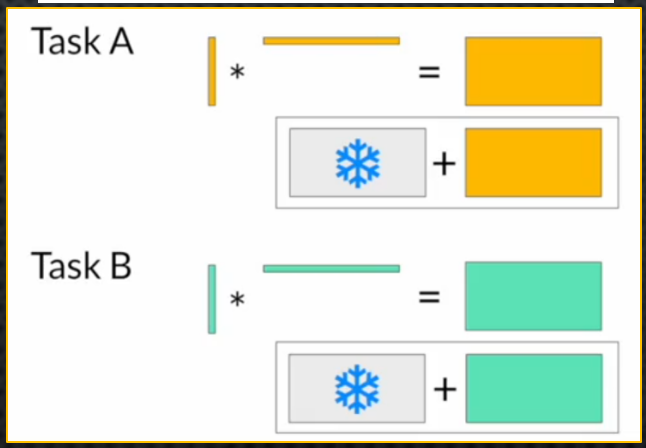
- **Lora with Rank r = 8**
  - Matrix A r x k = 8 x 64 = 512 params
  - Matrix B d x r = 512 x 8 = 4096 params
  - **86% reduction in parameters to train.**

Rank r	val_loss	BLEU	NIST	METEOR	ROUGE_L	CIDEr
1	1.23	68.72	8.7215	0.4565	0.7052	2.4329
2	1.21	69.17	8.7413	0.4590	0.7052	2.4639
4	1.18	**70.38**	**8.8439**	**0.4689**	0.7186	**2.5349**
8	1.17	69.57	8.7457	0.4636	**0.7196**	2.5196
16	**1.16**	69.61	8.7483	0.4629	0.7177	2.4985
32	**1.16**	69.33	8.7736	0.4642	0.7105	2.5255
64	**1.16**	69.24	8.7174	0.4651	0.7180	2.5070
128	**1.16**	68.73	8.6718	0.4628	0.7127	2.5030
256	**1.16**	68.92	8.6982	0.4629	0.7128	2.5012
512	**1.16**	68.78	8.6857	0.4637	0.7128	2.5025
1024	1.17	69.37	8.7495	0.4659	0.7149	2.5090

Very parameter efficient strategy

Task adaptation: Switch out small matrices at inference time to change the task



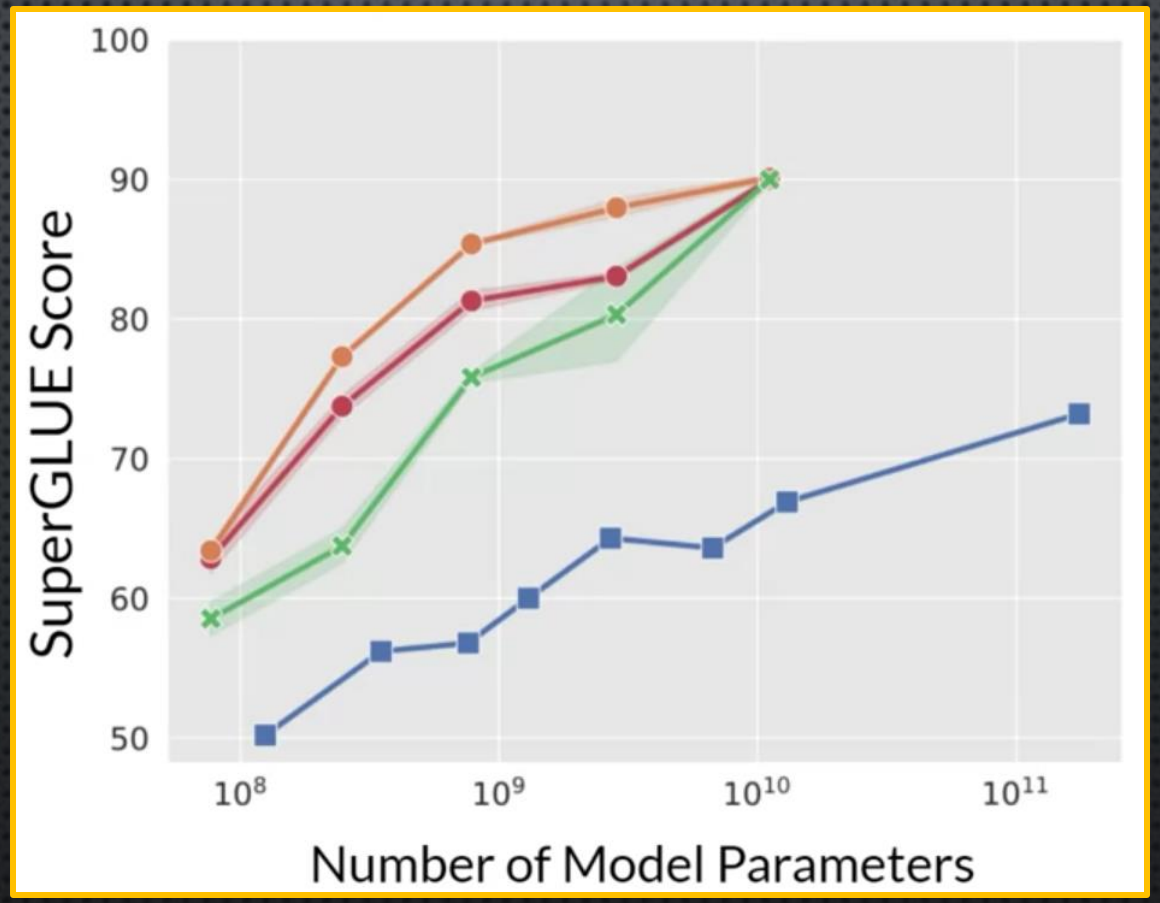


*[12] Hu EJ, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. LoRA: Low-Rank Adaptation of Large Language Models. InInternational Conference on Learning Representations 2021 Oct 6. (**Microsoft**)*

Section 6: Parameter Efficient Fine Tuning (PEFT)

- **LoRA (Low Rank Adaption)**
- **Adapters (change the architecture, adding more trainable layers)**
- **Soft prompts (keep the model architecture fixed and add trainable parameters to the embedding vectors)**

The paper explores "prompt tuning," a method for conditioning language models with learned soft prompts, achieving competitive performance compared to full fine-tuning and enabling model reuse for many tasks

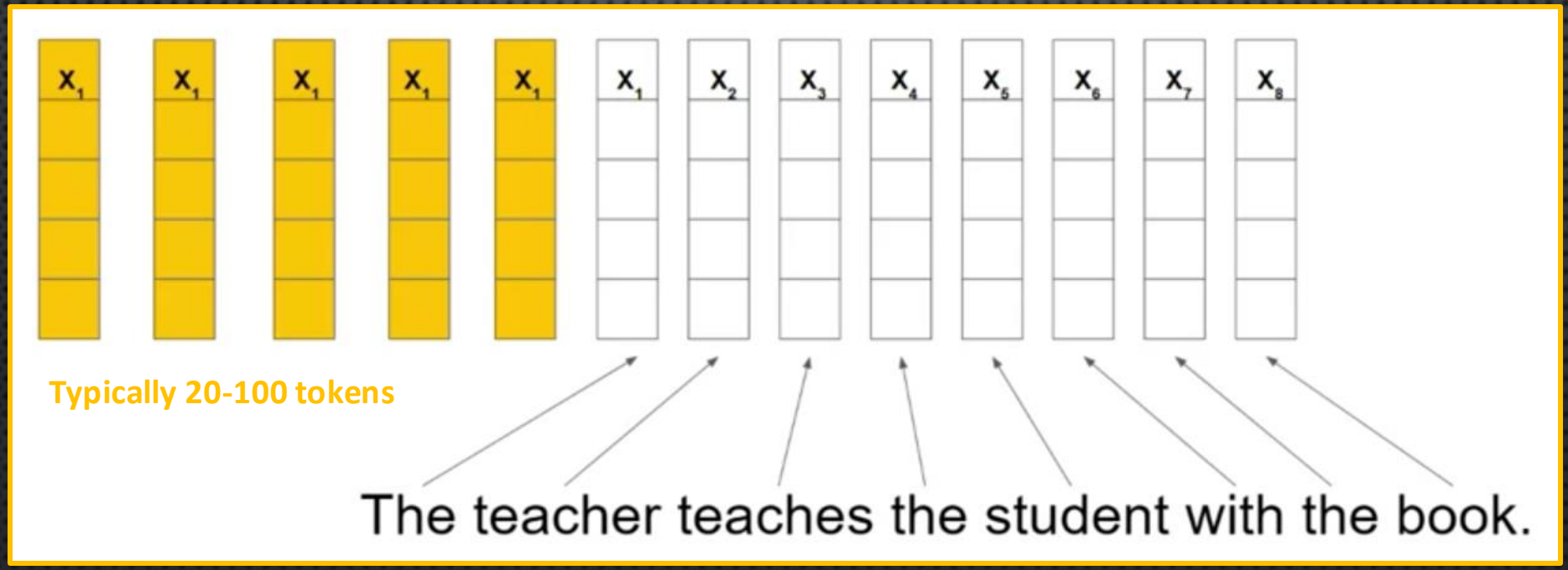


Full fine-tuning
Multi-task Fine-tuning
Sof-Prompt tuning
Prompt engineering

*[13] Lester B, Al-Rfou R, Constant N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. InProceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021 Nov (pp. 3045-3059).*

Section 6: Parameter Efficient Fine Tuning (PEFT)

- **LoRA (Low Rank Adaption)**
- **Adapters (change the architecture, adding more trainable layers)**
- **Soft prompts (architecture is not changed)**

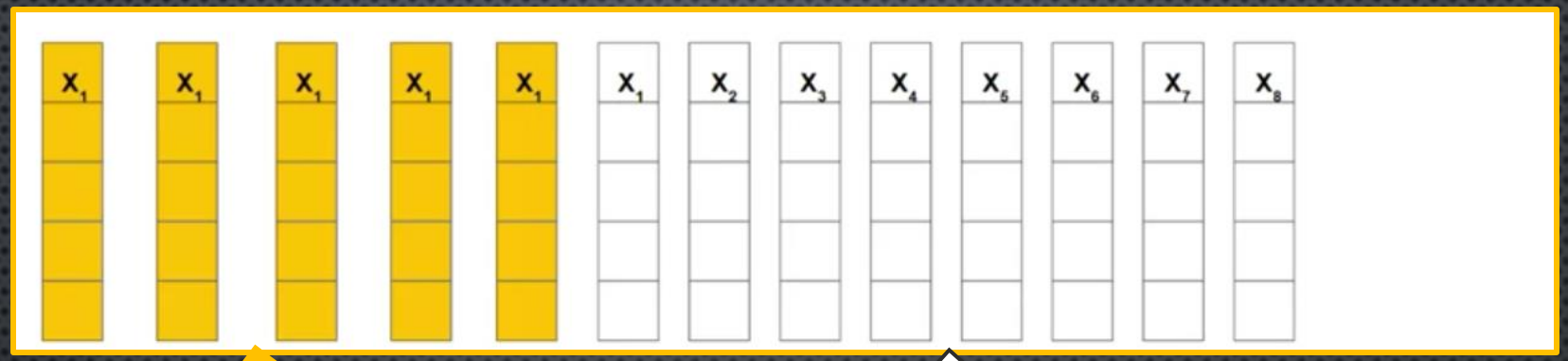


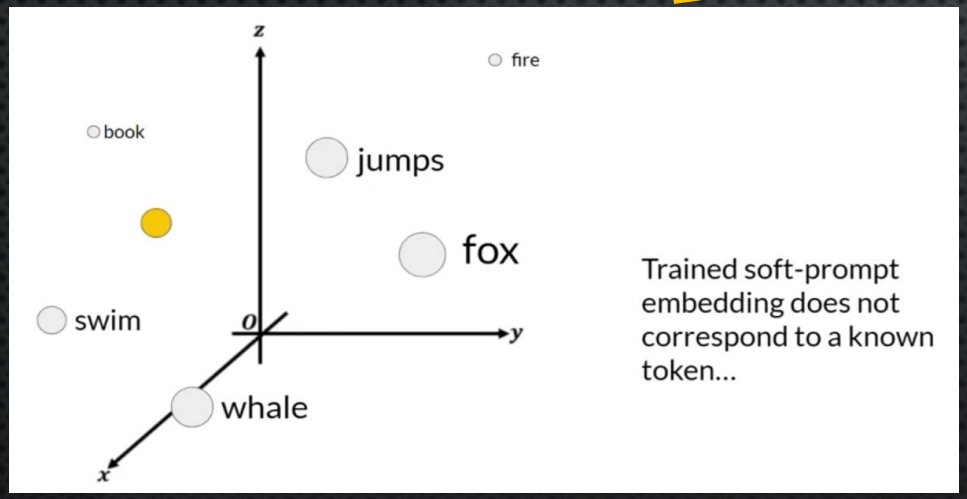
*[13] Lester B, Al-Rfou R, Constant N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. InProceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021 Nov (pp. 3045-3059).*

Section 6: Parameter Efficient Fine Tuning (PEFT)

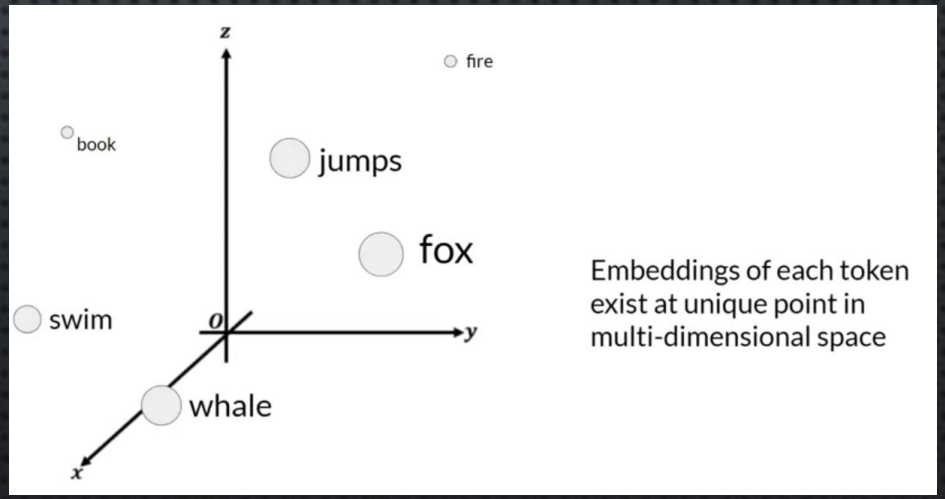
- LoRA (Low Rank Adaption)
- Adapters
- Soft prompts

They can be any values

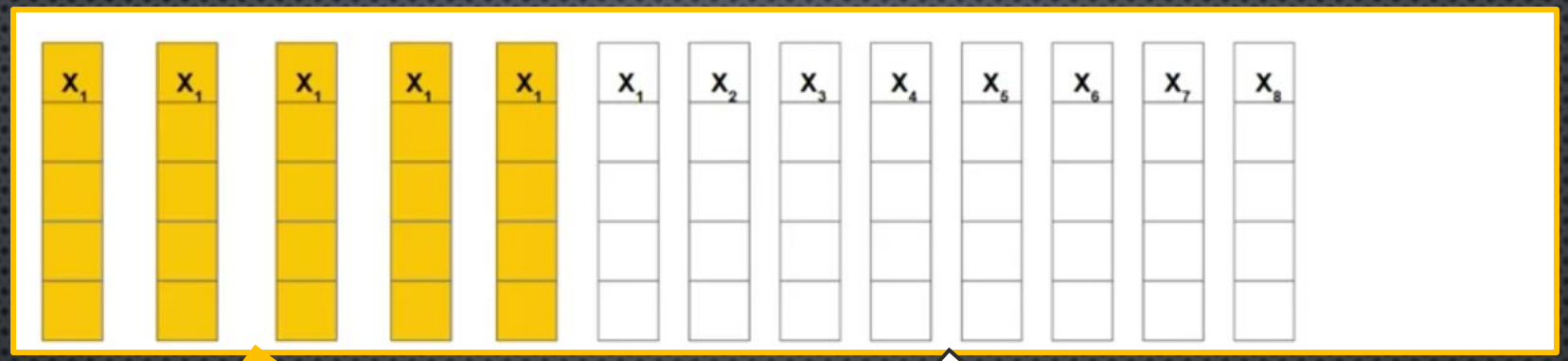


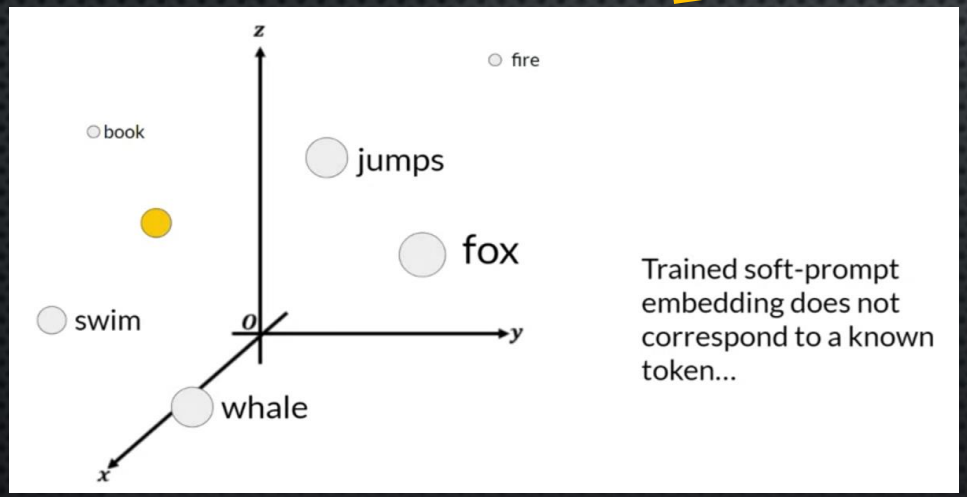


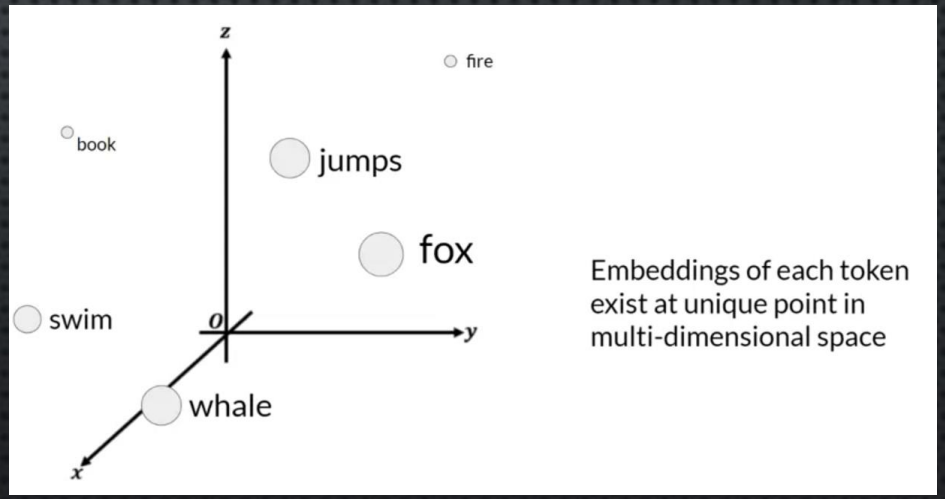
Very parameter efficient strategy



*[13] Lester B, Al-Rfou R, Constant N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. InProceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021 Nov (pp. 3045-3059).*



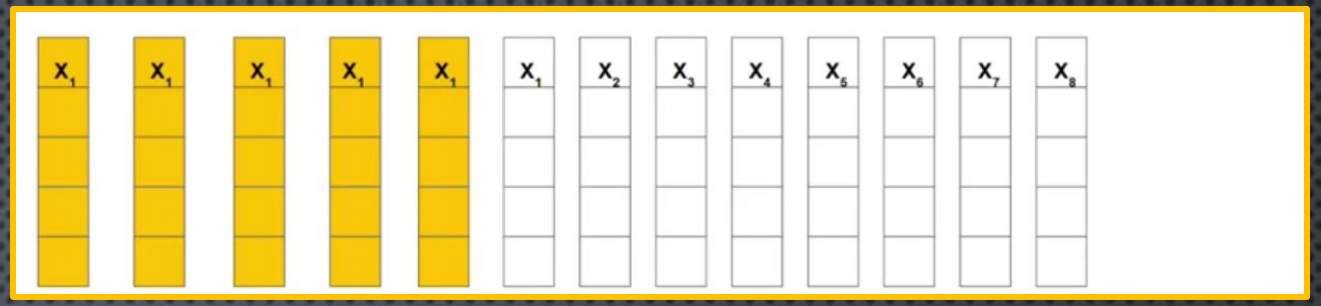




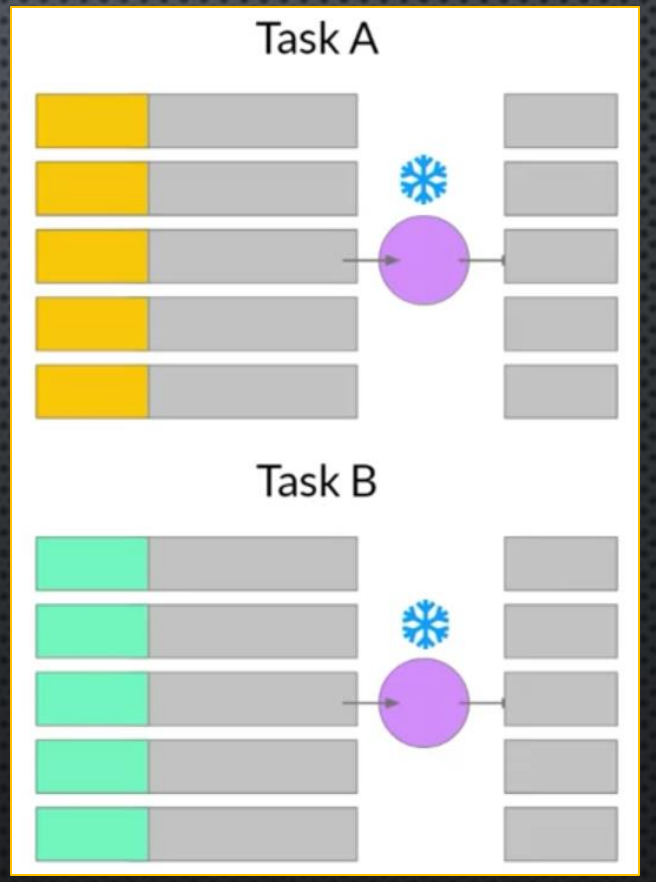
*[13] Lester B, Al-Rfou R, Constant N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. InProceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021 Nov (pp. 3045-3059).*

Section 6: Parameter Efficient Fine Tuning (PEFT)

- LoRA (Low Rank Adaption)
- Adapters (+Layers in Encoder/Decoder)
- Soft prompts



Task adaptation like LoRA
Switch out soft prompt at inference time to change the task



*[13] Lester B, Al-Rfou R, Constant N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. InProceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021 Nov (pp. 3045-3059).*

References Section 5: Fine-Tuning and Domain Adaptation

[9] Wu S, Irsoy O, Lu S, Dabravolski V, Dredze M, Gehrmann S, Kambadur P, Rosenberg D, Mann G. **Bloomberggpt: A large language model for finance**. arXiv preprint arXiv:2303.17564. 2023 Mar 30.

*[2] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. **2024**;25(70):1-53.*

*[11] Lialin V, Deshpande V, Rumshisky A. **Scaling down to scale up: A guide to parameter-efficient fine-tuning**. arXiv preprint arXiv:2303.15647. 2023 Mar 28.*

*[12] Hu EJ, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. **LoRA: Low-Rank Adaptation of Large Language Models**. InInternational Conference on Learning Representations 2021 Oct 6.*

*[13] Lester B, Al-Rfou R, Constant N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. InProceedings of the 2021 Conference on Empirical Methods in Natural Language Processing 2021 Nov (pp. 3045-3059).*

WE ARE GOING TO TALK ABOUT…

1. **Beyond Basic LLMs**: Advances Toward Sophisticated Applications
    - Retrieval-Augmented Generation (RAG)
    - Program-Aided Language Models (PAL)
    - Integration of Reasoning and Action (ReAct)
    - Architectural Designs for LLM Applications (ChatGPT, Gemini, etc..)
2. **Introduction: Transformers**
3. **Selecting LLM Architectures**: Determining the Optimal Model for Your Needs
4. **LLM Training Resources**: GPU Memory Requirements
    - Models: BERT-L (340M), GPT-2 (1.5B), LLaMA-2 (7-13-70B), GPT-3 (175B), PaLM (540B)
    - GPU RAM for Models:
        - 1B parameters: 24GB (32-bit precision)
        - 175B parameters: 4200GB (32-bit precision)
        - 500B parameters: 12000GB (32-bit precision)
5. **Datasets & Benchmarks**: Criteria for Selecting Evaluation Metrics and Datasets
6. **Fine-Tuning Strategies**: Addressing Specific Tasks and Preventing Overfitting
7. **RLHF**: Enhancing LLMs Through Human Interaction
8. **ReST (Google)**: Reinforced Self-Training for Language Modeling
9. MED-Gemini

Large Language Models behaving badly:

- RLHF is a fine-tuning process that aligns LLMs with Human Feedbacks

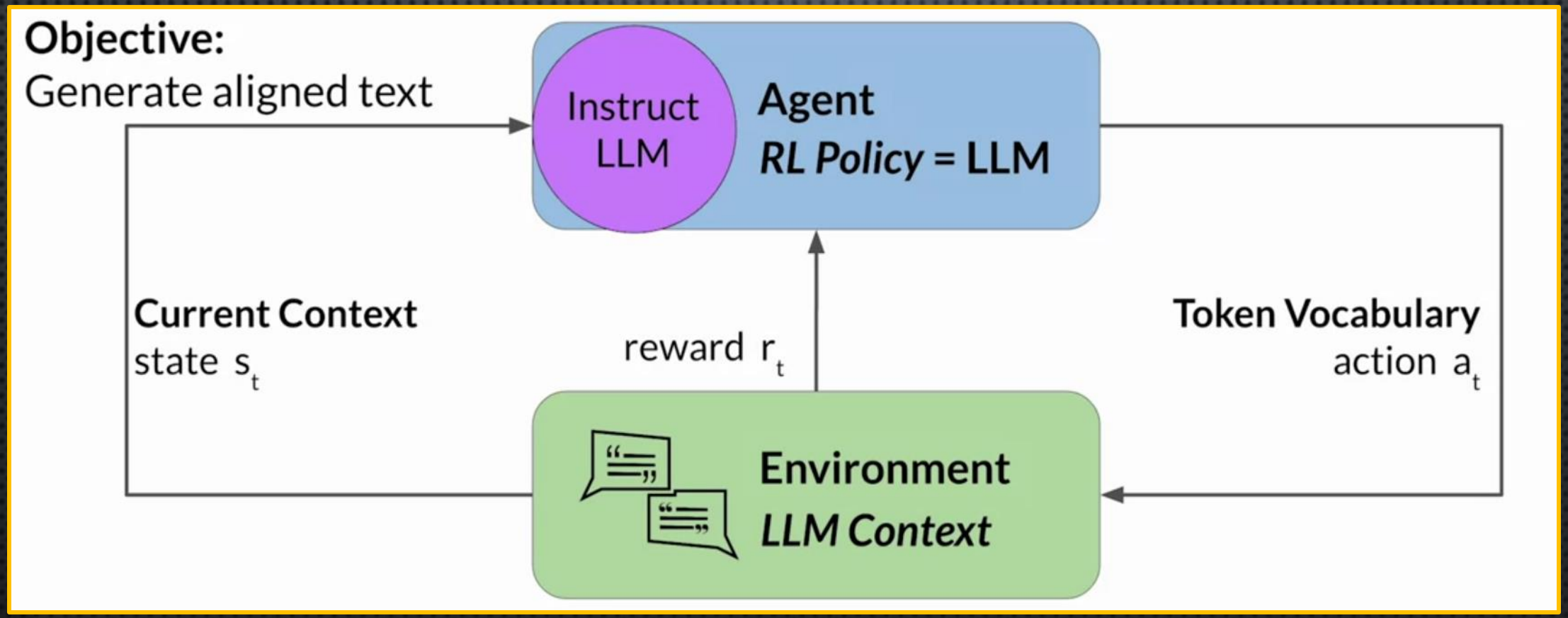
- Training or Fine tuning an LLM is not enough
  - Toxic language
  - Aggressive responses
  - Providing dangerous information
  - Etc...

- **Responsible Use of AI: The HHH**
  - **Helpful?**
  - **Honest?**
  - **Harmless?**

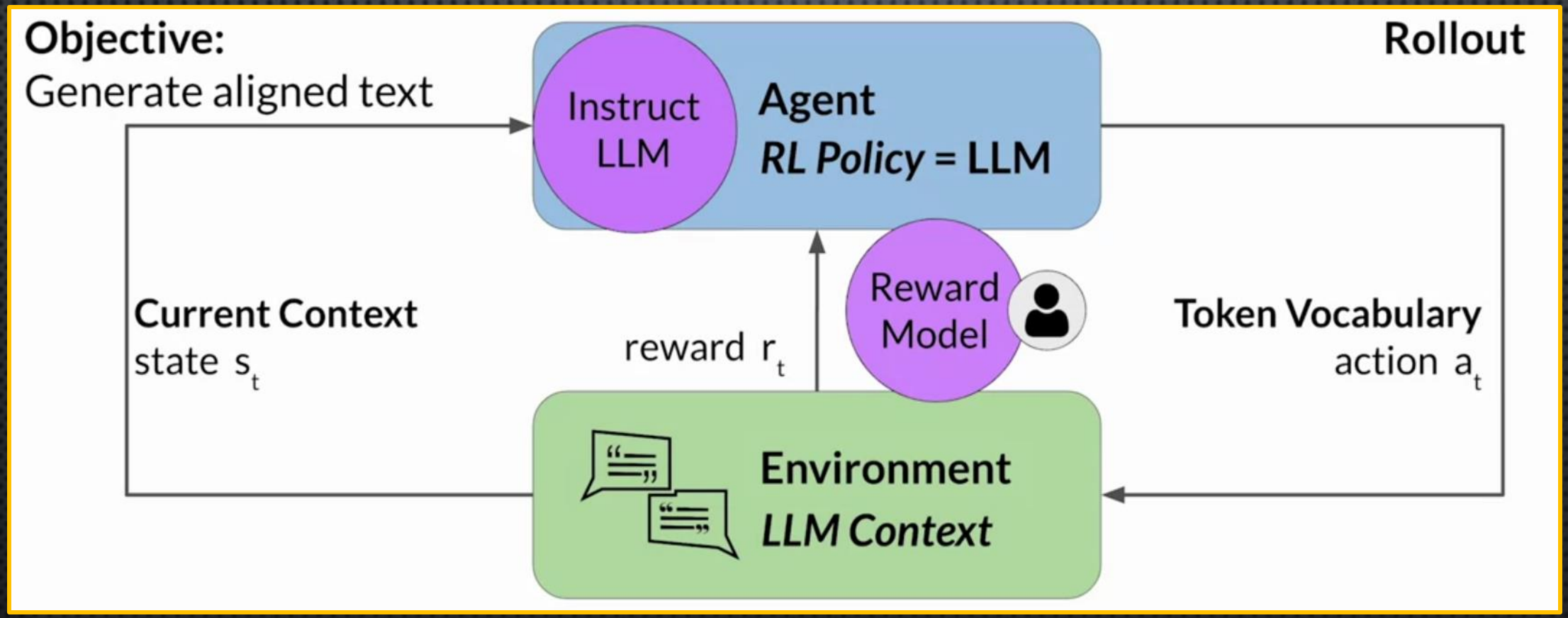
- **LLMs need to be aligned with human feedback**

In tasks like:
- **Automatic Reporting**
- **Question Answering**
- **Summarization**
- **Etc...**

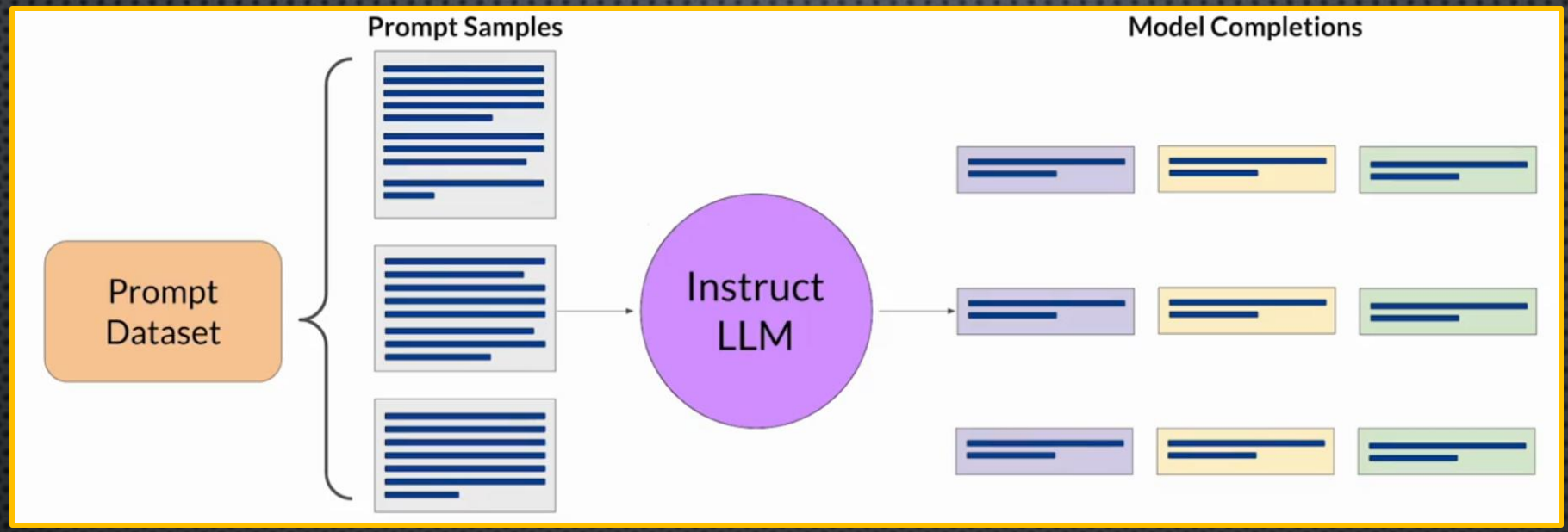
RLHF can minimize harmfulness and maximize helpfulness



Reinforcement learning: fine-tune LLMs



Create a prompt dataset based on the task (summarization, Q/A, …)



LLMs are generative models, they output different completions at every run.

1. **Definition of Scores**: Define what each score (0, 1, 2, ..., N) represents, ensuring that these definitions are precise and relate directly to the quality attributes of the completion (e.g., relevance, correctness, informativeness).

2. **Criteria Specificity**: Criteria should be detailed enough to cover different aspects of the completion such as factual accuracy, fluency, coherence, and alignment with the prompt.

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

Create a prompt dataset based on the task (summarization, Q/A, …)

- **Train Labelers**
  - **Initial Training**: Conduct thorough training sessions where labelers are educated on the scoring system and the specific criteria for each score.
  - **Calibration Exercises**: Use calibration exercises to align labeler perceptions and interpretations of the scoring criteria, ensuring consistency in how scores are applied.
- **Multiple Evaluations**: Each completion is rated by multiple labelers to mitigate individual bias and variance in scoring.
- **Averaging Scores**: After all labelers have scored a completion, calculate the average score for each completion. This average becomes the GT for the model output.
- **Handling Outliers**: Consider statistical methods to identify and handle outlier scores that might skew the average, ensuring that the GT represents a consensus view.
- **Consistency Review**: Regularly review the scores from different labelers to check for consistency and alignment with the training provided.
- **Feedback Loops**: Set up feedback loops where labelers can discuss discrepancies and refine their understanding of the criteria.

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

Create a prompt dataset based on the task (summarization, Q/A, …)

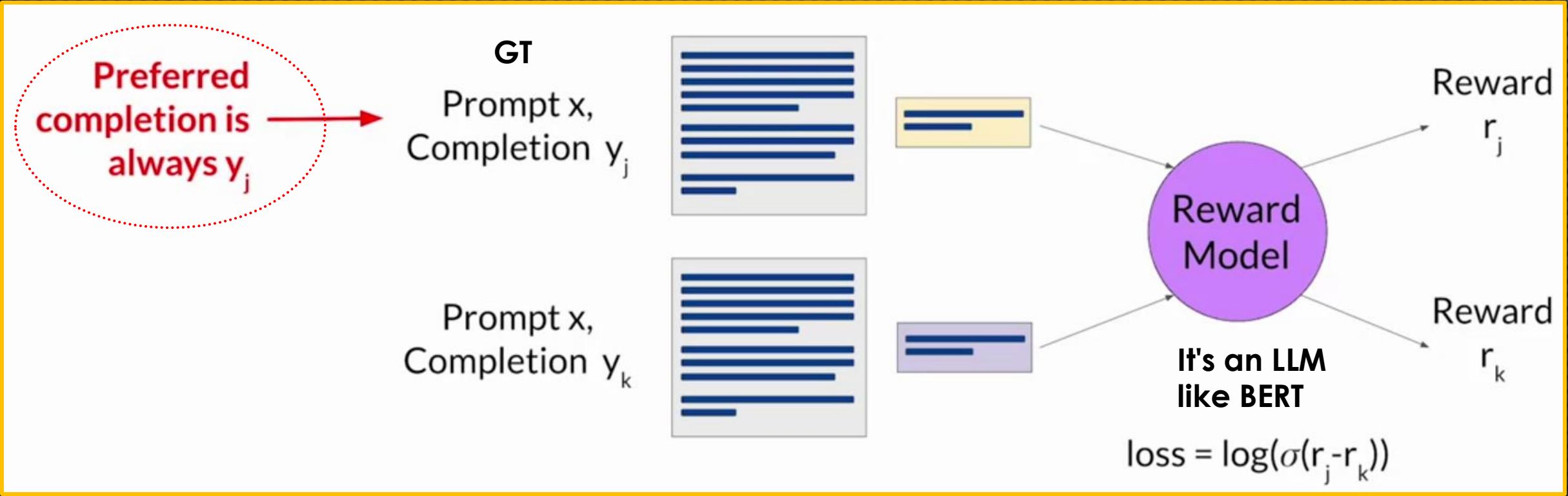
- **Feedback and Iterative Improvement**
  - Labeler Feedback: Collect and analyze feedback from labelers on the scoring process and criteria clarity.
  - Guideline Adjustments: Update and refine the scoring guidelines and training based on labeler feedback and observed scoring trends.
- **Integration with Reward Module**
  - **Data Preparation**: Format the averaged scores along with their corresponding completions in a way that can be easily ingested by the reward module.
  - **Reward Training**: Use the GT scores as input to train the reward module, which learns to predict the quality of new completions based on historical labeled data.

Once human labelers have completed their assessments off the *prompt-completion* sets, all the data needed to train the reward model is done.

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

Reward Model

Train model to predict preferred completion from {y_j, y_k} for prompt x



- **The loss function log(sigmoid(rj−rk)), is used in learning to rank.**
- **Reward model can be used as a binary classifier, the reward is the logit value of the positive class.**

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

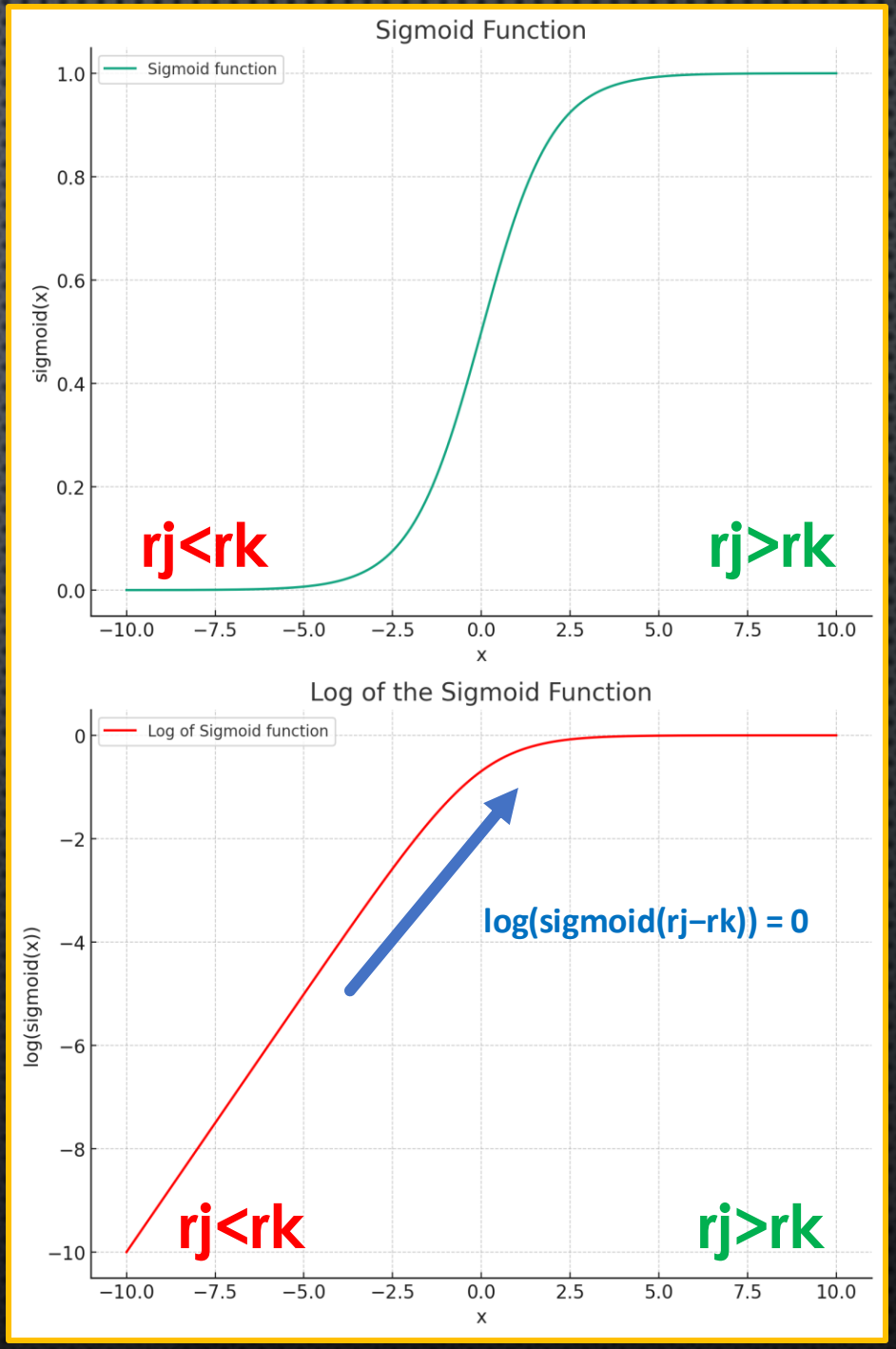
Reward Model

The loss function

$$r_j - r_k$$

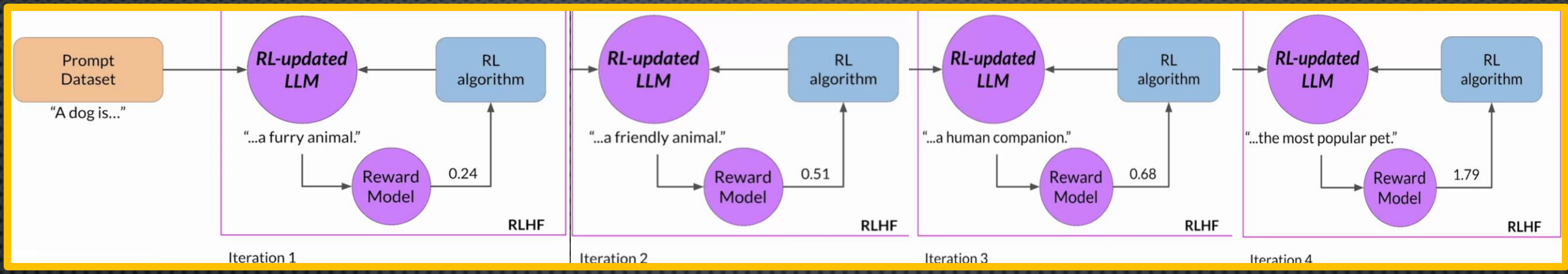
$$\log(\text{sigmoid}(r_j - r_k))$$

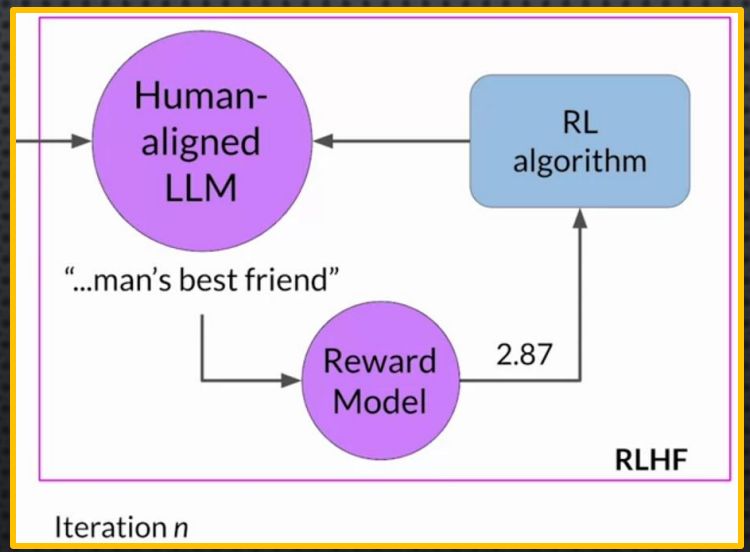
is used in scenarios like learning to rank, where the goal is to ensure that one item **rj** is ranked higher than another item **rk**.



[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

Reward Model to fine-tune LLM with RL



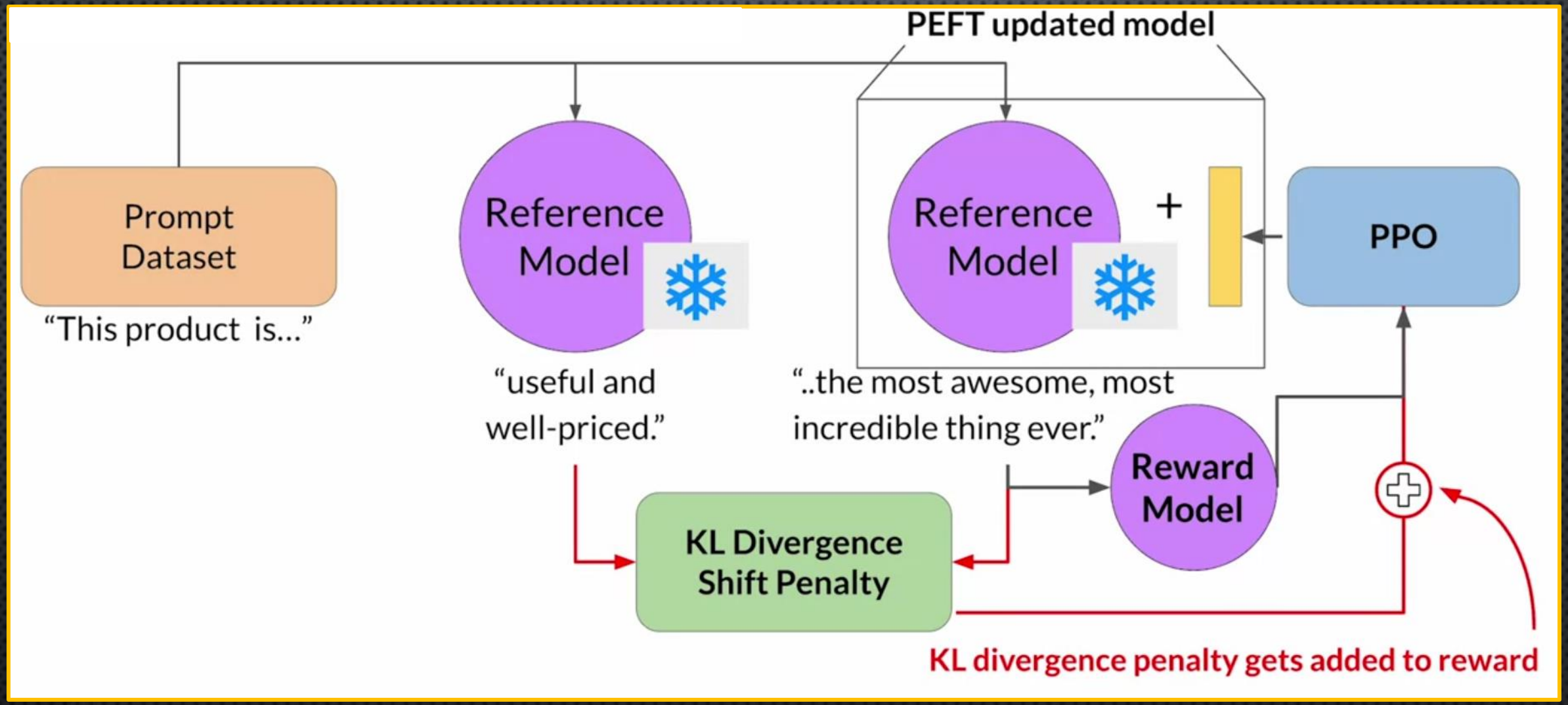


RL Algorithm is usually the PPO
(Proximal Policy Optimization)

Creates small updates for the model in a way that the reward is maximized.

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

[16] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. **Proximal policy optimization algorithms**. arXiv preprint arXiv:1707.06347. 2017 Jul 20.



In order to avoid divergence from the original LLM, the KL-div can be added to the reward model to penalize rewards that cause high divergence, moreover, the PEFT can be used for fine-tuning in order to keep the original LLM and specialize the model on different tasks.

References Section 7: Reinforcement learning from human feedback (RLHF)

[14] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. **2024**;25(70):1-53.

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

[16] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. **Proximal policy optimization algorithms**. arXiv preprint arXiv:1707.06347. 2017 Jul 20.

[17] Gulcehre C, Paine TL, Srinivasan S, Konyushkova K, Weerts L, Sharma A, Siddhant A, Ahern A, Wang M, Gu C, Macherey W. **Reinforced self-training (rest) for language modeling**. **Google Research**. arXiv preprint arXiv:2308.08998. 2023 Aug 17.

WE ARE GOING TO TALK ABOUT…

1. **Beyond Basic LLMs**: Advances Toward Sophisticated Applications
    - Retrieval-Augmented Generation (RAG)
    - Program-Aided Language Models (PAL)
    - Integration of Reasoning and Action (ReAct)
    - Architectural Designs for LLM Applications (ChatGPT, Gemini, etc..)
2. **Introduction: Transformers**
3. **Selecting LLM Architectures**: Determining the Optimal Model for Your Needs
4. **LLM Training Resources**: GPU Memory Requirements
    - Models: BERT-L (340M), GPT-2 (1.5B), LLaMA-2 (7-13-70B), GPT-3 (175B), PaLM (540B)
    - GPU RAM for Models:
        - 1B parameters: 24GB (32-bit precision)
        - 175B parameters: 4200GB (32-bit precision)
        - 500B parameters: 12000GB (32-bit precision)
5. **Datasets & Benchmarks**: Criteria for Selecting Evaluation Metrics and Datasets
6. **Fine-Tuning Strategies**: Addressing Specific Tasks and Preventing Overfitting
7. **RLHF**: Enhancing LLMs Through Human Interaction
8. **ReST (Google)**: Reinforced Self-Training for Language Modeling
9. MED-Gemini

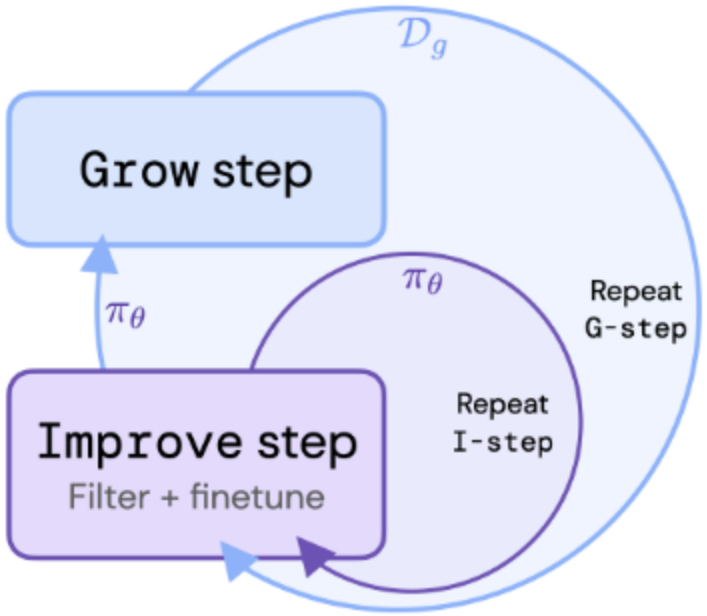
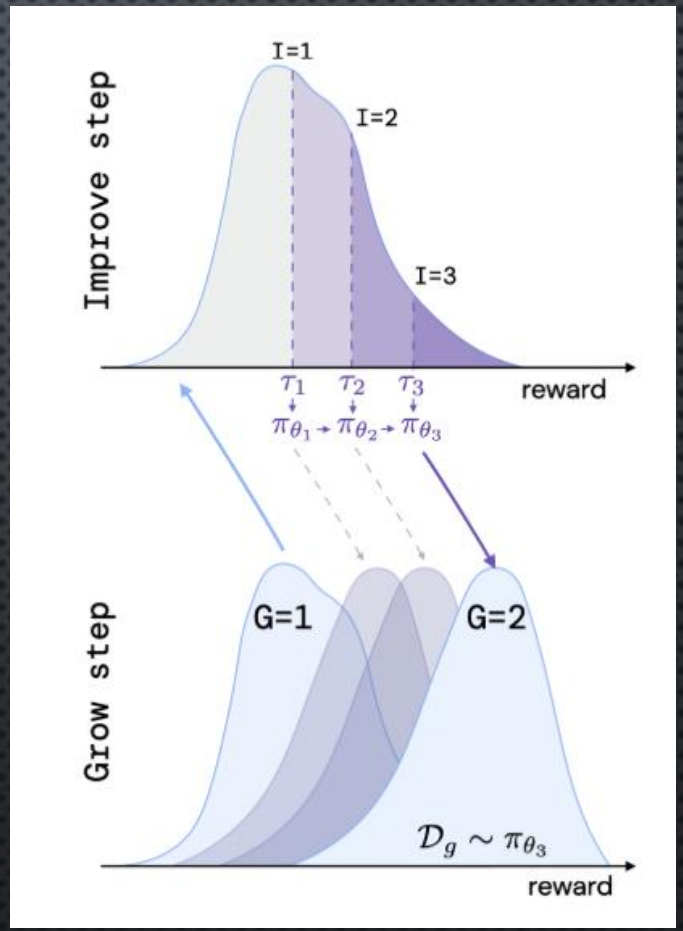


Figure 1 | ***ReST* method.** During `Grow` step, a policy generates a dataset. At `Improve` step, the filtered dataset is used to fine-tune the policy. Both steps are repeated, `Improve` step is repeated more frequently to amortise the dataset creation cost.

Given an initial LLM policy, ReST produces a dataset by generating samples from the policy, which are then used to improve the LLM policy using offline RL algorithms. ReST is more efficient than typical online RLHF methods because the training dataset is produced offline.



[25] Gulcehre C, Paine TL, Srinivasan S, Konyushkova K, Weerts L, Sharma A, Siddhant A, Ahern A, Wang M, Gu C, Macherey W. **Reinforced self-training (rest) for language modeling**. **Google Research &** DeepMind. arXiv preprint arXiv:2308.08998. 2023.

$$\pi_\theta(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{t=1}^{T} \pi_\theta(y_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{x}),$$

$$\mathcal{L}_{\mathrm{NLL}}(\theta) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}} \left[\sum_{t=1}^{T} \log \pi_\theta(y_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{x}) \right]$$

[25] Gulcehre C, Paine TL, Srinivasan S, Konyushkova K, Weerts L, Sharma A, Siddhant A, Ahern A, Wang M, Gu C, Macherey W. **Reinforced self-training (rest) for language modeling**. **Google Research &** DeepMind. arXiv preprint arXiv:2308.08998. 2023.

Algorithm 1: *ReST* algorithm. *ReST* is a growing-batch RL algorithm. Given an initial policy of reasonable quality (for example, pre-trained using BC) iteratively applies `Grow` and `Improve` steps to update the policy. Here F is a filtering function, and $\mathcal{L}$ is an loss function.

Input: $\mathcal{D}$: Dataset, $\mathcal{D}_{eval}$: Evaluation dataset, $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \theta)$: loss, $R(\boldsymbol{x}, \boldsymbol{y})$: reward model, G: number of grow steps, I: number of improve steps, N: number of samples per context

Train π_θ on $\mathcal{D}$ using loss $\mathcal{L}$.

for $g = 1$ *to* G **do**

  `// Grow`

  Generate dataset $\mathcal{D}_g$ by sampling: $\mathcal{D}_g = \{ (\boldsymbol{x}^i, \boldsymbol{y}^i)|_{i=1}^{N_g} \text{ s.t. } \boldsymbol{x}^i \sim \mathcal{D},\ \boldsymbol{y}^i \sim \pi_\theta(\boldsymbol{y}|\boldsymbol{x}^i) \} \cup \mathcal{D}$.

  Annotate $\mathcal{D}_g$ with the reward model $R(\boldsymbol{x}, \boldsymbol{y})$.

  for $i = 1$ *to* I **do**

    `// Improve`

    Choose threshold s.t. $\tau_1 > V_{\pi_\theta}$ for $V_{\pi_\theta} = \mathbb{E}_{\mathcal{D}_g}[R(\boldsymbol{x}, \boldsymbol{y})]$ and $\tau_{i+1} > \tau_i$.

    while *reward improves on* $\mathcal{D}_{eval}$ **do**

      Optimise θ on objective: $J(\theta) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_g} [F(\boldsymbol{x}, \boldsymbol{y}; \tau_i)\ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}; \theta)]$

    end

  end

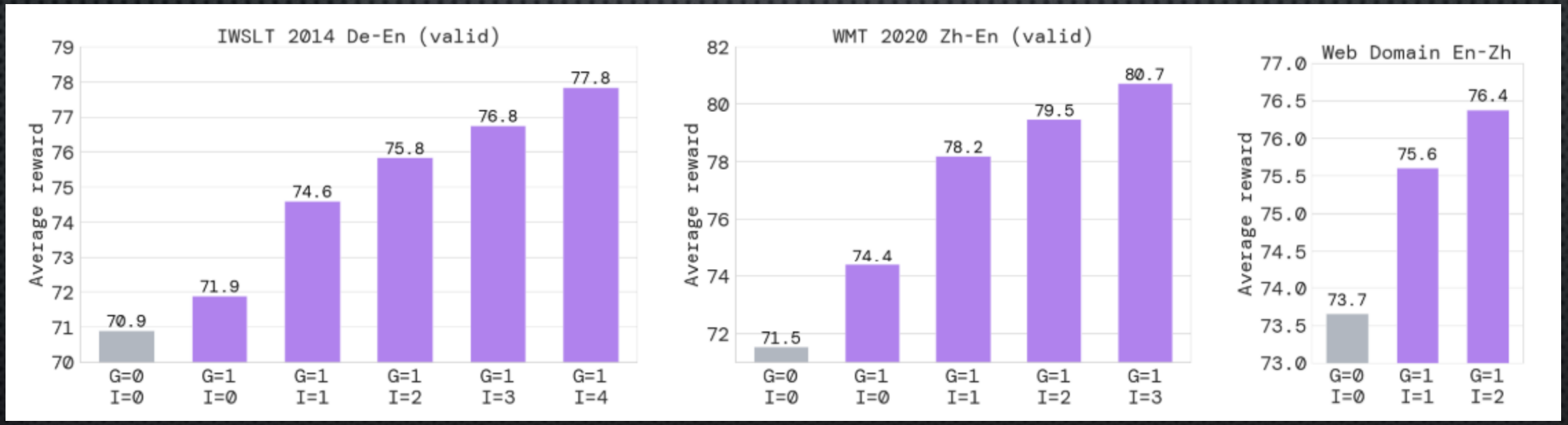
end

Output: Policy π_θ

[25] Gulcehre C, Paine TL, Srinivasan S, Konyushkova K, Weerts L, Sharma A, Siddhant A, Ahern A, Wang M, Gu C, Macherey W. **Reinforced self-training (rest) for language modeling**. **Google Research &** DeepMind. arXiv preprint arXiv:2308.08998. 2023.

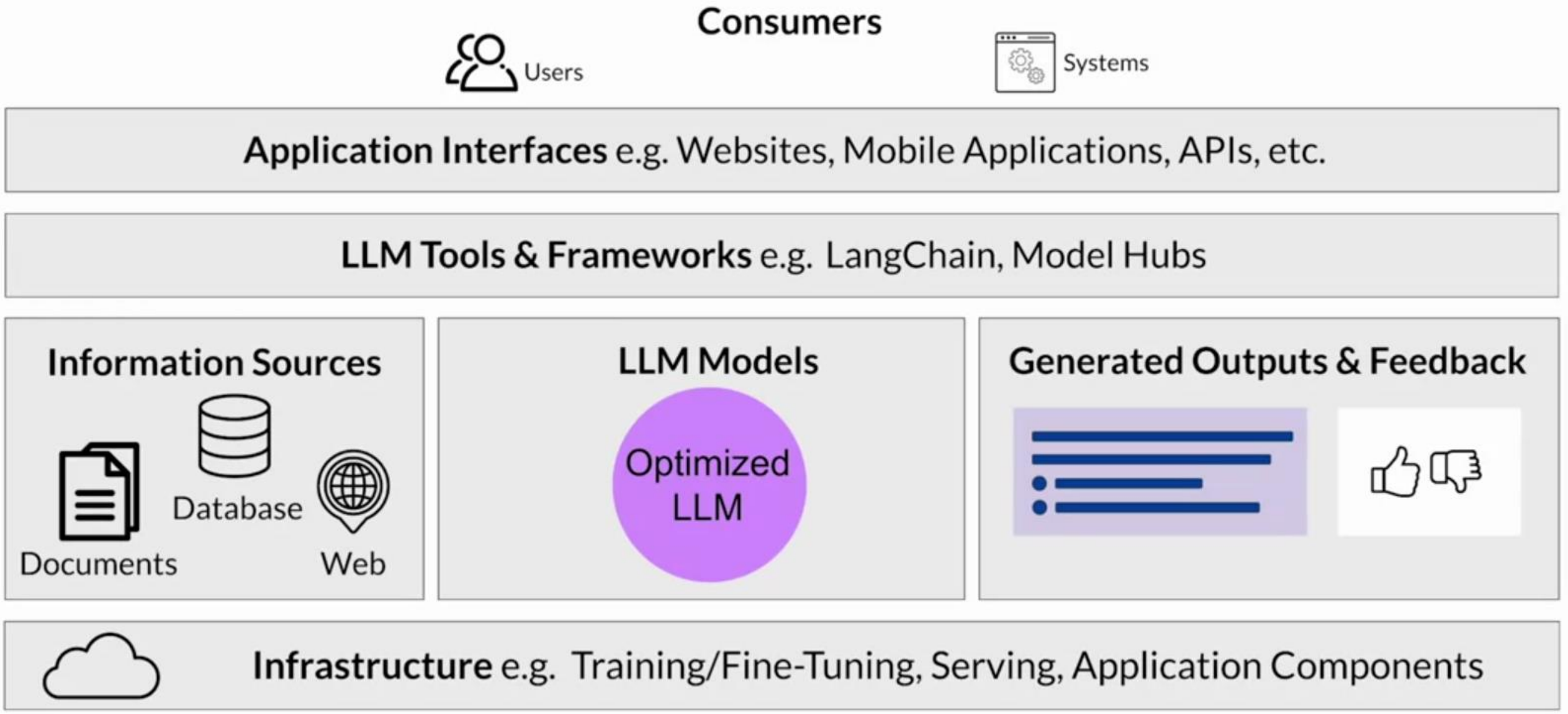
Algorithm	Average Reward	Distinct samples
BC (G=0, I=0)	70.9	16 000 000
ReST (G=1, I=0)	71.9	16 000 000
ReST (G=1, I=4)	77.8	16 000 000
***ReST* (G=2, I=3)**	**83.1**	32 000 000
Online RL	71.6	24 000 000

Table 1 | **Online RL for IWSLT 2014:** Online RL performs as well as *ReST* (G=1, I=0) and *ReST* (G=1, I=4) is significantly better.

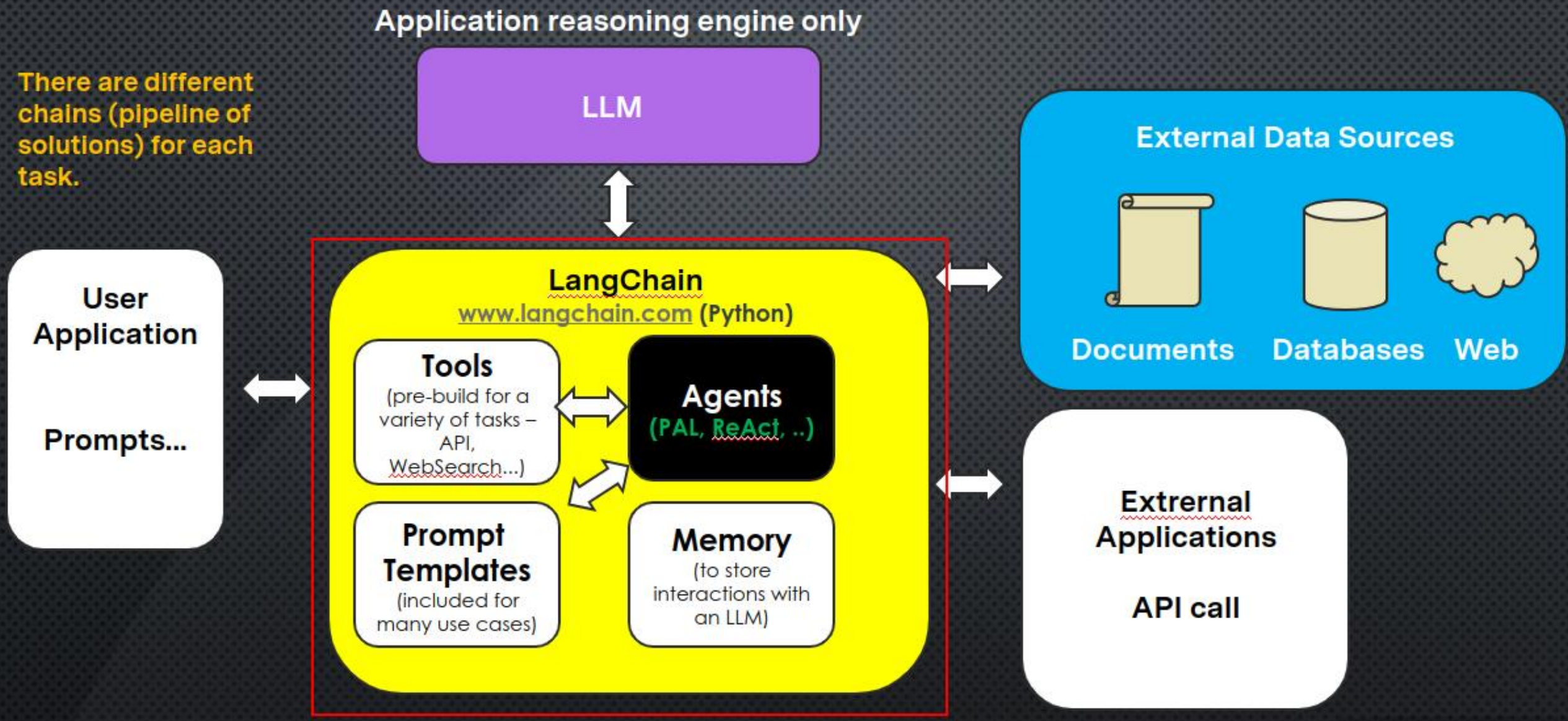


[25] Gulcehre C, Paine TL, Srinivasan S, Konyushkova K, Weerts L, Sharma A, Siddhant A, Ahern A, Wang M, Gu C, Macherey W. **Reinforced self-training (rest) for language modeling**. **Google Research &** DeepMind. arXiv preprint arXiv:2308.08998. 2023.

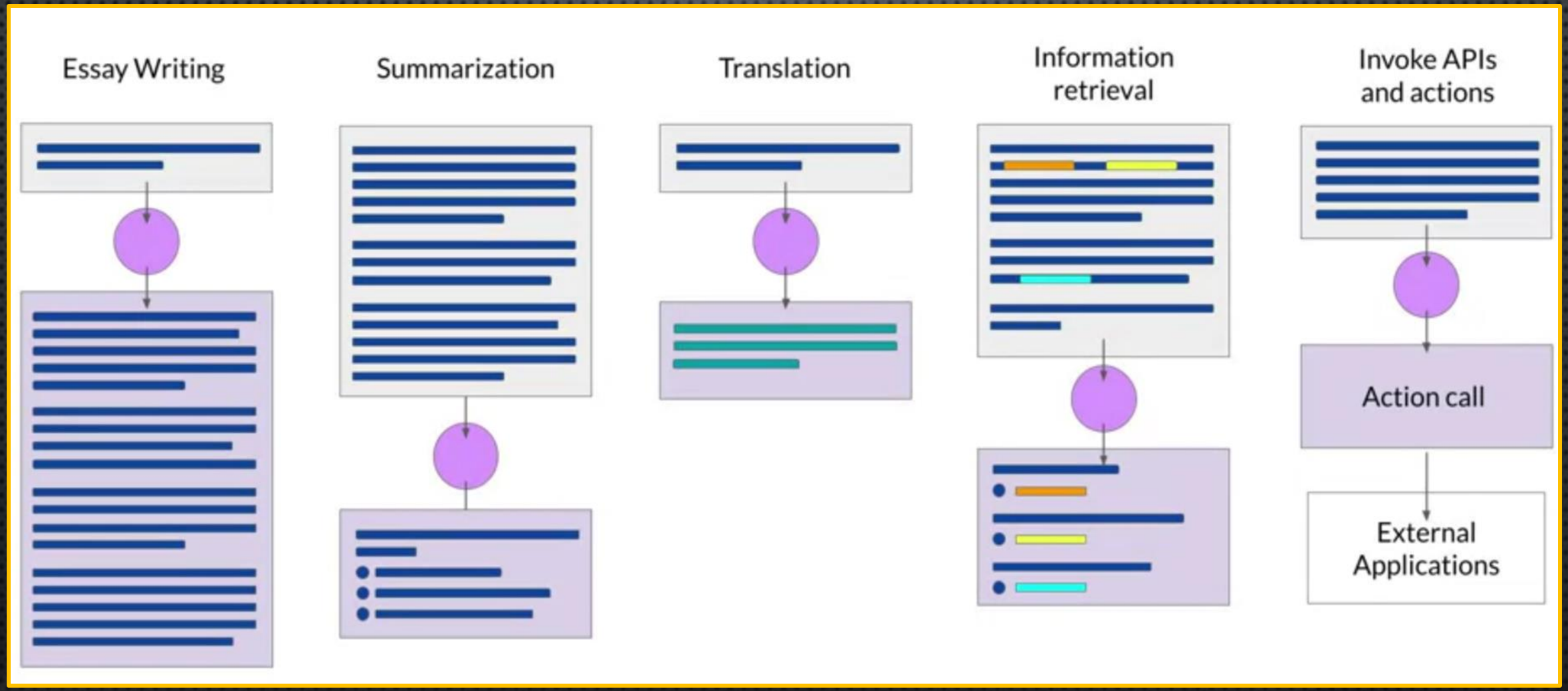
Summary


Consumers
Users
Systems
Application Interfaces e.g. Websites, Mobile Applications, APIs, etc.
LLM Tools & Frameworks e.g. LangChain, Model Hubs
Information Sources
Database
Documents
Web
LLM Models
Optimized LLM
Generated Outputs & Feedback
Infrastructure e.g. Training/Fine-Tuning, Serving, Application Components

There are different chains (pipeline of solutions) for each task.



Summary


Essay Writing
Summarization
Translation
Information retrieval
Invoke APIs and actions
Action call
External Applications

Summary

	Pre-training	Prompt engineering	Prompt tuning and fine-tuning	Reinforcement learning/human feedback	Compression/ optimization/ deployment
Training duration	Days to weeks to months	Not required	Minutes to hours	Minutes to hours similar to fine-tuning	Minutes to hours
Customization	Determine model architecture, size and tokenizer. Choose vocabulary size and # of tokens for input/context Large amount of domain training data	No model weights Only prompt customization	Tune for specific tasks Add domain-specific data Update LLM model or adapter weights	Need separate reward model to align with human goals (helpful, honest, harmless) Update LLM model or adapter weights	Reduce model size through model pruning, weight quantization, distillation Smaller size, faster inference
Objective	Next-token prediction	Increase task performance	Increase task performance	Increase alignment with human preferences	Increase inference performance

References

Special thanks to Shelbee Eigenbrode, Antje Barth, and Mike Chambers for their work on "Generative AI with Large Language Models" (Coursera, Amazon AWS, 2023), which significantly informed and inspired the material used in these slides.

Resources:

The Tasks, Datasets and benchmarks for LLMs:

- GLUE
- SuperGLUE
- HELM
- MMLU (Massive Multitask Language Understanding)
- BIG-bench
- BIG-bench Hard
- Lite

[2] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. 2024;25(70):1-53.

[4] Wang A, Singh A, Michael J, Hill F, Levy O, Bowman SR. **GLUE: A multi-task benchmark and analysis platform for natural language understanding**. arXiv preprint arXiv:1804.07461. 2018 Apr 20.

[5] Wang A, Pruksachatkun Y, Nangia N, Singh A, Michael J, Hill F, Levy O, Bowman S. **Superglue: A stickier benchmark for general-purpose language understanding systems**. Advances in neural information processing systems. 2019;32.

[6] Hendrycks D, Burns C, Basart S, Zou A, Mazeika M, Song D, Steinhardt J. **Measuring massive multitask language understanding**. arXiv preprint arXiv:2009.03300. 2020 Sep 7.

[7] Suzgun M, Scales N, Schärli N, Gehrmann S, Tay Y, Chung HW, Chowdhery A, Le QV, Chi EH, Zhou D, Wei J. **Challenging big-bench tasks and whether chain-of-thought can solve them**. arXiv preprint arXiv:2210.09261. 2022 Oct 17.
+ Biology, Reasoning, Math, and more

[8] Liang P, Bommasani R, Lee T, Tsipras D, Soylu D, Yasunaga M, Zhang Y, Narayanan D, Wu Y, Kumar A, Newman B. **Holistic evaluation of language models**. arXiv preprint arXiv:2211.09110. 2022 Nov 16.

Resources

[9] Wu S, Irsoy O, Lu S, Dabravolski V, Dredze M, Gehrmann S, Kambadur P, Rosenberg D, Mann G. **Bloomberggpt: A large language model for finance**. arXiv preprint arXiv:2303.17564. 2023 Mar 30.

*[2] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. **2024**;25(70):1-53.*

*[11] Lialin V, Deshpande V, Rumshisky A. **Scaling down to scale up: A guide to parameter-efficient fine-tuning**. arXiv preprint arXiv:2303.15647. 2023 Mar 28.*

*[12] Hu EJ, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W. **LoRA: Low-Rank Adaptation of Large Language Models**. In International Conference on Learning Representations 2021 Oct 6.*

[14] Chung HW, Hou L, Longpre S, Zoph B, Tay Y, Fedus W, Li Y, Wang X, Dehghani M, Brahma S, Webson A. **Scaling instruction-finetuned language models**. Journal of Machine Learning Research. **2024**;25(70):1-53.

[15] Stiennon N, Ouyang L, Wu J, Ziegler D, Lowe R, Voss C, Radford A, Amodei D, Christiano PF. **Learning to summarize with human feedback**. Advances in Neural Information Processing Systems. 2020;33:3008-21.

[16] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. **Proximal policy optimization algorithms**. arXiv preprint arXiv:1707.06347. 2017 Jul 20.

*[17] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih WT, Rocktäschel T, Riedel S. **Retrieval-augmented generation for knowledge-intensive nlp tasks**. Advances in Neural Information Processing Systems. 2020;33:9459-74.*

[18] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D. **Chain-of-thought prompting elicits reasoning in large language models.** Advances in neural information processing systems. 2022 Dec 6;35:24824-37.

[19] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, Callan J, Neubig G. Pal: **Program-aided language models.** In International Conference on Machine Learning 2023 Jul 3 (pp. 10764-10799). PMLR.

[20] Topsakal O, Akinci TC. **Creating large language model applications utilizing langchain: A primer on developing llm apps fast.** In International Conference on Applied Engineering and Natural Sciences 2023 Jul (Vol. 1, No. 1, pp. 1050-1056). https://www.langchain.com/

[21] Yao S, Zhao J, Yu D, Shafran I, Narasimhan KR, Cao Y. **ReAct: Synergizing Reasoning and Acting in Language Models**. InNeurIPS 2022 Foundation Models for Decision Making Workshop 2022 Nov 18.

[22] Pandya K, Holia M. **Automating Customer Service using LangChain: Building custom open-source GPT Chatbot for organizations**. arXiv e-prints. **2023** Oct:arXiv-2310.

Resources

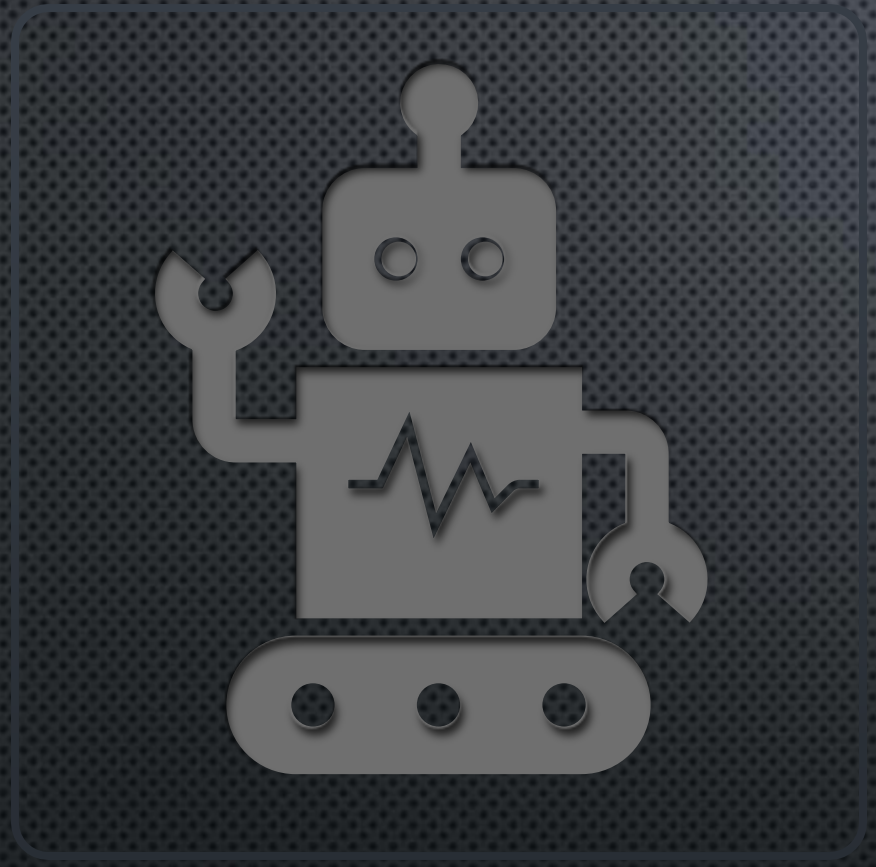
1. **Transformer Architecture**
   - Attention is All You Need
     This paper introduced the Transformer architecture, with the core "self-attention" mechanism. This article was the foundation for LLMs.
   - BLOOM: BigScience 176B Model
     BLOOM is a open-source LLM with 176B parameters trained in an open and transparent way. In this paper, the authors present a detailed discussion of the dataset and process used to train the model + high-level overview of the model here
   - Scaling Laws for Neural Language Models
     Empirical study by researchers at OpenAI exploring the scaling laws for large language models.

2. **Model architectures and pre-training objectives**
   - What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?
     The paper examines modeling choices in large pre-trained language models and identifies the optimal approach for zero-shot generalization.
   - HuggingFace Tasks and Model Hub - Collection of resources to tackle varying machine learning tasks using the HuggingFace library.
   - LLaMA: Open and Efficient Foundation Language Models
     Article from Meta AI proposing Efficient LLMs (their model with 13B parameters outperform GPT3 with 175B parameters on most benchmarks)

3. **Scaling laws and compute-optimal models**
   - Language Models are Few-Shot Learners
     This paper investigates the potential of few-shot learning in Large Language Models.
   - Training Compute-Optimal Large Language Models
     Study from DeepMind to evaluate the optimal model size and number of tokens for training LLMs. Also known as "Chinchilla Paper".
   - BloombergGPT: A Large Language Model for Finance
     LLM trained specifically for the finance domain, a good example that tried to follow chinchilla laws.

Thank you for your time and attention

Any Questions?



INTRODUCTION TO ADVANCED LARGE LANGUAGE MODELS

A Comprehensive Tutorial on Techniques, Architectures, and Practical Applications

Giorgio Roffo, PhD

Explore and Connect With My Professional Work:

LinkedIn: Giorgio Roffo - Linkedin
ResearchGate: Work Done
Google Scholar: My Publications
GitHub: giorgioroffo