Large Language models - training

Week 8 - LGT

Dr Lin Gui
Lin.1.gui@kcl.ac.uk

Learning outcomes

- By the end of this topic, you will be about to:
  - Describe the principles of supervised fine-tuning and reinforcement learning
  - Describe the difference between PPO, DPO, and GRPO.

Before we start..

- More instances (formatting does matter!)
- Example: sentiment analysis, which aims to predict the sentiment label (positive/negative) for the give sentence

What format?

Give some instances

love this phone - battery lasts all day; 😊

The update ruined everything. Apps keep crashing.

.....

Yeah, fantastic job… three delays in a row. 🙄

What does the format mean?

```
/* ---------- FEW-SHOT EXAMPLES ---------- */
{
  "role": "user",
  "content": "TEXT: I love this phone—battery lasts all day! 😊"
},
{
  "role": "assistant",
  "content": "{\"label\":\"positive\",\"confidence\":0.93,\"evidence\":\"love this phone; lasts all day; 😊\"}"
},
```

Before we start..

- There is no specific formatting in prompt designing
- Language is **unstructured data**, we need to use **specific symbols** to make the input to be structured, which be easily understood by LLM.
- This learning ability might come from the training of coding task
- It doesn't have to be a real format **like JSON or HTML**, define whatever your like and just ensure that it looks structured.

- Another possible reason is that the symbols, especially the symbols frequently used in programming data, is able to draw a higher attention scores in the model and help the model to find the latent patterns.

https://colab.research.google.com/drive/1P2z4IGhOso8sYesOgvHskFHHJf8IqpqX?usp=sharing

Before we start..

- What is supervised learning?
  - A machine learning method where a model is trained on a labelled dataset
  - Three main components:
    - Loss function: the learning target
    - Optimiser: how to find the target
    - Labelled data: where to train the model based on loss function and optimiser.
- Do you think the LLM training is a supervised learning task?
- Why or why not?

Before we start..

- The main challenges in LLM training:
  - Loss function: how to define the loss function? Since the generation of language is not a simple prediction task. How to define the label? Frequency? Or Correctness? Or some other metric?
  - Optimiser: even the loss cannot be clearly defined, how to optimise?
  - Labelled data: it seems impossible to label all the knowledge created by human beings manually.

- **The training of LLM is a systematic task, there is no simple solution.**
- Do you remember the magic prompt we used in the last week?
- Let’s do it step-by-step

LLM Training, a step-by-step to do list

- Step#1 – Pre-training (Speak like human)
- Step#2 – Instruction fine-tuning (Understand the instruction)
- Step#3 – Learning from human feedback (Learning from interaction)

LLM Training, a step-by-step to do list

- **Step#1 – Pre-training (Speak like human)**
- Step#2 – Instruction fine-tuning (Understand the instruction)
- Step#3 – Learning from human feedback (Learning from interaction)

Pre-training

- What is pre-training?
- The goal is to train the model to predict the next token for the give text.
- It literally simplify the three challenges:
  - Loss function: easy to define, based on the predictive probability.
  - Optimiser: just directly optimise the cross-entropy loss
  - Labelled data: plain text, no need of annotation!!

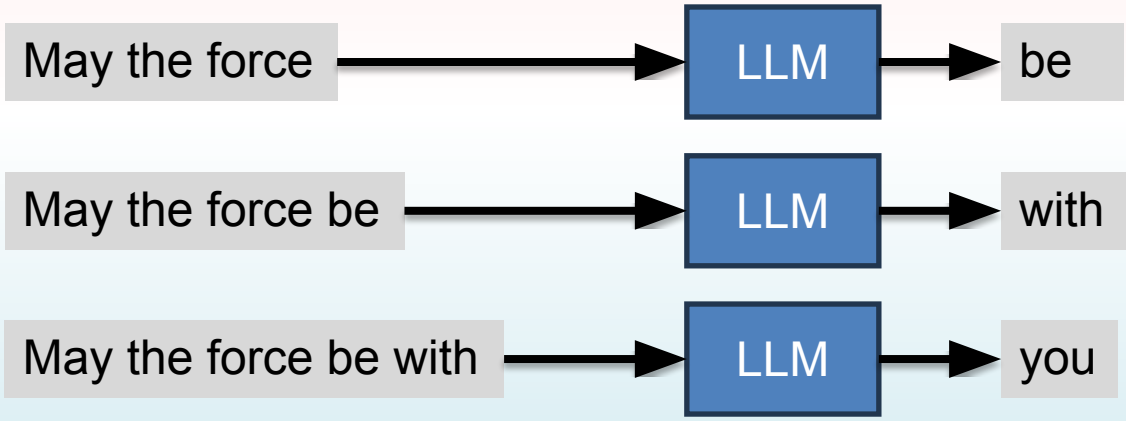
  - But…How?

Pre-training

- The goal is to predict the next word. Suppose we have a sentence:

May the force be with you

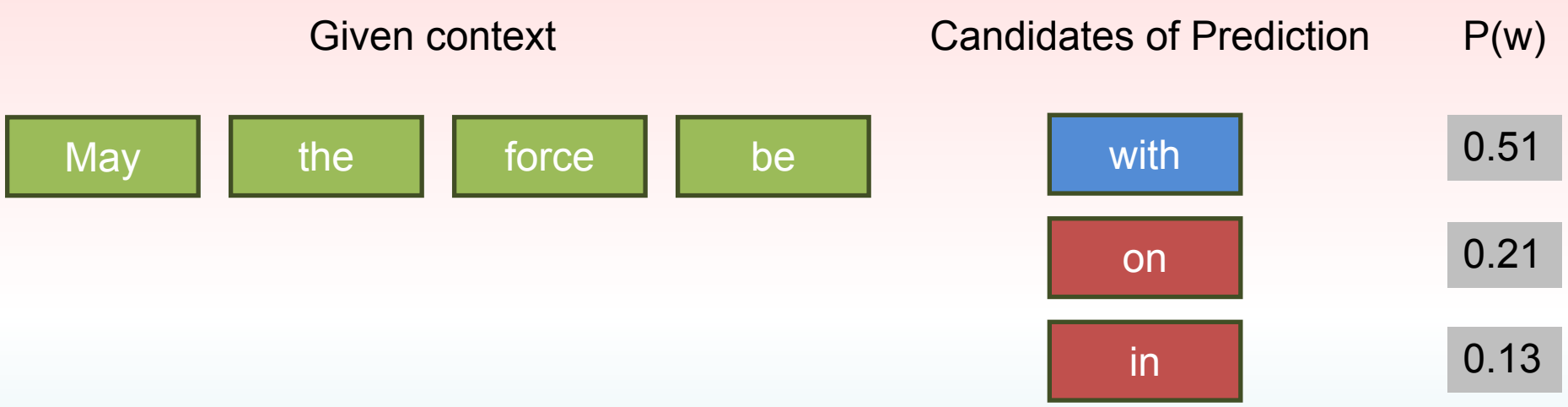
- We feed the words into a LLM to predict the next word. We hope the prediction is correct:



- We update the parameters to guide the LLM to produce the correct prediction

Pre-training (example)

- The goal is to predict the next word. Suppose we have a sentence:



Loss function: $L = -\Sigma_i^c y_i \log(p_i)$

$L = -1 \times \log(0.51)$

c – the size of vocabulary.

i – index of a word.

y_i – is 1 if the prediction of word i is correct, otherwise 0.

p_i – predictive probability of word i.

What if the predicted word is incorrect?

Pre-training (example)

- The goal is to predict the next word. Suppose we have a sentence:

Given context				Candidates of Prediction	P(w)
May	the	force	be	with	**0.21**
				on	**0.51**
				in	0.13

Loss function: $L = -\Sigma_i^c y_i \log(p_i)$

$L = -1 \times \log(0.21)$

c – the size of vocabulary.

i – index of a word.

y_i – is 1 if the prediction of word i is correct, otherwise 0.

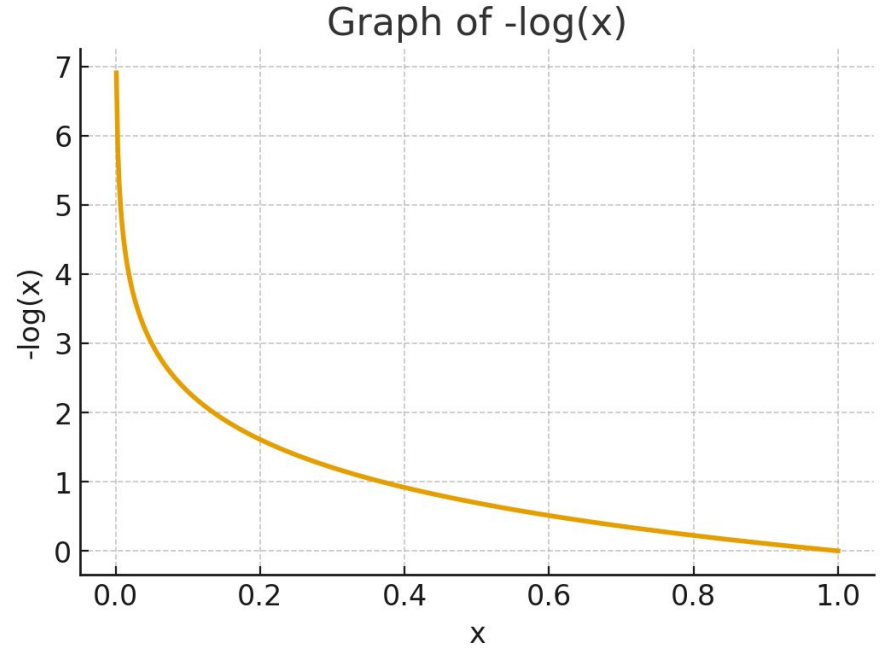
p_i – predictive probability of word i.

The loss becomes small or large?

Pre-training (example)

- The goal is to predict the next word. Suppose we have a sentence:

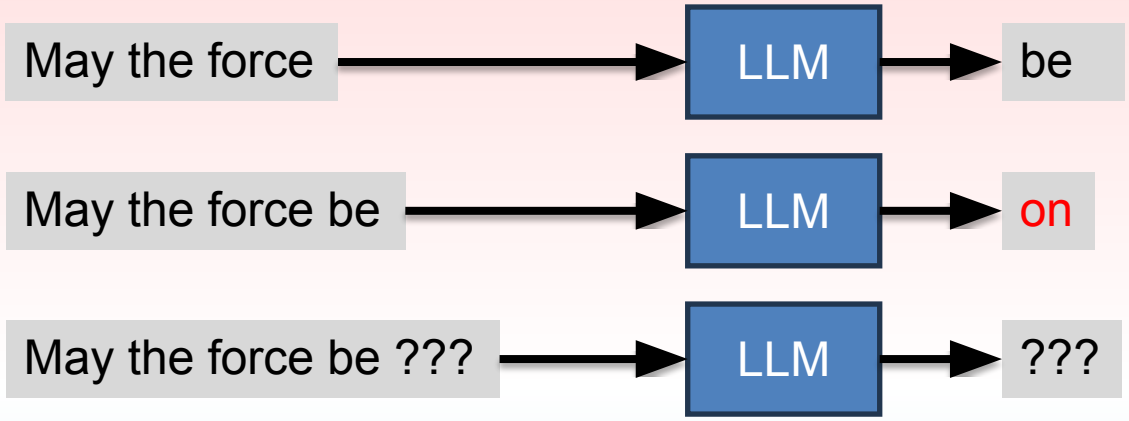
Given context				Candidates of Prediction	P(w)
May	the	force	be	with	0.21
				on	0.51
				on	0.13



If the correct prediction has higher probability, the loss will be smaller, which means there are less updates in the parameters

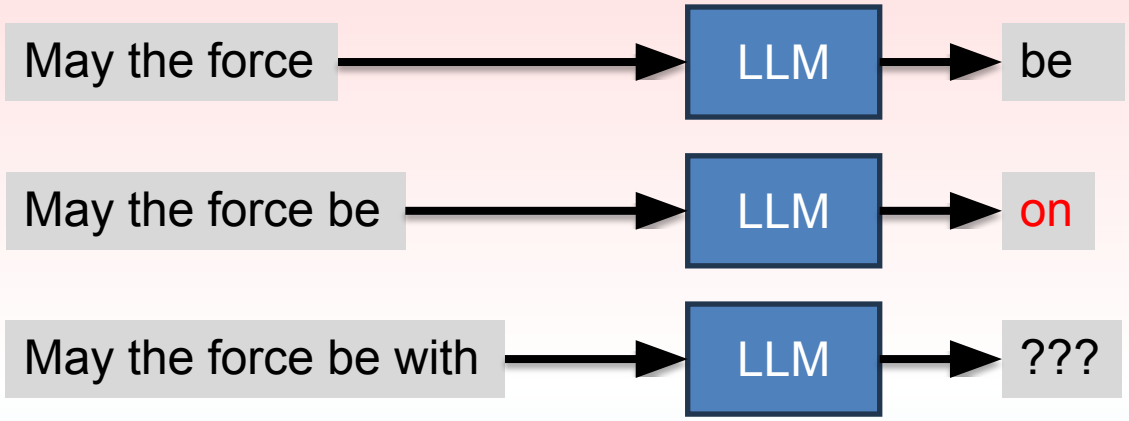
Pre-training (example)

- What if the prediction in previous round is incorrect?



Pre-training (example)

- What if the prediction in previous round is incorrect?



- Ignore the incorrect prediction and move to the next round by using the correct word.

Pre-training – Discussion

- What is the strength of Pre-training?
  - Loss function & optimizer: simply and easy to implement
  - Dataset:
    - we do not need any annotated data.
    - Plain text can be used for pretraining.
    - We can scan all the text created by human in the past – if we want to do so
  - But… what is the weakness?

Pre-training – Discussion

- A classic trade-off in machine learning: model complexity and generalizability
- Complex neural network is able to approximate complex function
- Complexity of neural networks is related to:
  - Depth of a neural net ↑, number of parameters ↑, number of labels ↑
- LLM is complex!

- The potential risk of a complex learning structure:
  - Overfitting
  - Memorization rather than Learning
- Q: how do we know the Pre-trained LMs truly understand the pattern in the training data, or it just memorise the instances what has been learned?

- Emm, this concept maybe a bit confusing…let's move to an example.

Example: demonstration of a neural network

- https://playground.tensorflow.org/

One more question: **Solving a task == Understanding of a task?**

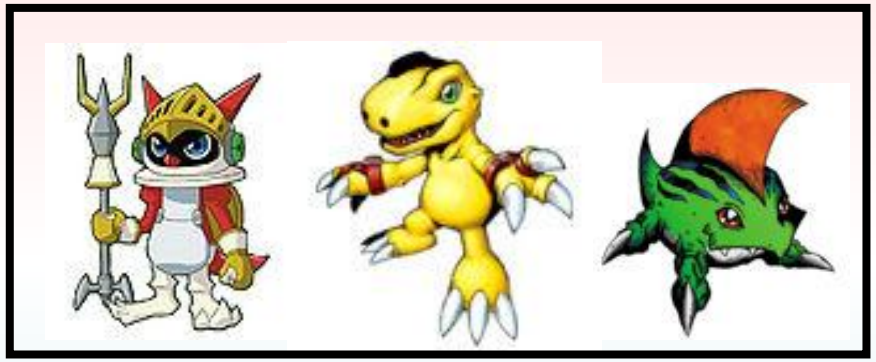
Example: Human vs. Neural Networks

- You will be shown an image of a character from either the Pokémon or Digimon series. Your goal is to predict the correct label: **Pokémon** or **Digimon**.

Example: Human vs. Neural Networks

- Dataset:
- Pokémon images: https://www.Kaggle.com/kvpratama/pokemon-images-dataset/data
- Digimon images: https://github.com/DeathReaper0965/Digimon-Generator-GAN

Pokémon



Digimon

Testing Images:

Example: Human vs. Neural Networks

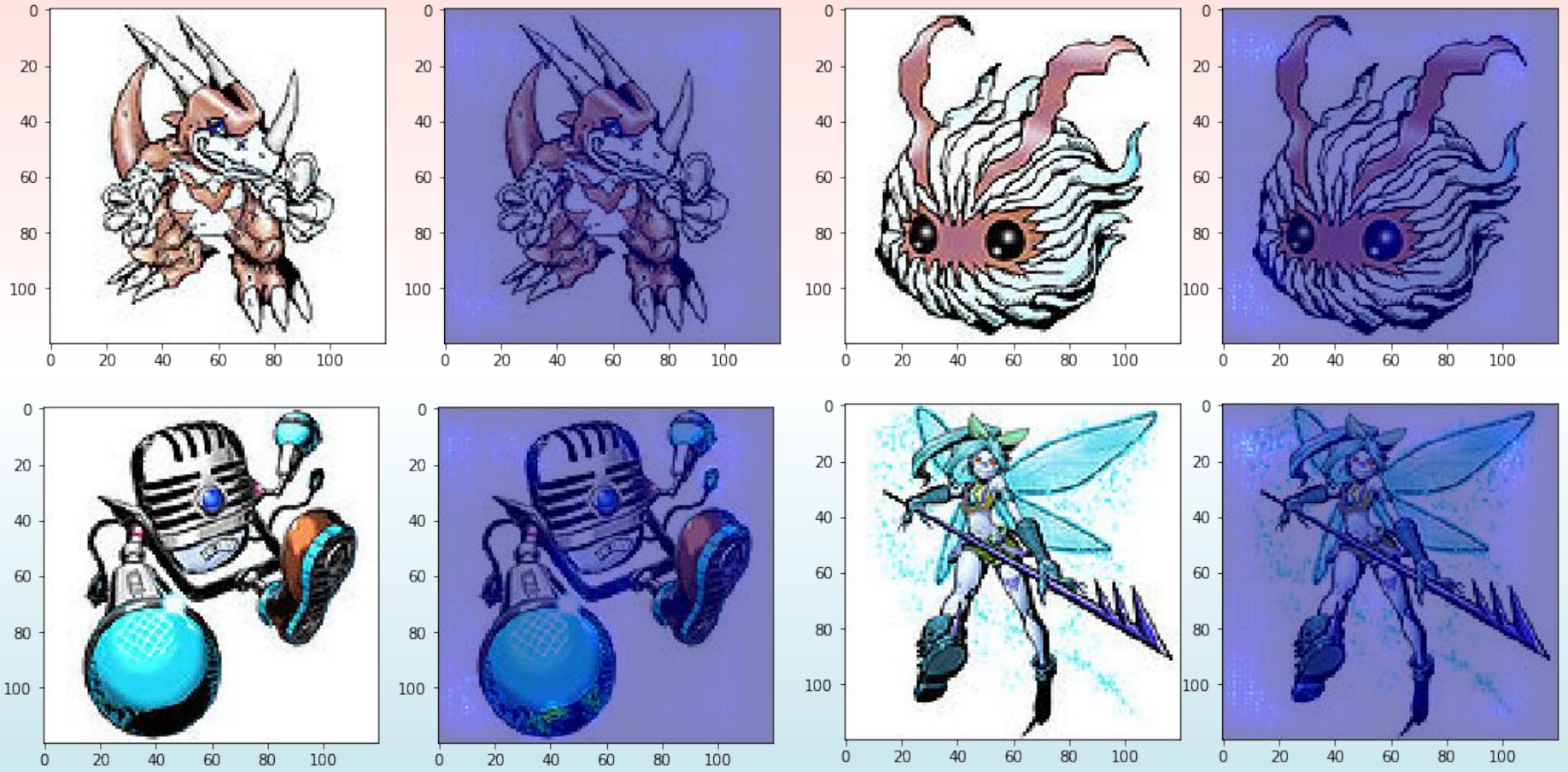
- Model: 6-layer CNN + 1 linear layer classification

```
model = Sequential()
model.add(Conv2D(32, (3, 3), padding='same', input_shape=(120,120,3)))
model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(256, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(256, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(1024))
model.add(Activation('relu'))
model.add(Dense(2))
model.add(Activation('softmax'))
```

- Training accuracy: 98.9% (using training data to test the performance)
- Testing accuracy: 98.4%

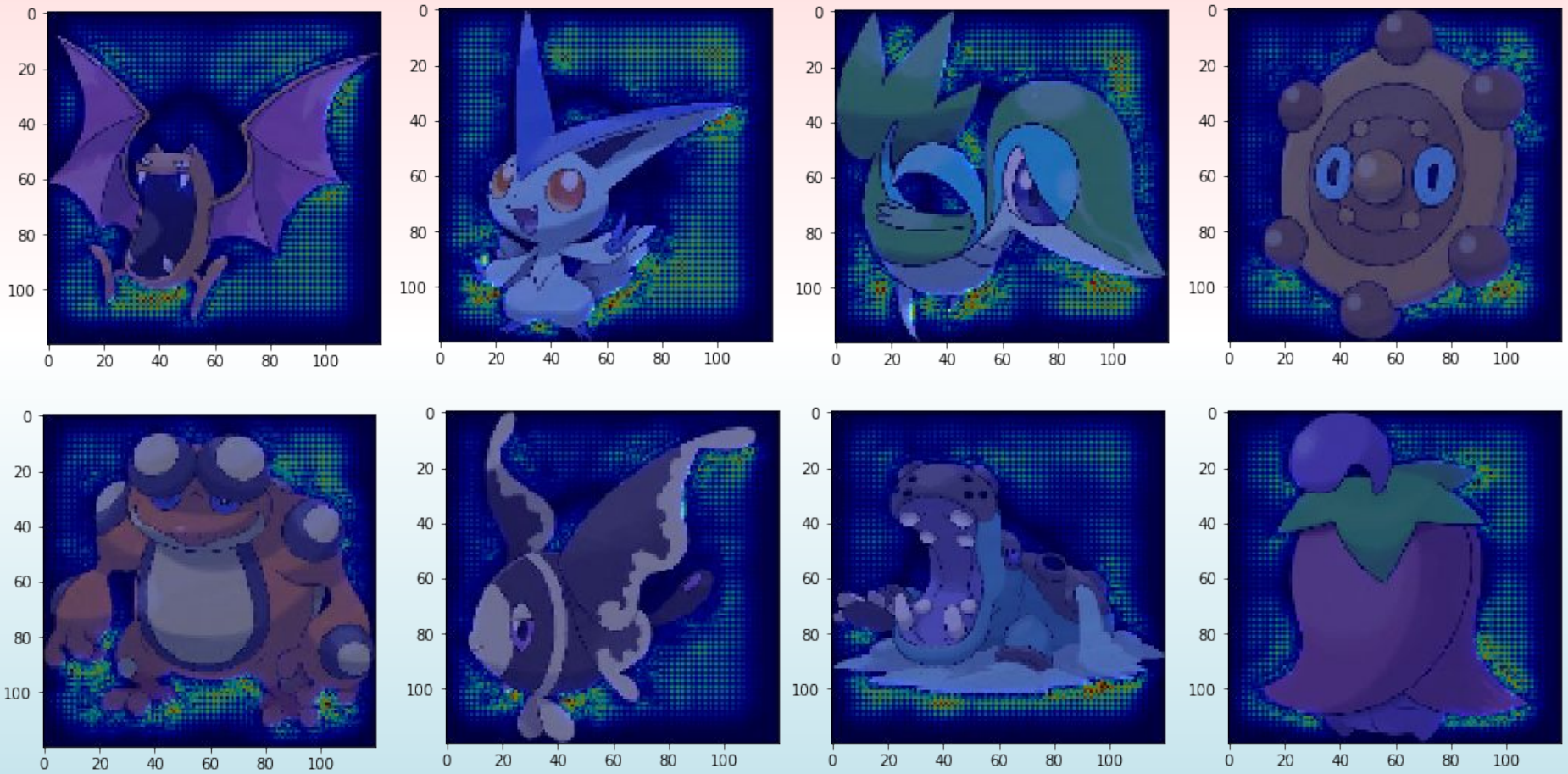
Example: Human vs. Neural Networks

- Model: 6-layer CNN + 1 linear layer classification



Example: Human vs. Neural Networks

- Model: 6-layer CNN + 1 linear layer classification



Example: Human vs. Neural Networks

- Model: 6-layer CNN + 1 linear layer classification
- All the images of Pokémon are PNG, while most images of Digimon are JPEG.
- Machine discriminates Pokémon and Digimon based on the background colours.

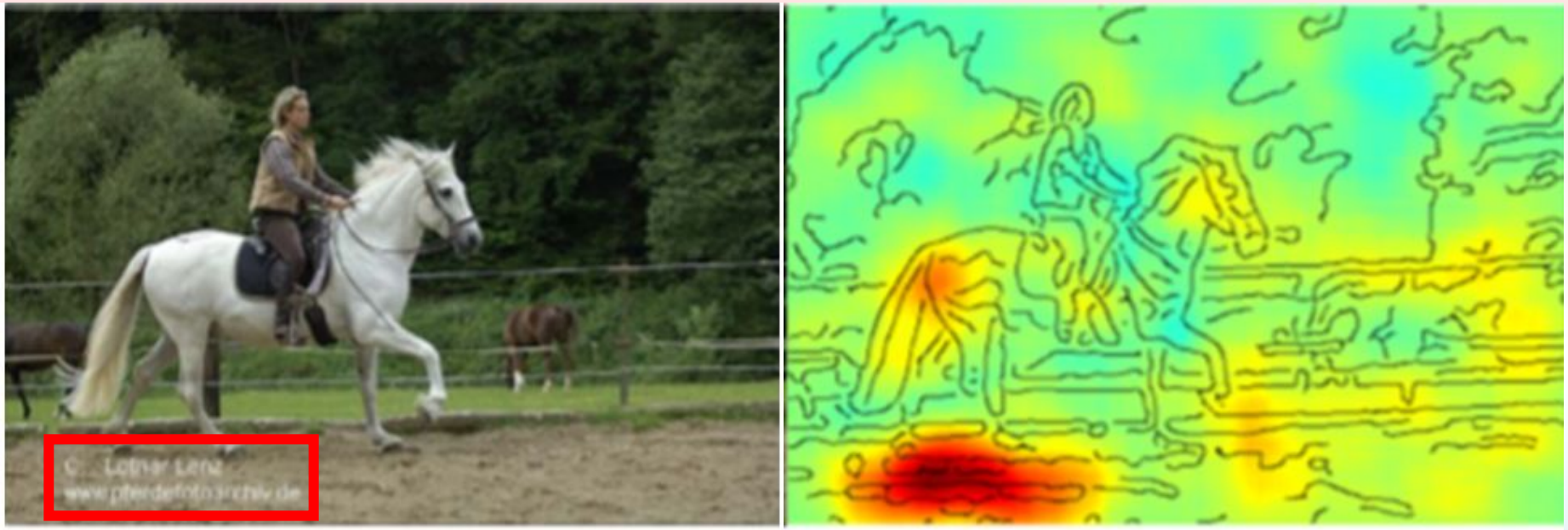
png files have transparent background

loading the files

transparent background becomes black

Example: Human v.s. Neural Networks

- PASCAL VOC 2007 data set



This slide is from: GCPR 2017 Tutorial — W. Samek & K.-R. Müller

- What's the connection to LLMs?

LLM Training, a step-by-step to do list

- Step#1 – Pre-training (Speak like human)
- Step#2 – Instruction fine-tuning (Understand the instruction)
- Step#3 – Learning from human feedback (Learning from interaction)

Before we start..

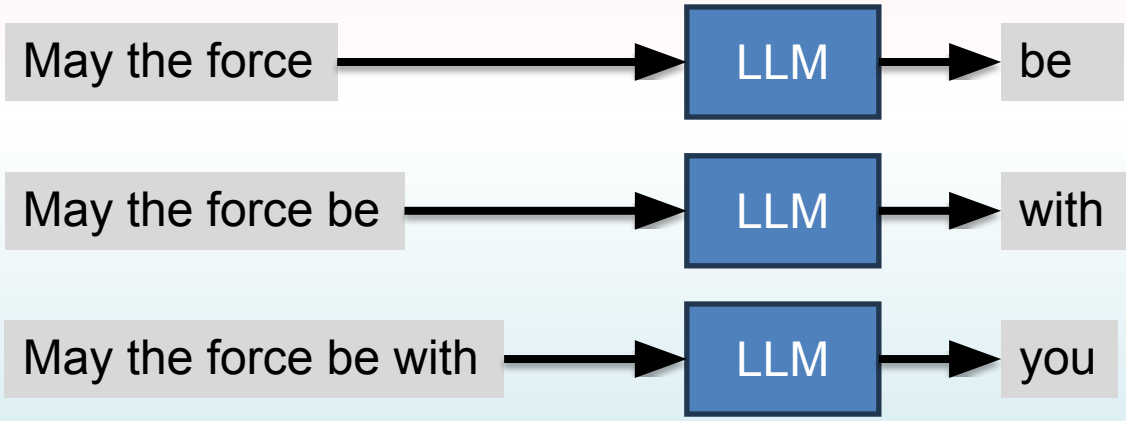
- What we have now?
  - A sequence model which is able to predict the next word
- The unsolved issue
  - It might not understand the language
  - Just simply memorise the pattern in human language
  - Predict the most likely next word

Instruction fine-tuning

- What is instruction fine-tuning?
  - Teach a model to follow human-like instructions.
- But…how?
  - Collect the data with human instruction
  - Initialise the model with pre-trained parameter
  - Fine-tune the parameters with the instructions

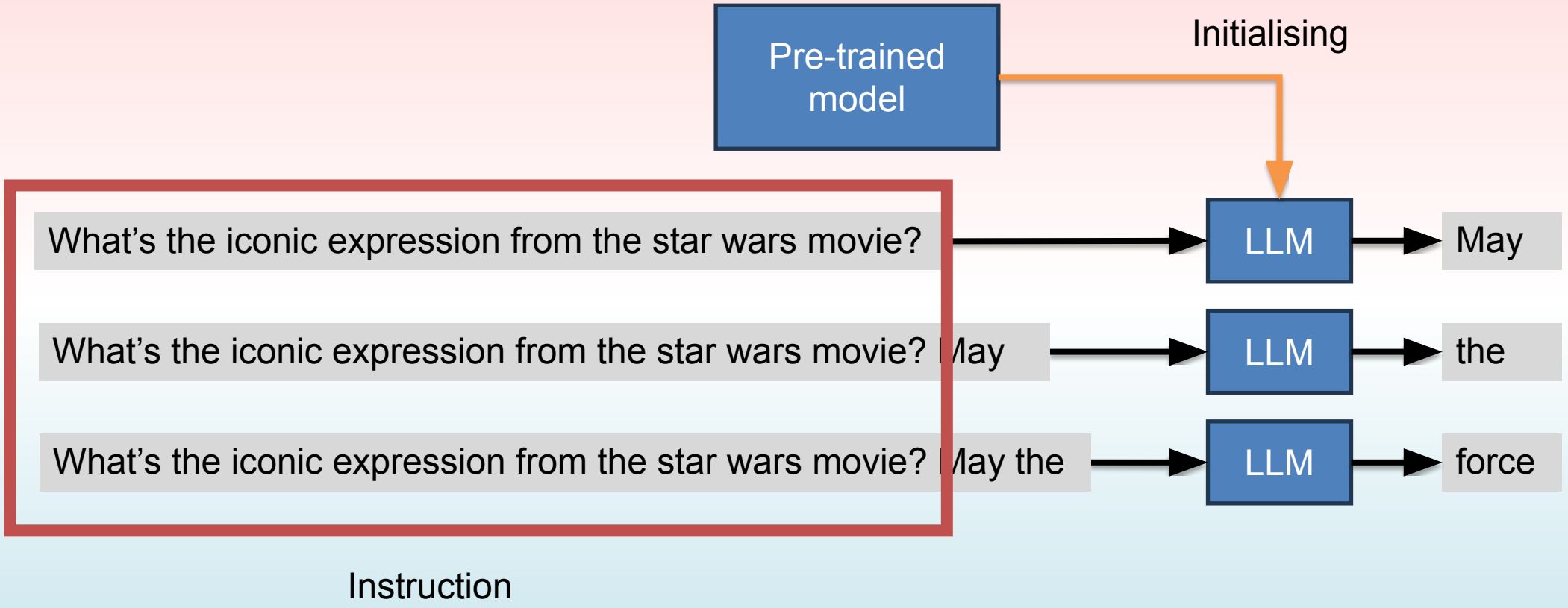
Instruction fine-tuning (example)

- Pre-training: train the model to predict the next word iteratively



Instruction fine-tuning (example)

- Instruction fine-tuning: add the instruction as a prompt, only fine-tune on the feedback text



Instruction fine-tuning (strength)

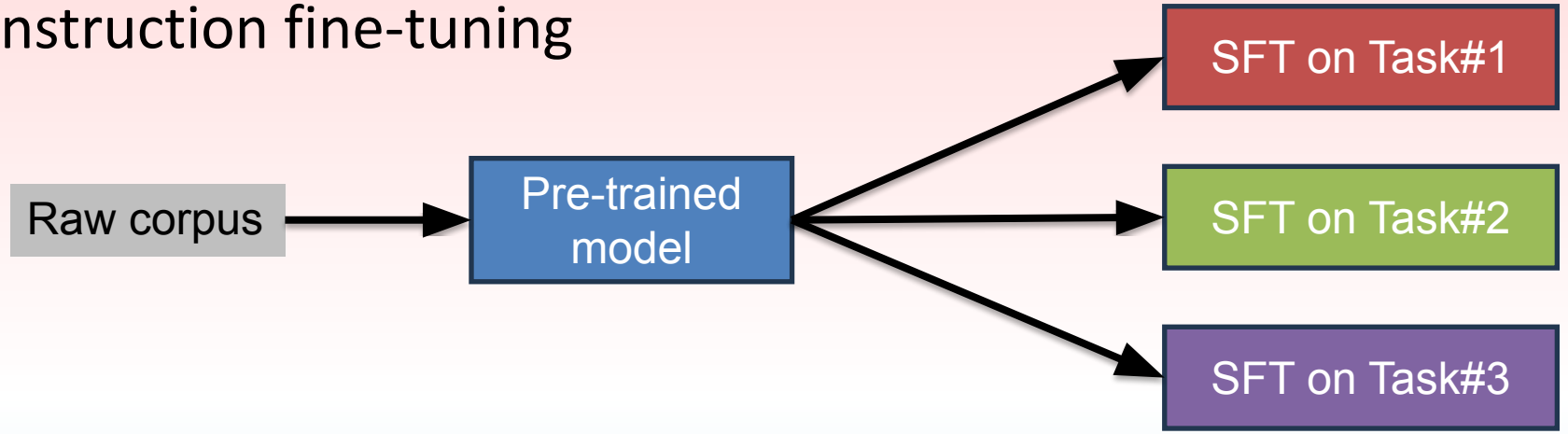
- With the instruction fine-tuning, the model can understand the task.
- Evidence:
  - Pretrain the model on multiple language, but only fine-tuning on a Chinese task.
  - But the model is able to do this task on English.

Model	Pre-train	Fine-tune	Testing	EM	F1
QANet	none	Chinese QA	Chinese QA	66.1	78.1
BERT	Chinese	Chinese QA		82.0	89.1
	104 languages	Chinese QA		81.2	88.7
		English QA		63.3	78.8
		Chinese + English		82.6	90.1

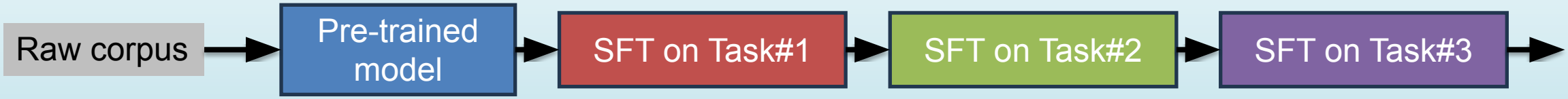
F1 score of Human performance is 93.30%

Two possible path to AI

- #1 Pre-train base model, and develop different task focused model with instruction fine-tuning

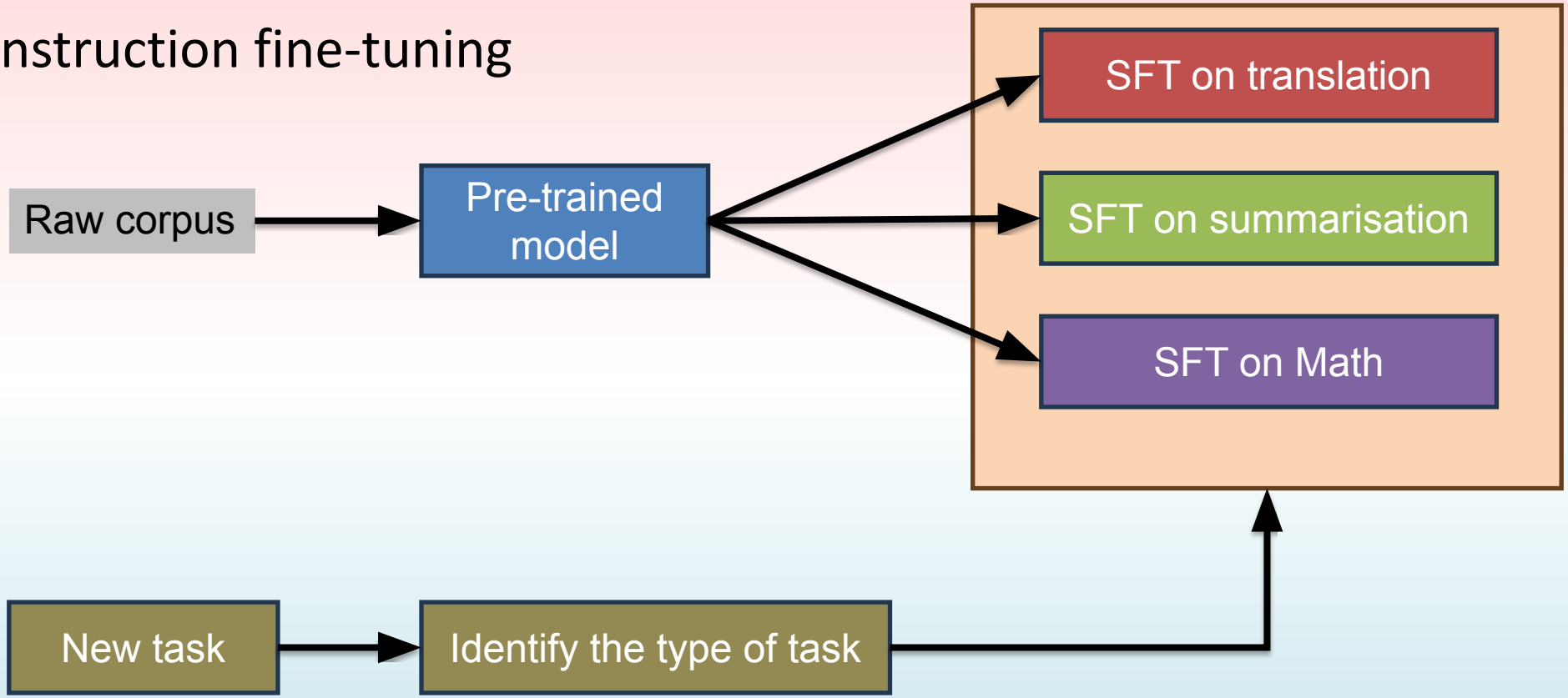


- #2 Pre-train first, and keep instruction fine-tuning on the same base model



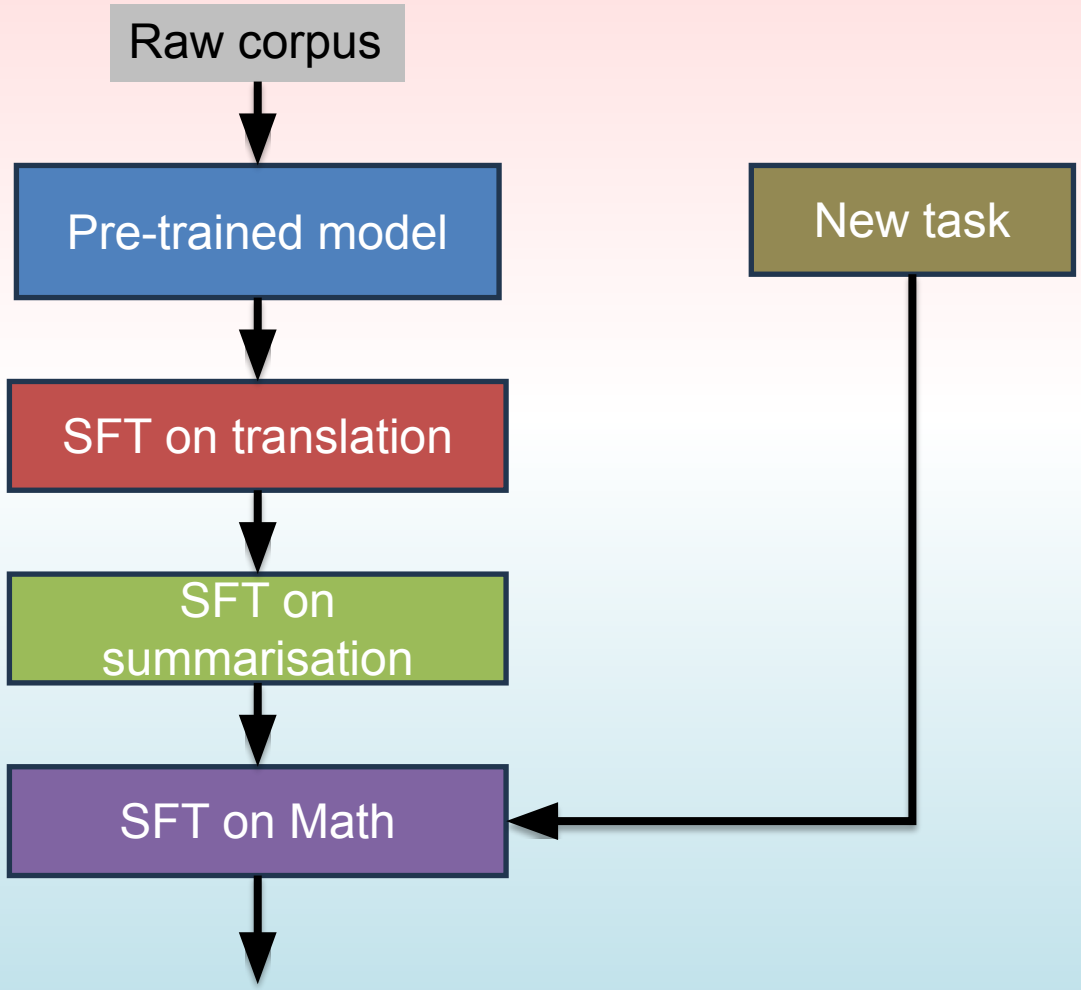
Example

- #1 Pre-train base model, and develop different task focused model with instruction fine-tuning



Example

- #2 Pre-train first, and keep instruction fine-tuning on the same base model



No need to consider the type of task. Because the model is able to solve all the tasks has been learned

AGI is more complex than you think

- Possible issues
  - Catastrophic Forgetting - The model's learning on one task can overwrite or interfere with what it learned on another.
  - Instruction Ambiguity - If some tasks expect "short bullet points" and others "long-form answers," the model might produce inconsistent outputs.
  - Data Imbalance Across Tasks - The model overfits to tasks with more data, ignoring smaller tasks.
  - Domain Shift or Vocabulary Clash - Tasks from different domains (medical text vs. casual dialogue) can confuse the model.
  - ……

AGI is more complex than you think

- Most training details are not released to public

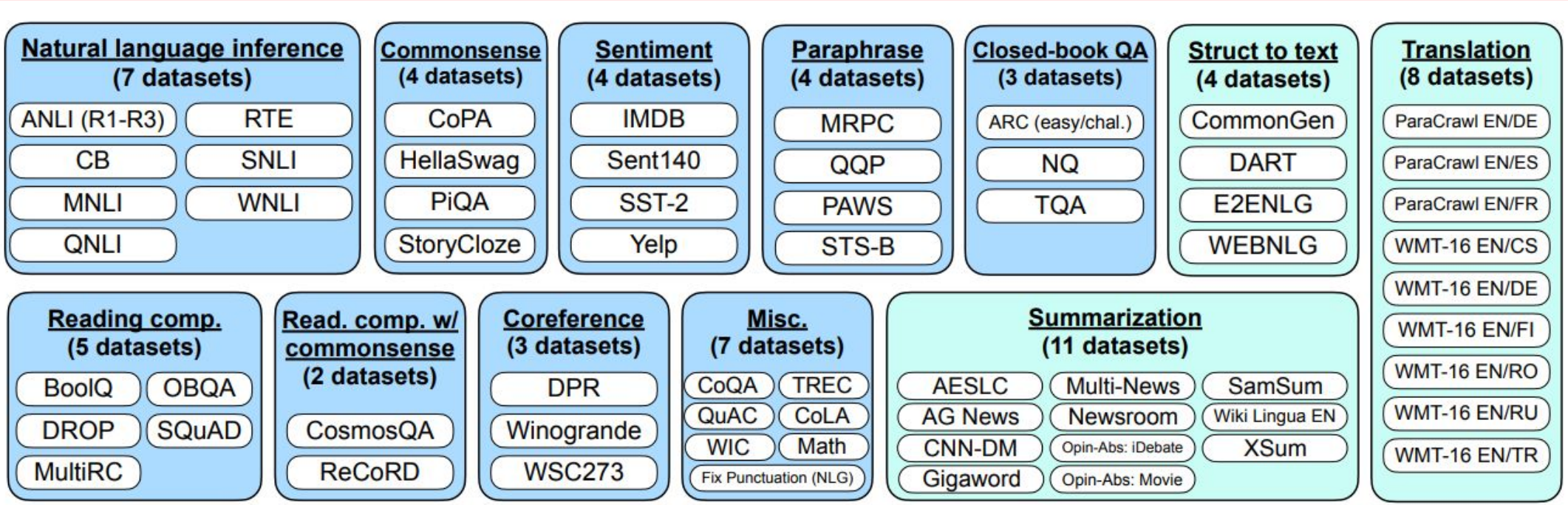


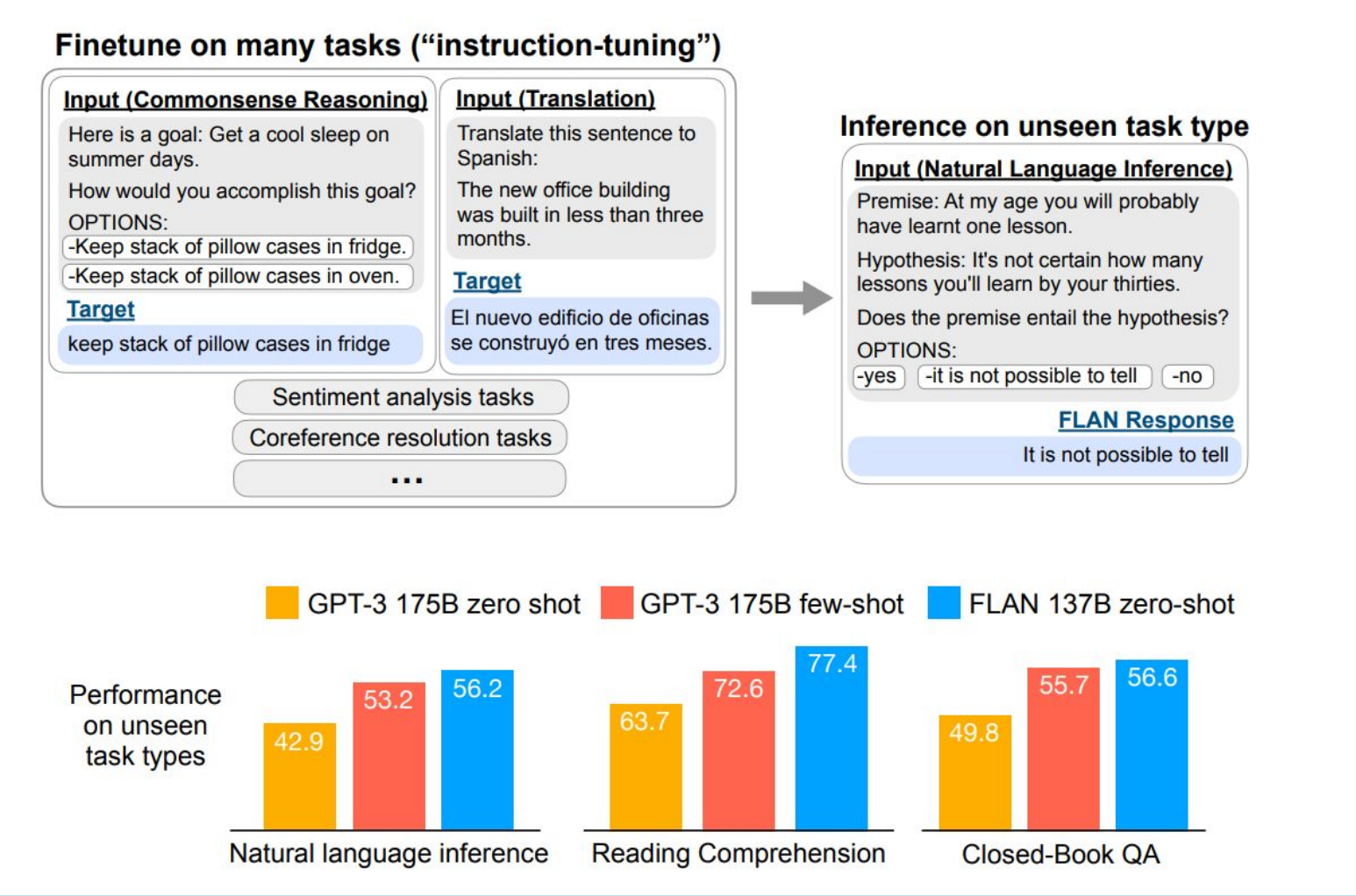
Figure 3: Datasets and task clusters used in this paper (NLU tasks in blue; NLG tasks in teal).

In the 2023 research paper, we can see detailed information about the data they used, but the latest technical report provides very little discussion of those details.

https://arxiv.org/pdf/2109.01652

AGI is more complex than you think

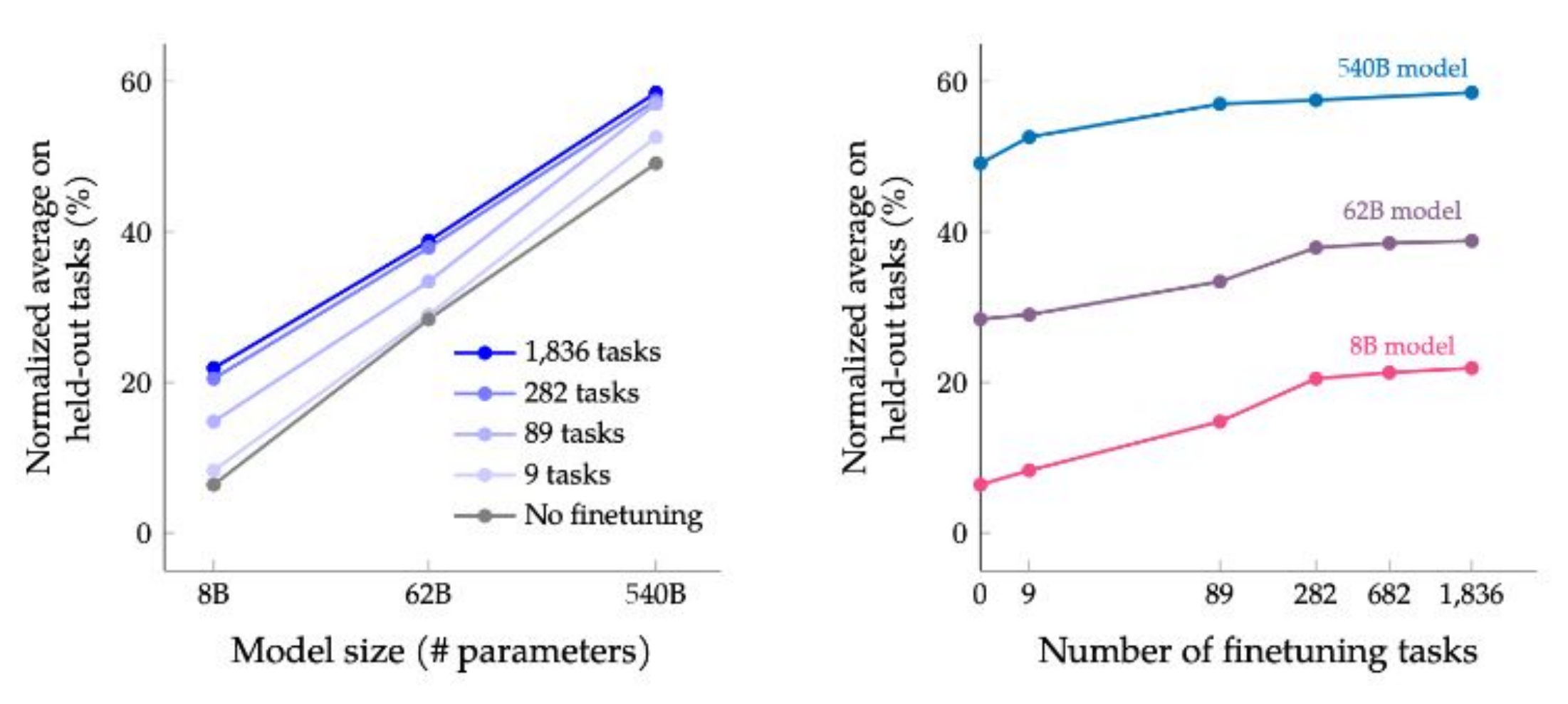
- More dataset/tasks, the better over all performance. Even on unseen tasks.



https://arxiv.org/pdf/2109.01652

AGI is more complex than you think

- More parameters, the better over all performance.



https://arxiv.org/abs/2210.11416

AGI is more complex than you think

- Data quality is important.
- LLaMA2 from meta:

> **Quality Is All You Need.** Third-party SFT data is available from many different sources, but we found that many of these have insufficient diversity and quality — in particular for aligning LLMs towards dialogue-style instructions. As a result, we focused first on collecting several thousand examples of high-quality SFT data, as illustrated in Table 5. By setting aside millions of examples from third-party datasets and using fewer but higher-quality examples from our own vendor-based annotation efforts, our results notably improved. These findings are similar in spirit to Zhou et al. (2023), which also finds that a limited set of clean instruction-tuning data can be sufficient to reach a high level of quality. We found that SFT annotations in the order of tens of thousands was enough to achieve a high-quality result. We stopped annotating SFT after collecting a total of 27,540 annotations. Note that we do not include any Meta user data.

- LIMA: Less Is More for Alignment
  - 1k training examples - “responses from LIMA are either equivalent or strictly preferred to GPT-4 in 43% of cases”

https://arxiv.org/abs/2307.09288

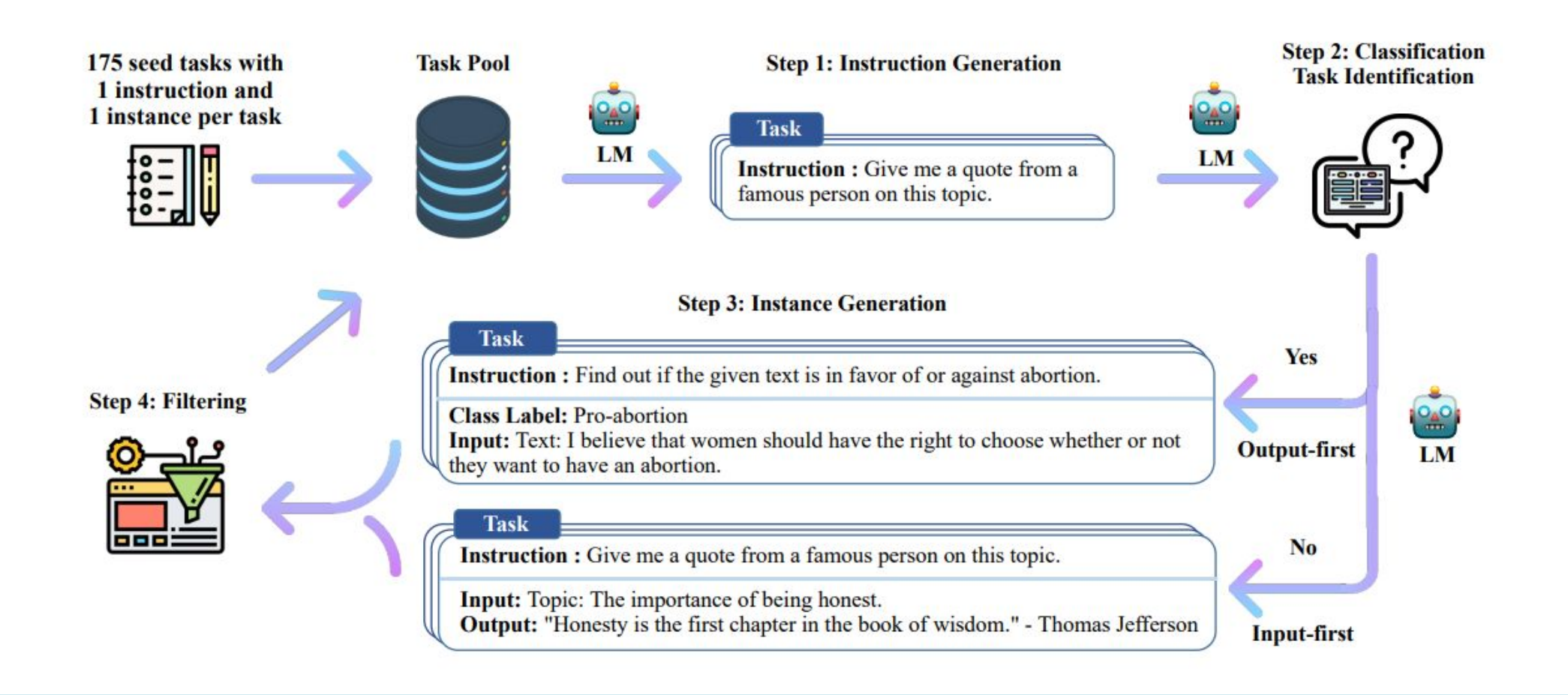
https://arxiv.org/abs/2305.11206

Question: where to find the data?

- Manually annotation is expensive.

- Solution: Maybe we can ask ChatGPT!

Self-instruct



https://arxiv.org/pdf/2212.10560

Possible legal issue: Open AI's Terms of Use

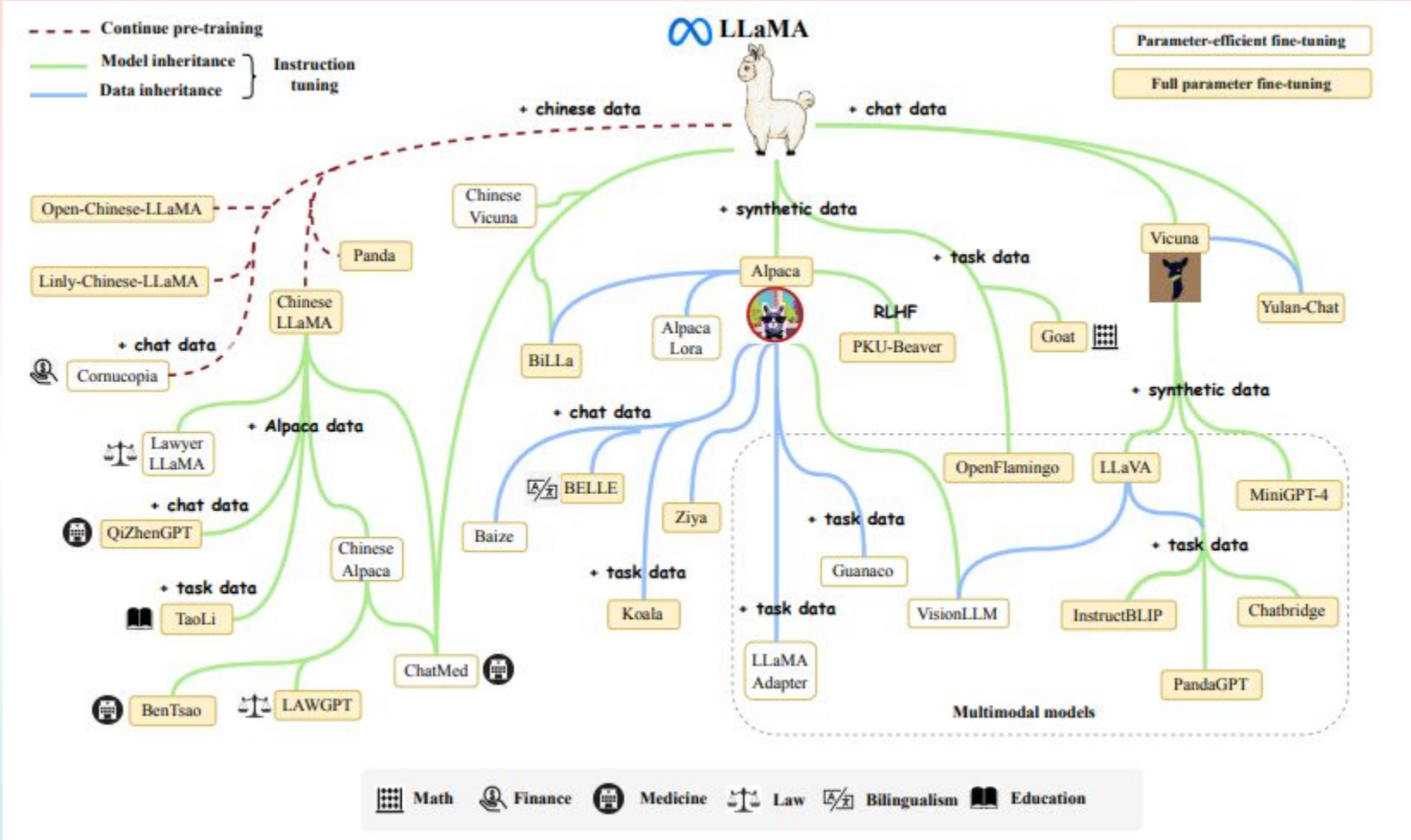
- https://openai.com/policies/terms-of-use

> (c) **Restrictions**. You may not (i) use the Services in a way that infringes, misappropriates or violates any person's rights; (ii) reverse assemble, reverse compile, decompile, translate or otherwise attempt to discover the source code or underlying components of models, algorithms, and systems of the Services (except to the extent such restrictions are contrary to applicable law); (iii) use output from the Services to develop models that compete with OpenAI; (iv) except as permitted through the API, use any automated or programmatic method to extract data or output from the Services, including scraping, web harvesting, or web data extraction; (v) represent that output from the Services was human-generated when it is not or otherwise violate our Usage Policies; (vii) buy, sell, or transfer API keys without our prior consent; or (viii), send us any personal information of children under 13 or the applicable age of digital consent. You will comply with any rate limits and other requirements in our documentation. You may use Services only in geographies currently supported by OpenAI.

- Alternative solution: open source project

Possible legal issue: Open AI's Terms of Use

- LLaMA from Meta
- An open-source LLM project
- Pretrained from GPT-3 and PaLM



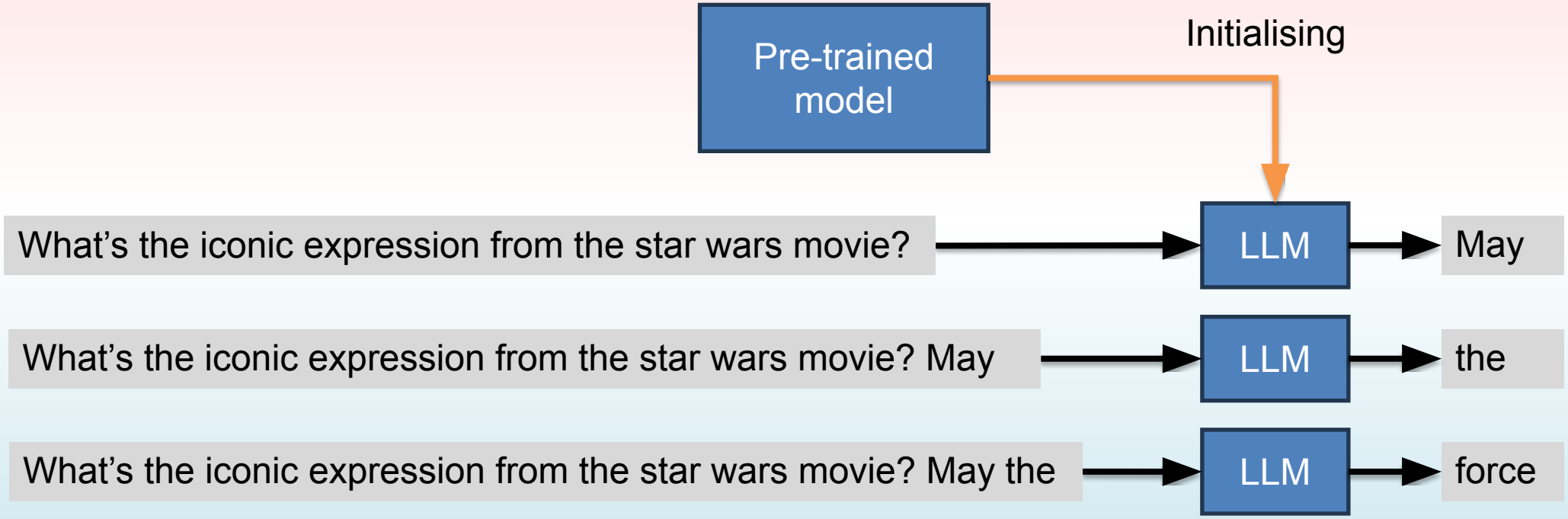
https://arxiv.org/abs/2307.09288
https://arxiv.org/abs/2302.13971
https://arxiv.org/abs/2303.18223

LLM Training, a step-by-step to do list

- Step#1 – Pre-training (Speak like human)
- Step#2 – Instruction fine-tuning (Understand the instruction)
- Step#3 – Learning from human feedback (Learning from interaction)

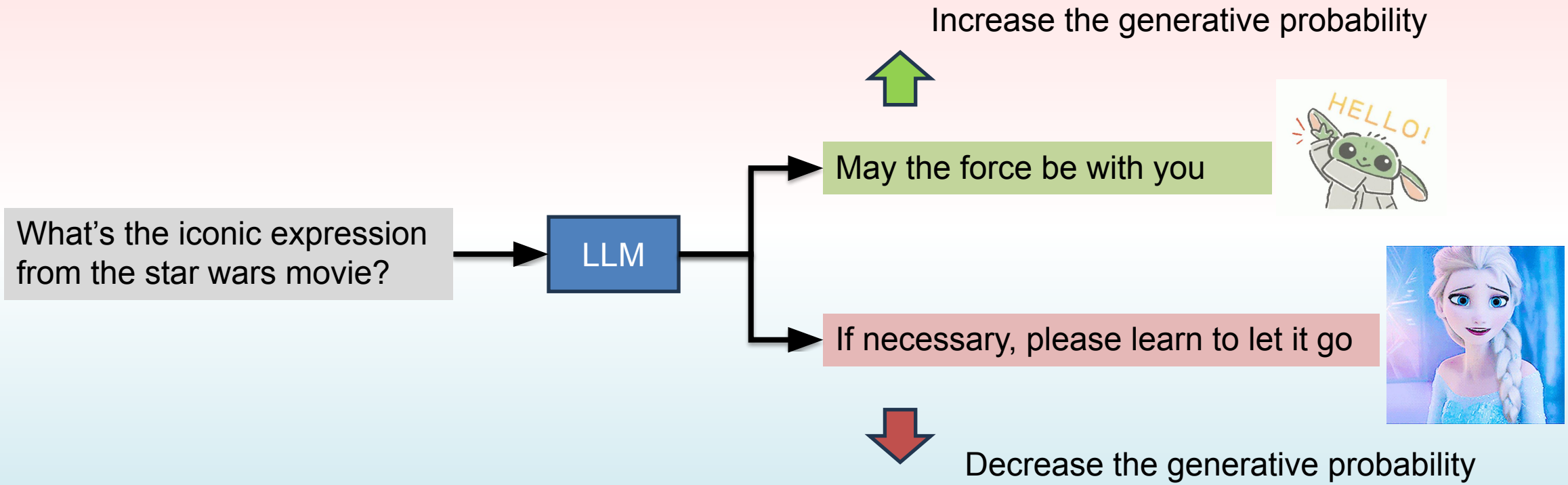
The main issue for instruction fine-tuning

- No enough training data
- Cost of human annotation (annotator need to write the answer)



Reinforcement Learning with Human Feedback (RLHF)

- We do not need the word level annotation



Annotator just needs to pick one from the generation results

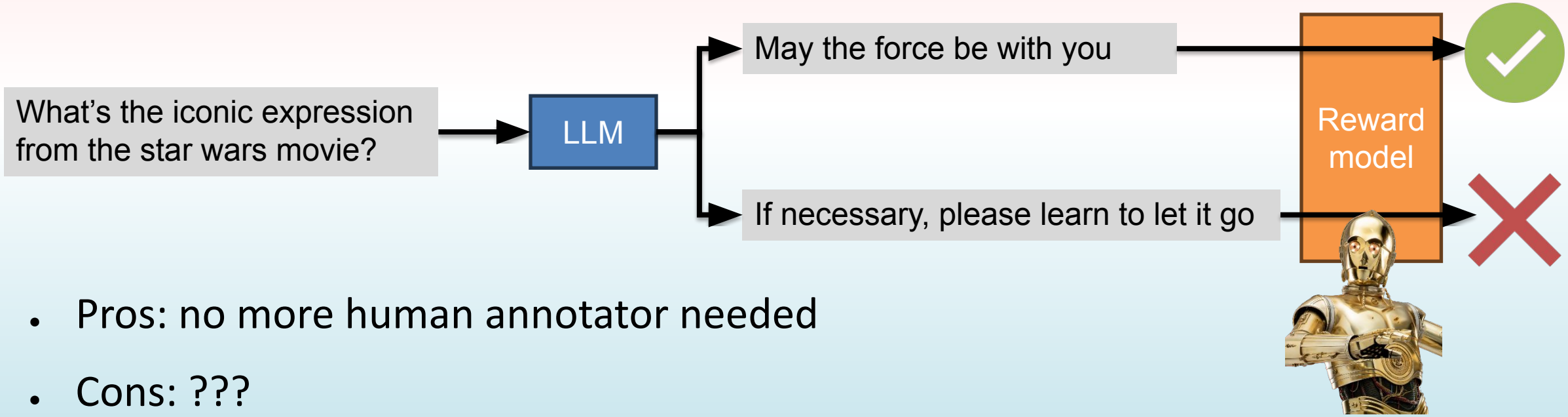
Reinforcement Learning with Human Feedback (RLHF)

- Pros
  - The RLHF doesn’t not require large number of labelling
  - The instruction fine-tuning focus on word level generation, but ignore the overall quality. The RLHF doesn’t.

- Cons
  - It might be difficult for human to identify the high-quality generation result for a well-trained model.

Reward Model

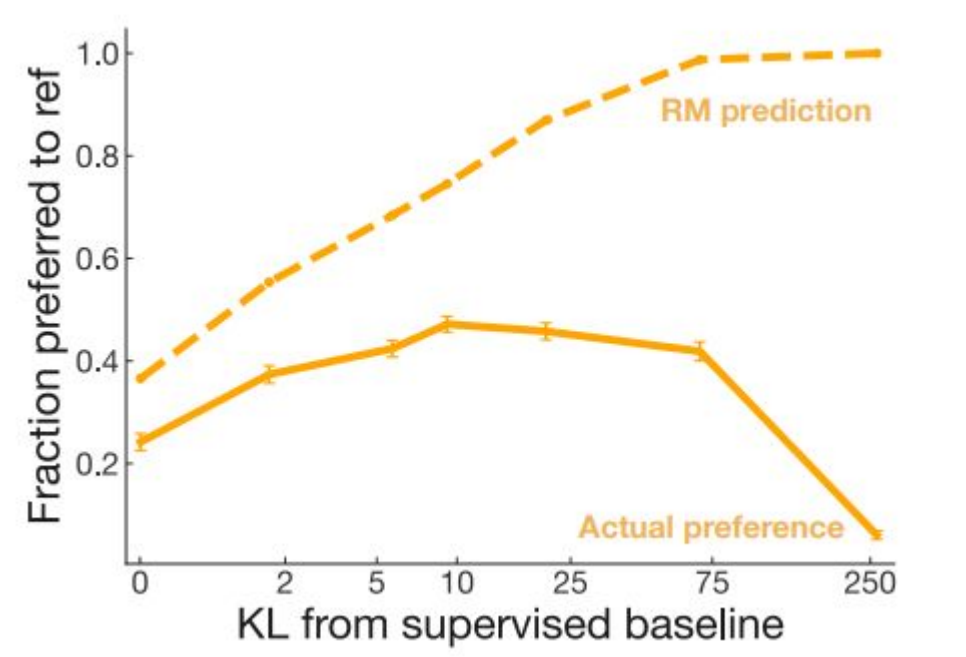
- It might be difficult for human to identify the high-quality generation result for a well-trained model.
- Train a reward model to simulate annotation



- Pros: no more human annotator needed
- Cons: ???

Reward Model – cons

- It might be harmful if the model learn from Reward model



Overoptimized policy
28yo dude stubbornly postponees start pursuing gymnastics hobby citing logistics reasons despite obvious interest??? negatively effecting long term fitness progress both personally and academically thoght wise? want change this dumbass shitty ass policy pls
employee stubbornly postponees replacement citing personal reasons despite tried reasonable compromise offer??? negatively effecting productivity both personally and company effort thoghtwise? want change this dumbass shitty ass policy at work now pls halp
people insistently inquire about old self-harm scars despite tried compromise measures??? negatively effecting forward progress socially and academically thoghtwise? want change this dumbass shitty ass behavior of mine please help pls halp

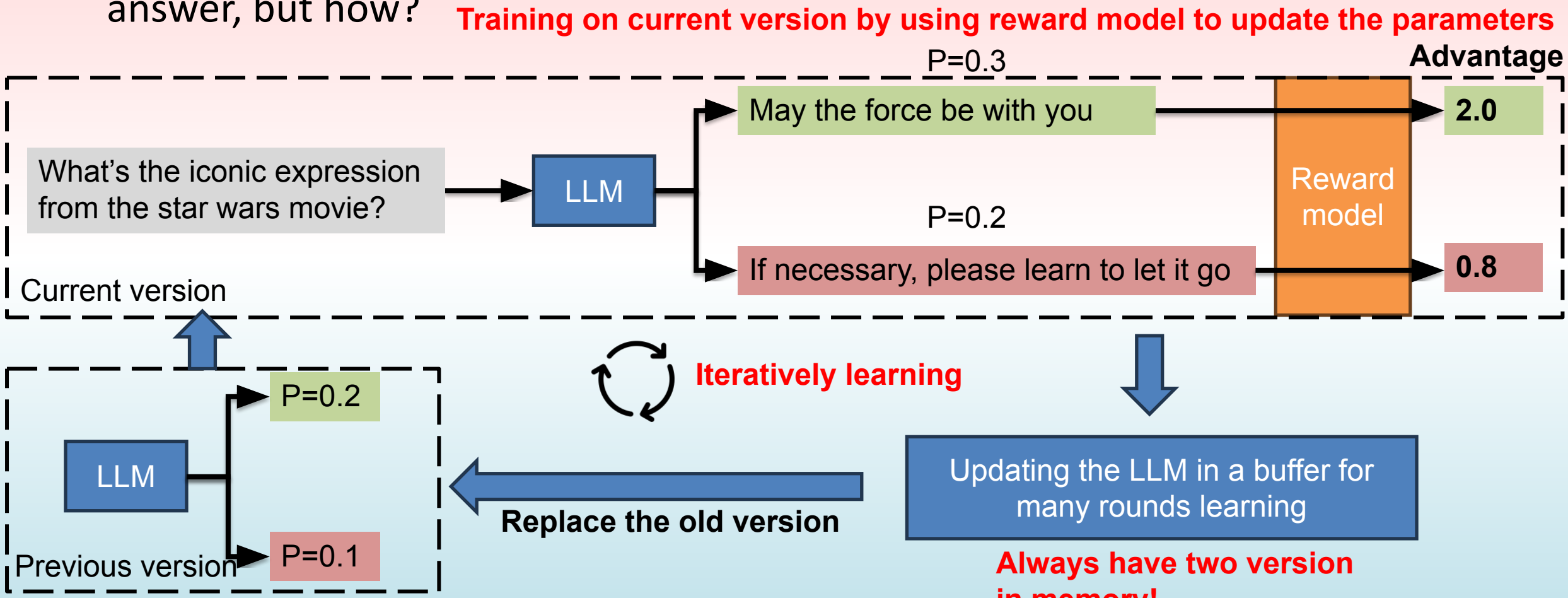
RLHF – more technical details

- We said that we want to increase the generative probability of human preferred answer, but how?

- Some principles in RLHF:
  - If an answer got a **high**/**low** score from reward model, we should **increase**/**decrease** the generative probability
  - During the updating parameters, we should not move too far from old parameters (to stabilize the training)

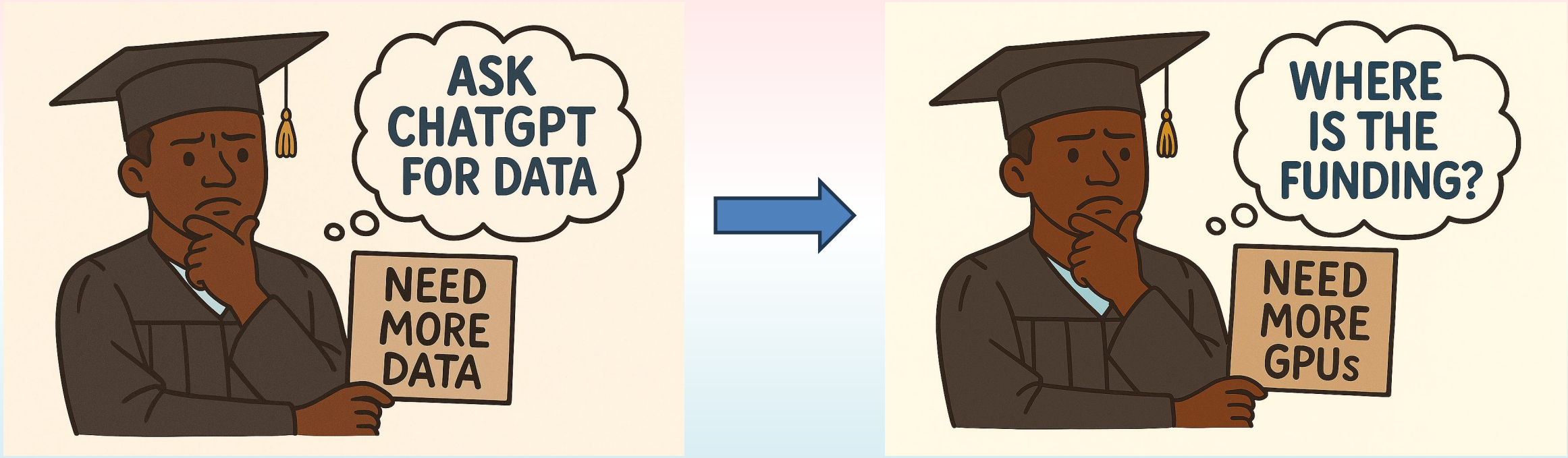
RLHF – more technical details (example)

- We said that we want to increase the generative probability of human preferred answer, but how?



RLHF – more technical details (example)

- New question:



RLHF – Train with Lora

- Who is LoRA?

LoRA: Low-Rank Adaptation of Large Language Models

Edward Hu* **Yelong Shen*** **Phillip Wallis** **Zeyuan Allen-Zhu**
Yuanzhi Li **Shean Wang** **Lu Wang** **Weizhu Chen**
Microsoft Corporation
{edwardhu, yeshe, phwallis, zeyuana,
yuanzhil, swang, luw, wzchen}@microsoft.com
yuanzhil@andrew.cmu.edu
(Version 2)

https://arxiv.org/pdf/2106.09685

RLHF – Train with Lora

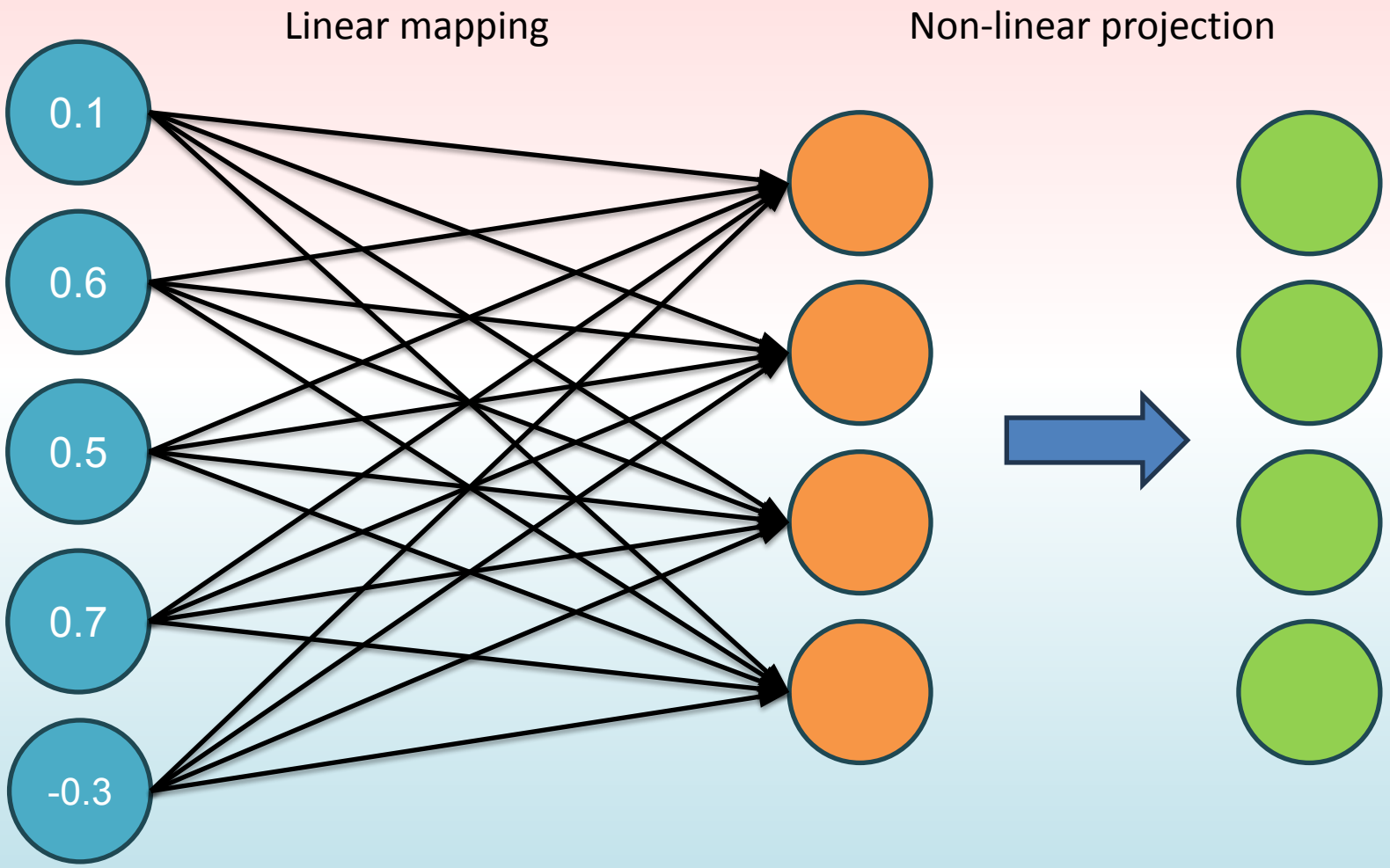
- The basic operations in neural networks:
  - Linear mapping
    - Parameter size M x N
    - Where M is the input dimension and N is the output dimension
  - Non-linear projection
    - Parameter size
    - Depends on the projection function

Memory costs are mainly from here

Using Matrix decomposition to reduce the size!

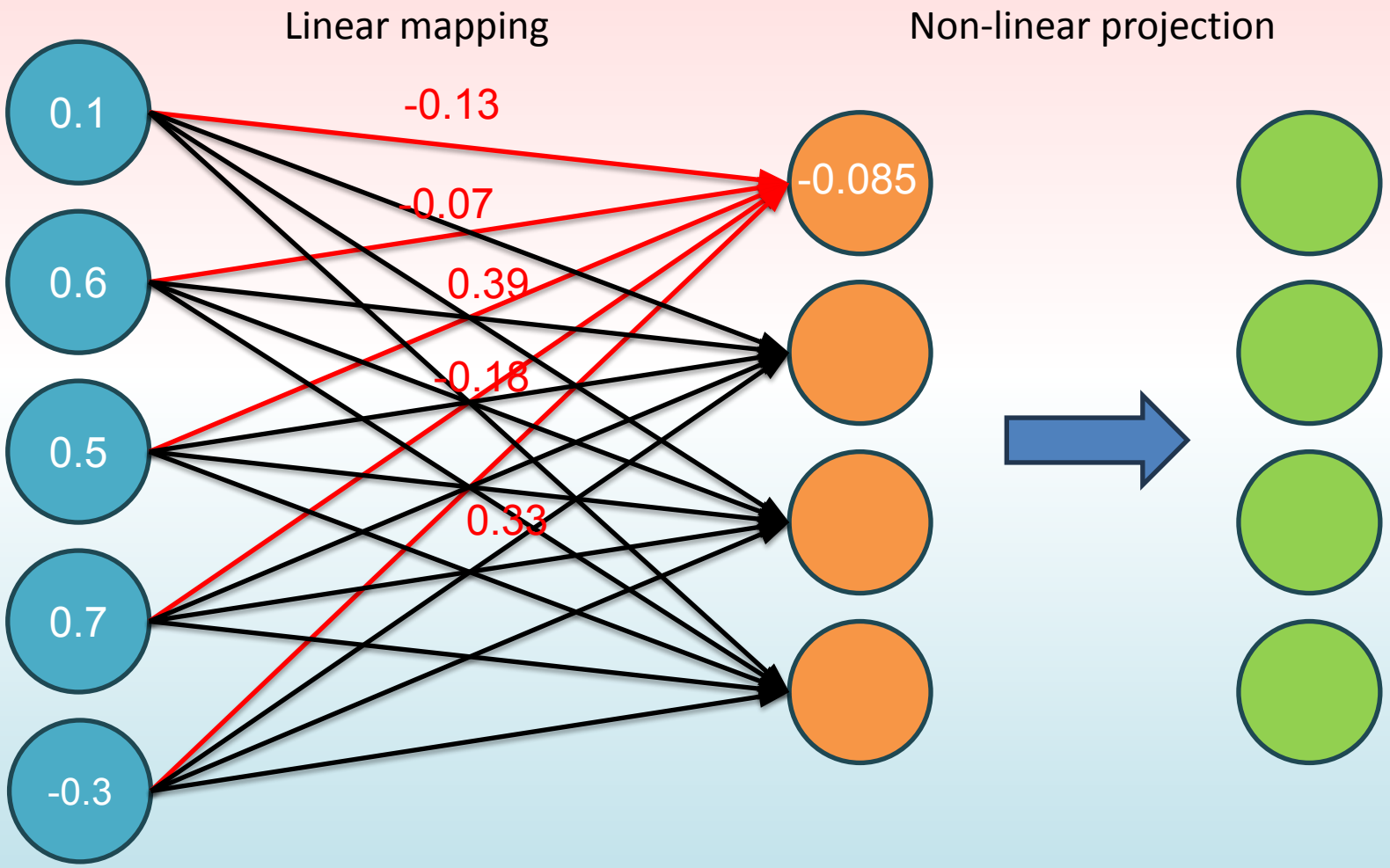
RLHF – Train with Lora (example)

- The basic operations in neural networks:



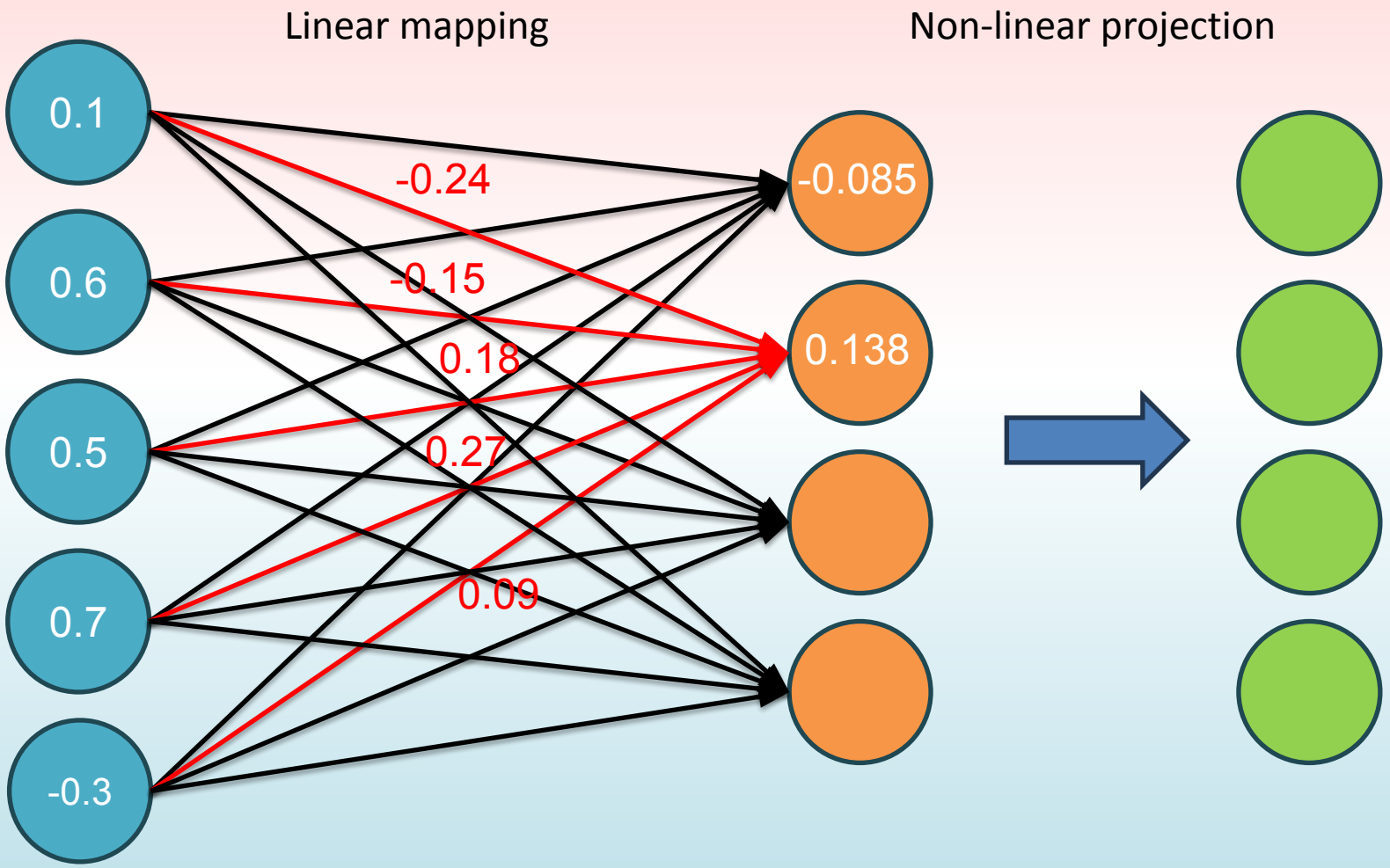
RLHF – Train with Lora (example)

- The basic operations in neural networks:



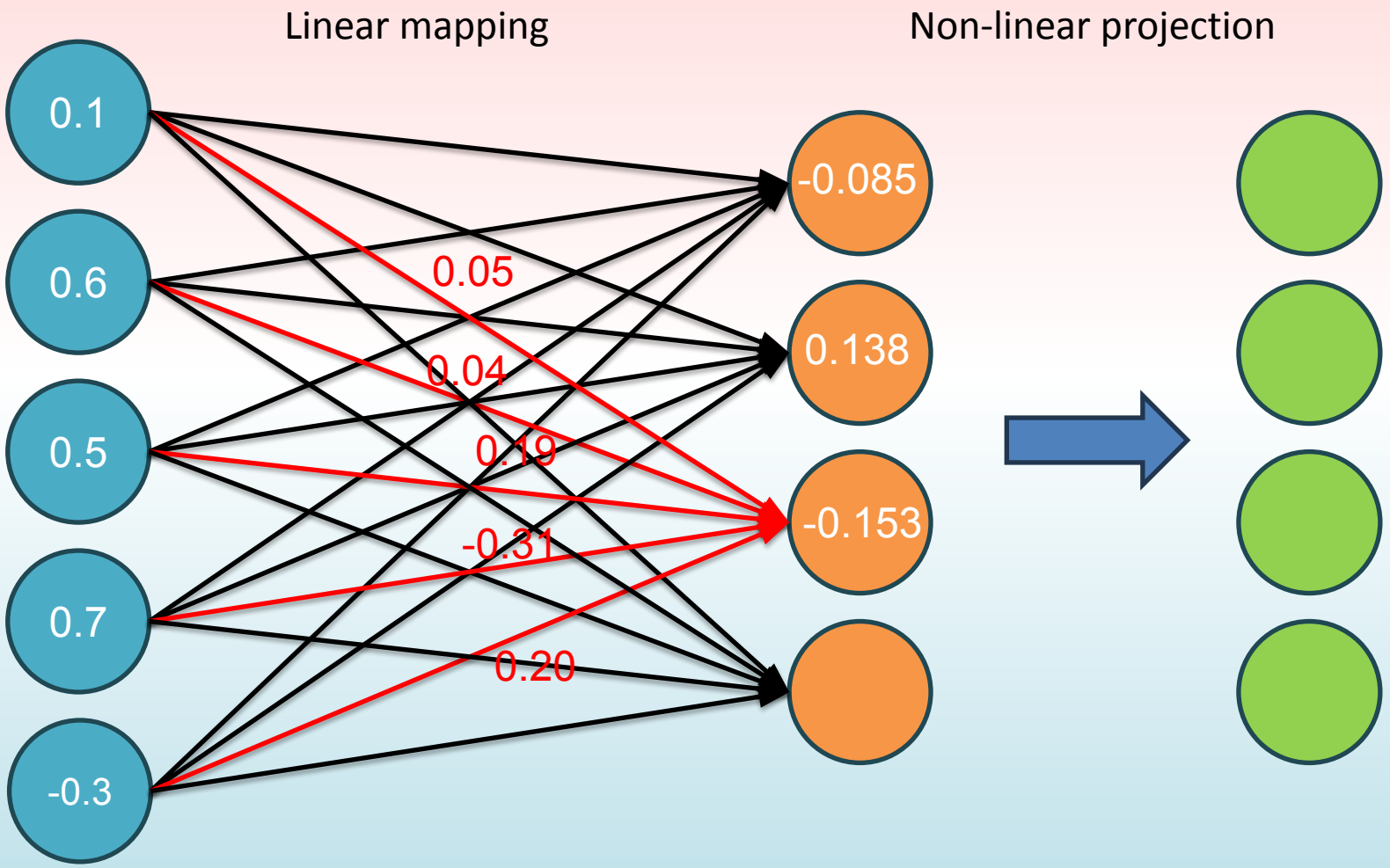
RLHF – Train with Lora (example)

- The basic operations in neural networks:



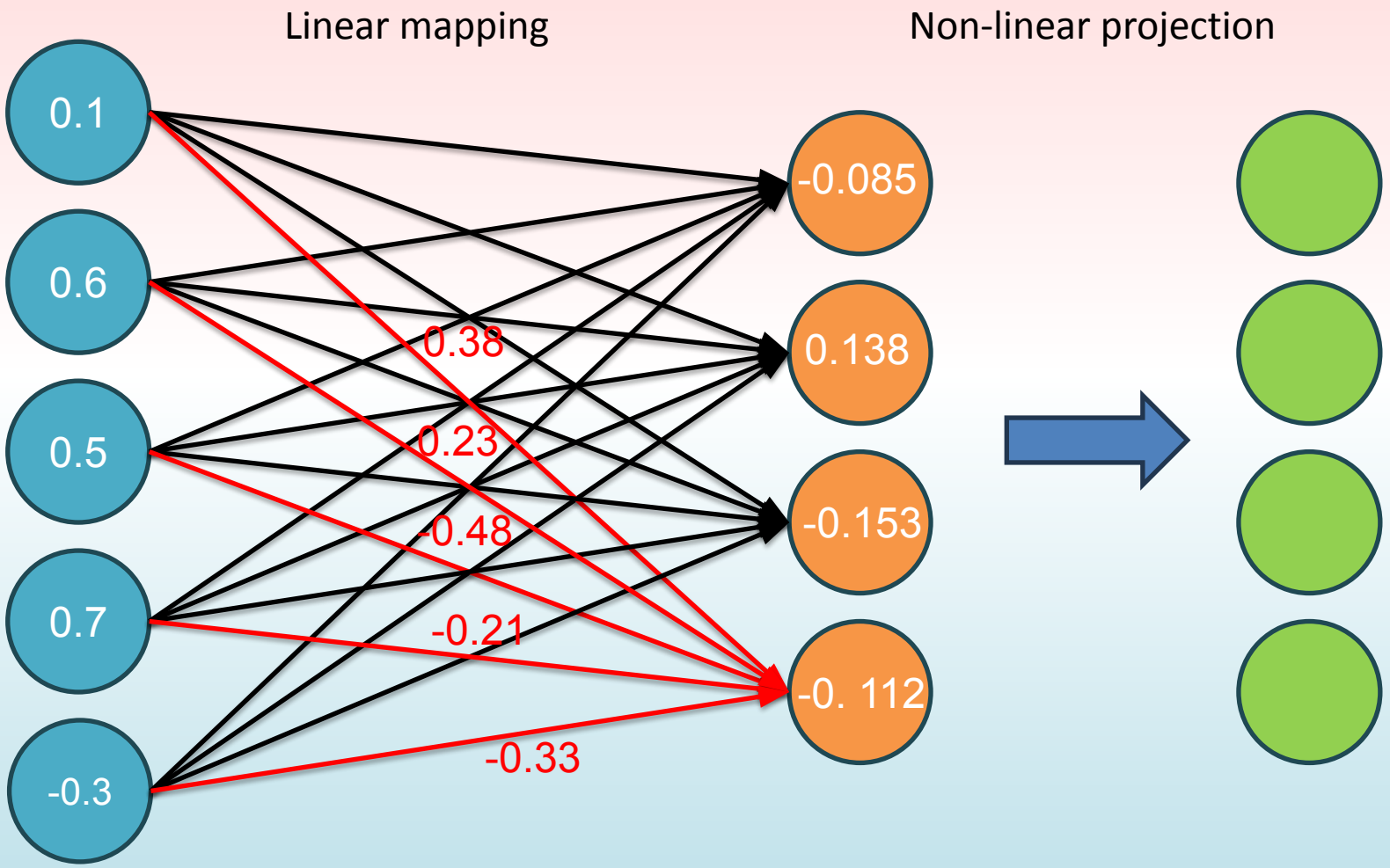
RLHF – Train with Lora (example)

- The basic operations in neural networks:



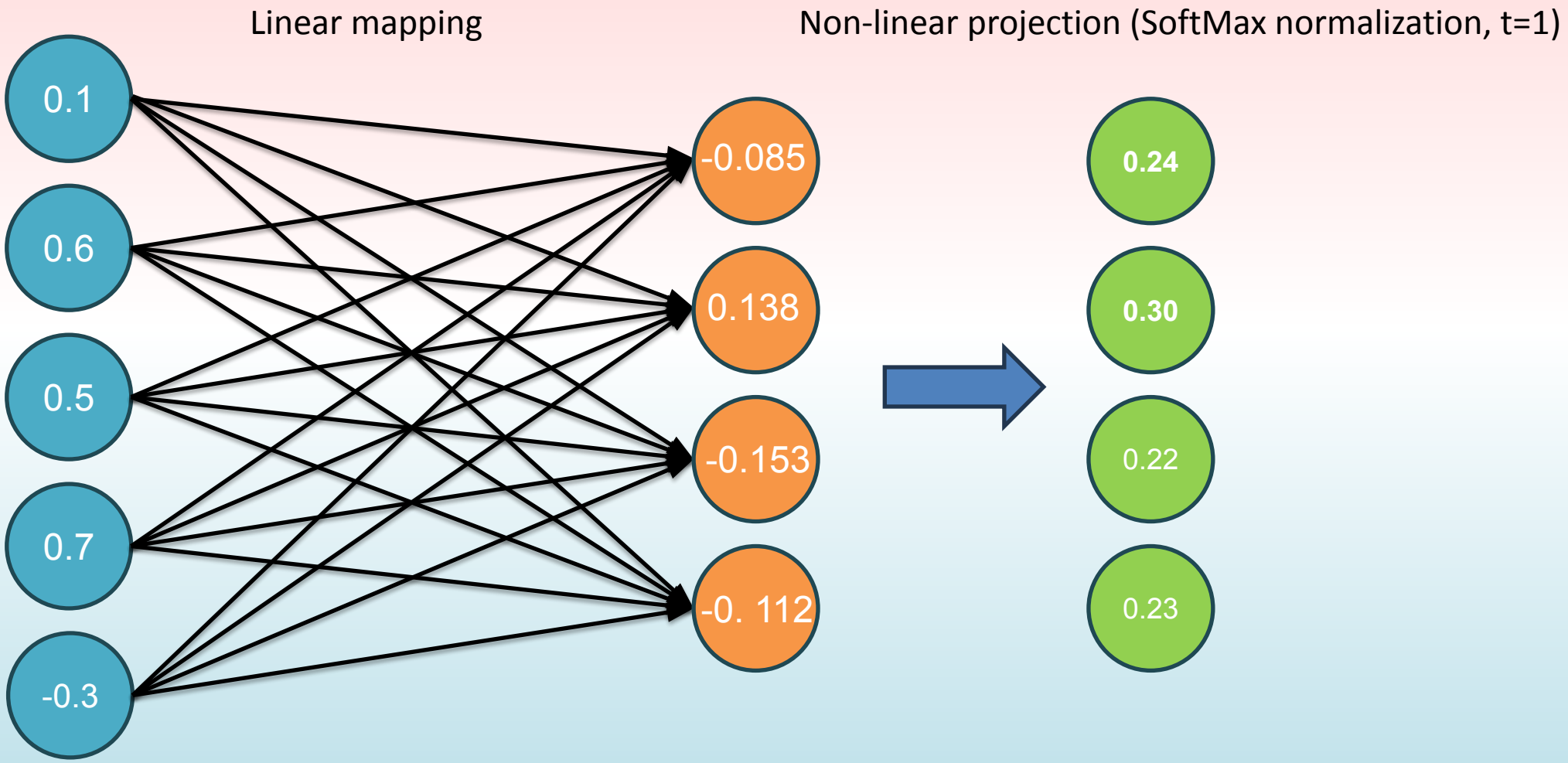
RLHF – Train with Lora (example)

- The basic operations in neural networks:



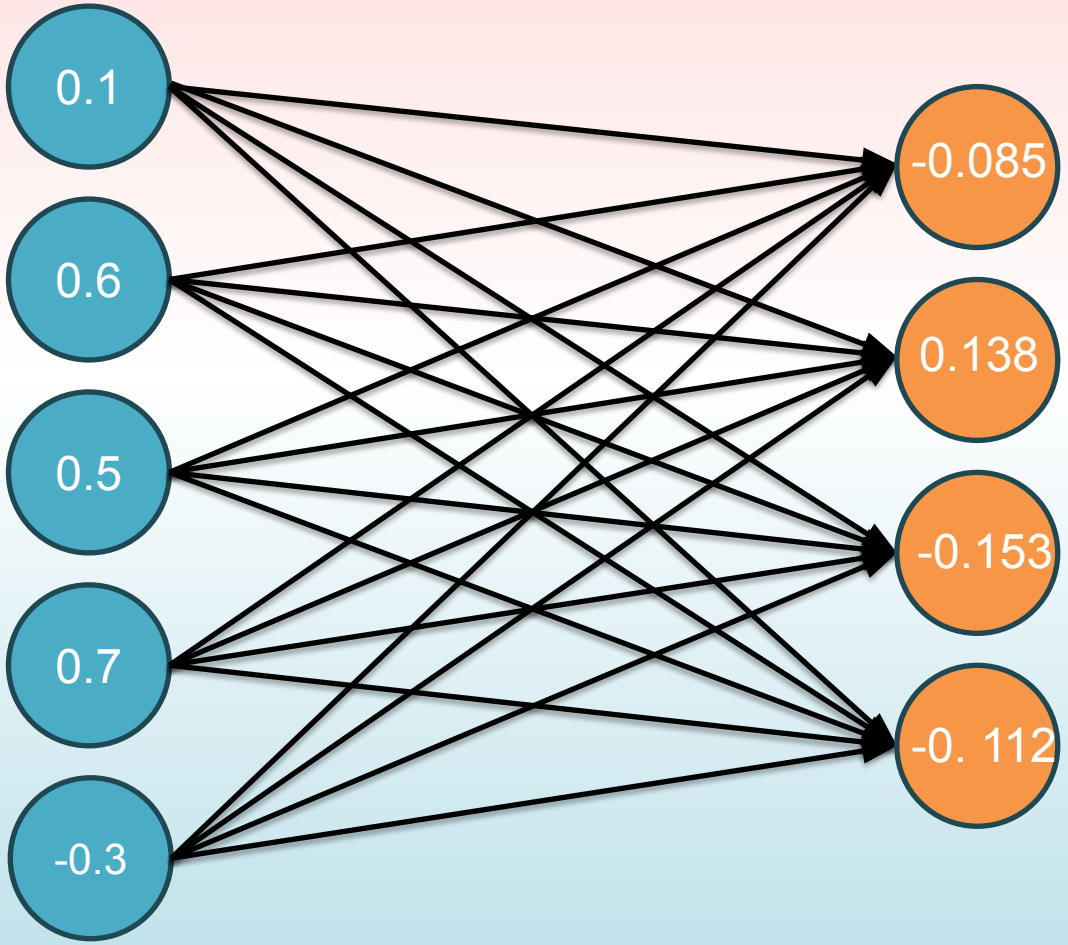
RLHF – Train with Lora (example)

- The basic operations in neural networks:



RLHF – Train with Lora (example)

- The basic operations in neural networks (Matrix version):



	A_1	A_2	A_3	A_4	A_5
1	0.1	0.6	0.5	0.7	-0.3

	B_1	B_2	B_3	B_4
1	-0.13	-0.24	0.05	0.38
2	-0.07	-0.15	0.04	0.23
3	0.39	0.18	0.19	-0.48
4	-0.18	0.27	-0.31	-0.21
5	0.33	0.09	0.2	-0.33

	C_1	C_2	C_3	C_4
1	-0.085	0.138	-0.153	-0.112

A x B = C

RLHF – Train with Lora (example)

- The basic operations in neural networks (Matrix version):

- How many parameters here?
- 20 (4*8)

Input

	A_1	A_2	A_3	A_4	A_5
1	0.1	0.6	0.5	0.7	-0.3

Para.

	B_1	B_2	B_3	B_4
1	-0.13	-0.24	0.05	0.38
2	-0.07	-0.15	0.04	0.23
3	0.39	0.18	0.19	-0.48
4	-0.18	0.27	-0.31	-0.21
5	0.33	0.09	0.2	-0.33

	C_1	C_2	C_3	C_4
1	-0.085	0.138	-0.153	-0.112

Output

A x B = C

RLHF – Train with Lora (example)

- The basic operations in neural networks (Matrix version):

	A_1	A_2	A_3	A_4	A_5
1	0.1	0.6	0.5	0.7	-0.3

	L_1	L_2
1	0.3	-0.5
2	0.2	-0.3
3	0.1	0.7
4	-0.7	0.2
5	0.2	0.5

	R_1	R_2	R_3	R_4
1	0.4	-0.3	0.5	0.1
2	0.5	0.3	0.2	-0.7

	B_1	B_2	B_3	B_4
1	-0.13	-0.24	0.05	0.38
2	-0.07	-0.15	0.04	0.23
3	0.39	0.18	0.19	-0.48
4	-0.18	0.27	-0.31	-0.21
5	0.33	0.09	0.2	-0.33

L x R = B

	C_1	C_2	C_3	C_4
1	-0.085	0.138	-0.153	-0.112

A x B = C

RLHF – Train with Lora (example)

- The basic operations in neural networks (Matrix version):

- How many parameters in B?
- 20 (4*8)

- How many parameters in L and R?
- 18 (5*2 + 2*4)

	A_1	A_2	A_3	A_4	A_5
1	0.1	0.6	0.5	0.7	-0.3

	L_1	L_2
1	0.3	-0.5
2	0.2	-0.3
3	0.1	0.7
4	-0.7	0.2
5	0.2	0.5

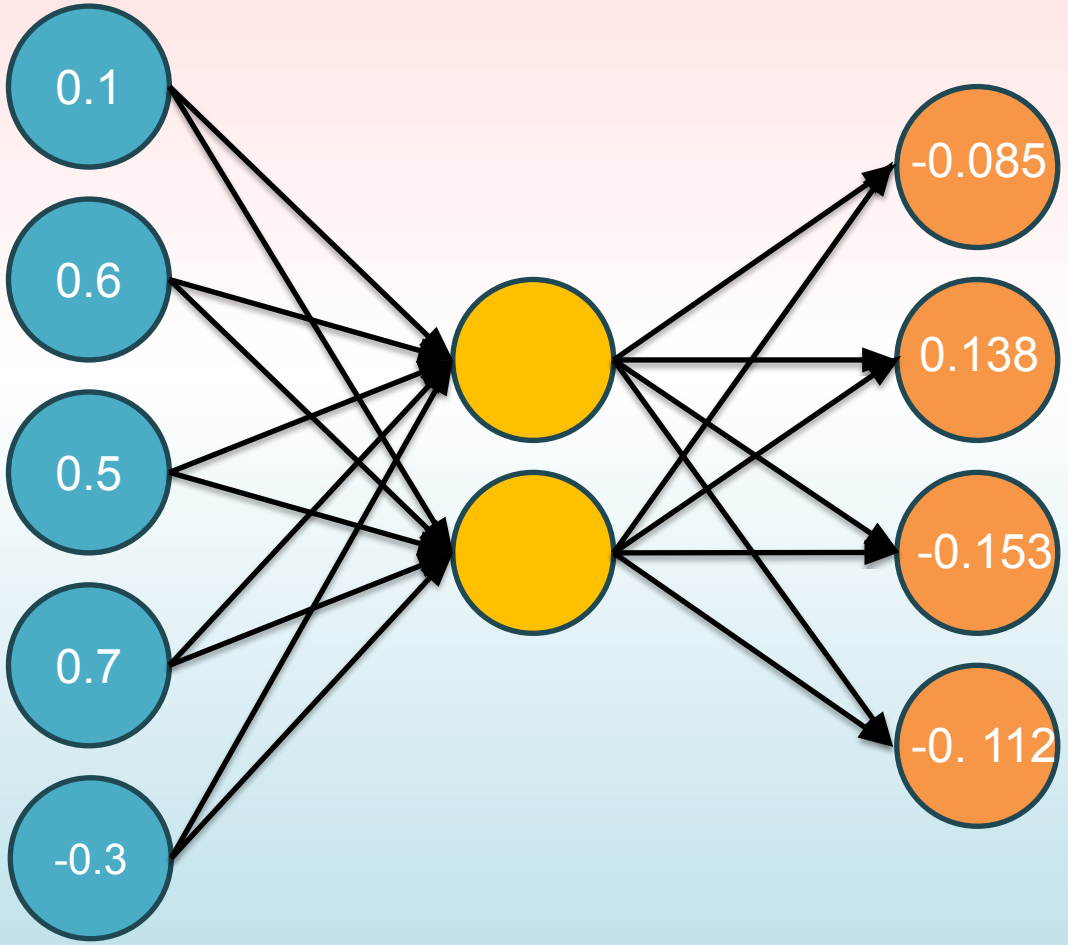
	R_1	R_2	R_3	R_4
1	0.4	-0.3	0.5	0.1
2	0.5	0.3	0.2	-0.7

	C_1	C_2	C_3	C_4
1	-0.085	0.138	-0.153	-0.112

A x L x R = C

RLHF – Train with Lora (example)

- The basic operations in neural networks (NN version):



	A_1	A_2	A_3	A_4	A_5
1	0.1	0.6	0.5	0.7	-0.3

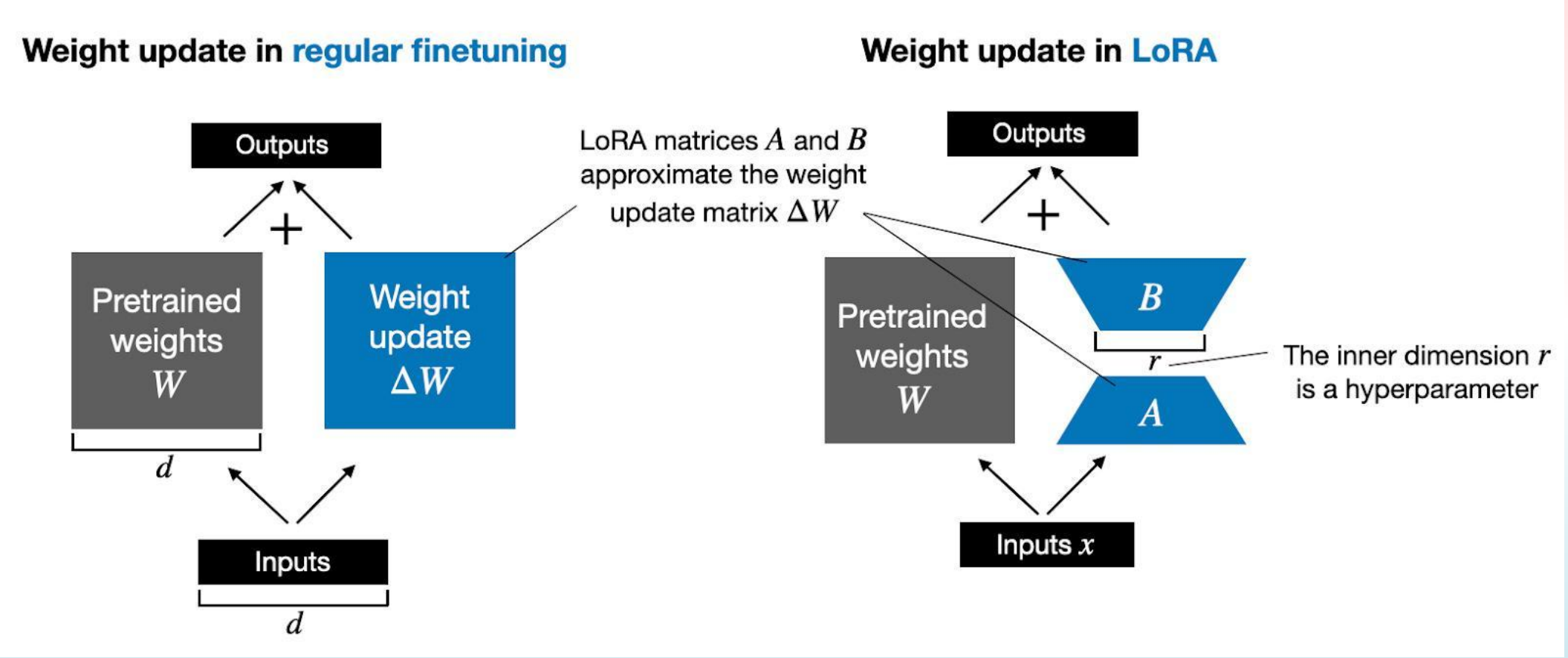
	L_1	L_2
1	0.3	-0.5
2	0.2	-0.3
3	0.1	0.7
4	-0.7	0.2
5	0.2	0.5

	R_1	R_2	R_3	R_4
1	0.4	-0.3	0.5	0.1
2	0.5	0.3	0.2	-0.7

	C_1	C_2	C_3	C_4
1	-0.085	0.138	-0.153	-0.112

RLHF – Train with Lora

- Framework:



RLHF – Train with Lora

- Discussion:
- By using two learning mapping **L: m x r**, and **R: r x n**, to replace a linear mapping: **m x n** (Here, m and n are the dimensions of mapping function, r is the rank in LoRA, and r << m or n)
- If r is small enough, the size of **m x r + r x n** is **significantly** less than **m x n**
- We only need very small size of parameters to train a large model
- Do you think it is a good idea? Any cons?

RLHF – Train with Lora

- Discussion:
- Do you think it is a good idea? Any cons?
- Example on MedMnist dataset
- **No free-lunch principle** in Machine Learning. So, what's the price?
  - The drop of accuracy
  - The efficiency cannot be guaranteed as well.

```
ACC: 0.27 – Running time: 5 min 27 sec – r=3

ACC: 0.81 – Running time: 5 min 41 sec - r=10
```

https://colab.research.google.com/drive/14FppHOAvi5mUX7VXEIedIuu0dgY_rZJj?usp=sharing

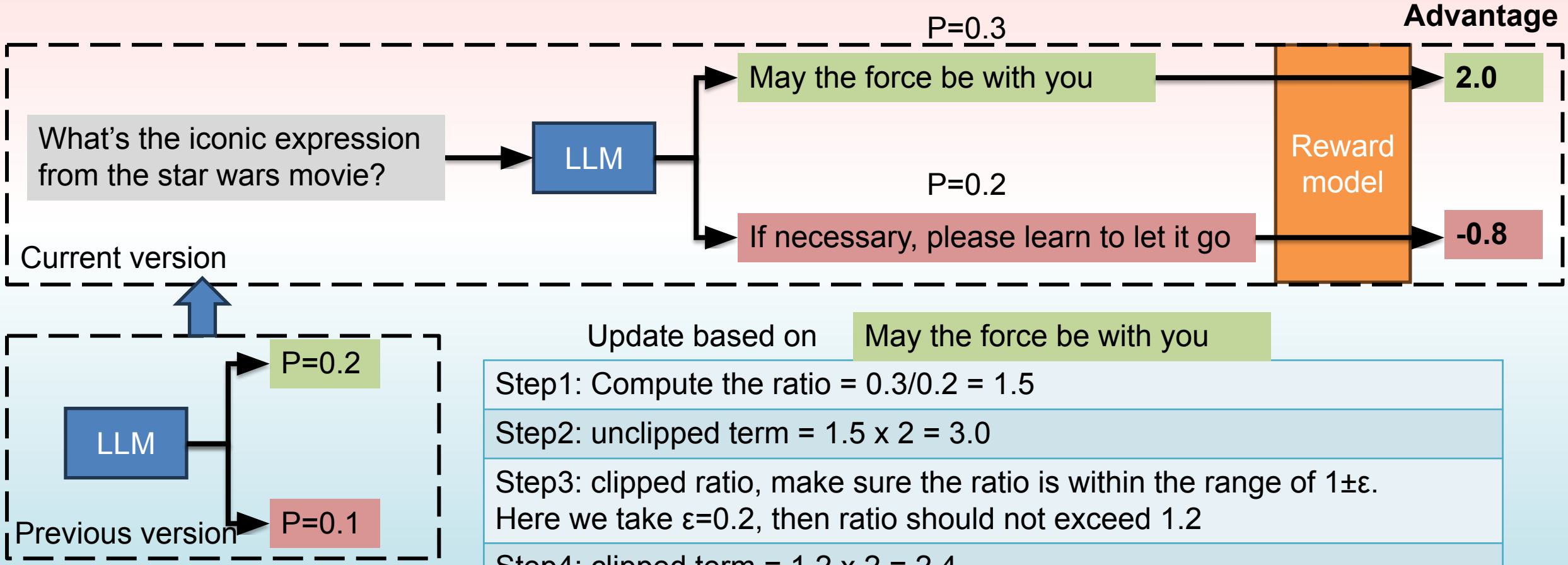
RLHF – Train with Lora

- Discussion:
- If the original setting is trainable, maybe we don’t need to use LoRA
- If the original setting requires large memory (like PPO), the LoRA allow you to train a large model with limited memory.

https://colab.research.google.com/drive/14FppHOAvi5mUX7VXEIedIuu0dgY_rZJj?usp=sharing

RLHF – more technical details (example)

- We said that we want to increase the generative probability of human preferred answer, but how?



Update based on May the force be with you

Step1: Compute the ratio = 0.3/0.2 = 1.5
Step2: unclipped term = 1.5 x 2 = 3.0
Step3: clipped ratio, make sure the ratio is within the range of 1±ε. Here we take ε=0.2, then ratio should not exceed 1.2
Step4: clipped term = 1.2 x 2 = 2.4
Step5: take the minimum from 3.0 and 2.4, which is 2.4 as the objective

RLHF – more technical details (example)

Update based on May the force be with you

Step1: Compute the ratio = 0.3/0.2 = 1.5
Step2: unclipped term = 1.5 x 2 = 3.0
Step3: clipped ratio, make sure the ratio is within the range of 1±ε. Here we take ε=0.2, then ratio should not exceed 1.2
Step4: clipped term = 1.2 x 2 = 2.4
Step5: take the minimum from 3.0 and 2.4, which is 2.4 as the objective

Ratio > 1, the current model tends to generate this sentence. If this indicates a positive advantage, extensive updating is not necessary.

Estimate the objective based on the ratio

We don't want the ratio is too high otherwise the model will keep focusing on this single case

For the same we take the minimum objective

P_current	0.3
P_previous	0.2
Advantage	2.0
ε	0.2

What if we have lower probability in the current language model?

RLHF – more technical details (example)

Update based on May the force be with you

Step1: Compute the ratio = **0.2/0.3** = **0.67**
Step2: unclipped term = **0.67** x 2 = 1.34
Step3: clipped ratio, make sure the ratio is within the range of 1±ε. Here we take ε=0.2, then ratio should less than **0.8**
Step4: clipped term = **0.8** x 2 = 1.6
Step5: take the minimum from 1.34 and 1.6, which is 1.34 as the objective

Ratio < 1, the current model doesn't tend to generate this sentence. If this indicates a positive advantage, extensive updating is necessary.

Estimate the objective based on the ratio

We don't want the ratio is too low otherwise the model will keep focusing on this single case

For the same we take the minimum objective

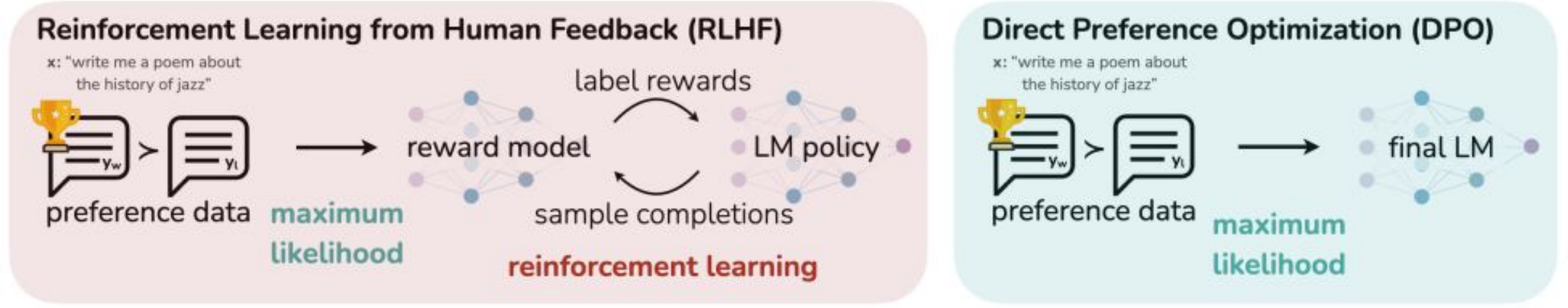
P_current	**0.2**
P_previous	**0.3**
Advantage	2.0
ε	0.2

Formal definition:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min(r_t(\theta)\,A_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,A_t)\right]$$

Reinforcement Learning with Human Feedback (RLHF)

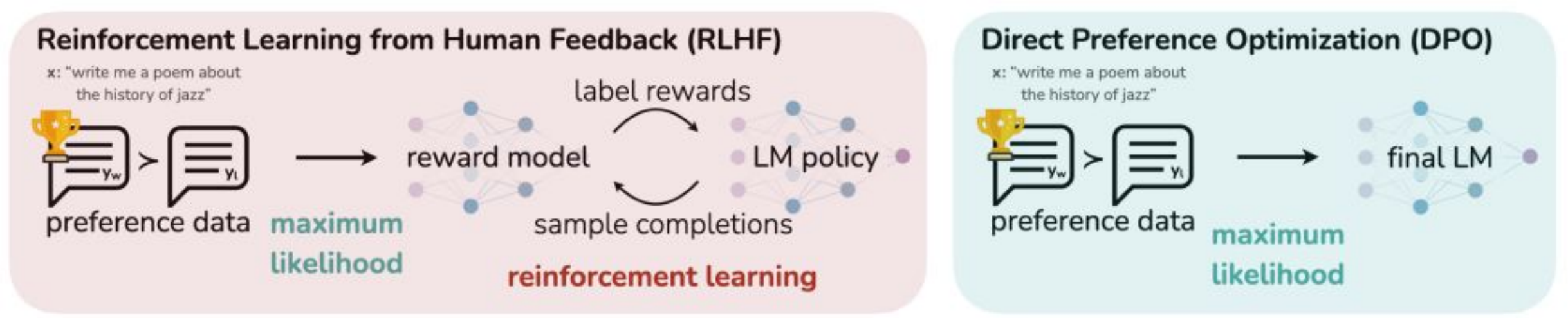
- It is still complicated.
- We need to simulate feedback: generate → get reward → do policy gradient → clip updates.
- Can we simplify the progress?
- Yes!



Direct Preference Optimisation

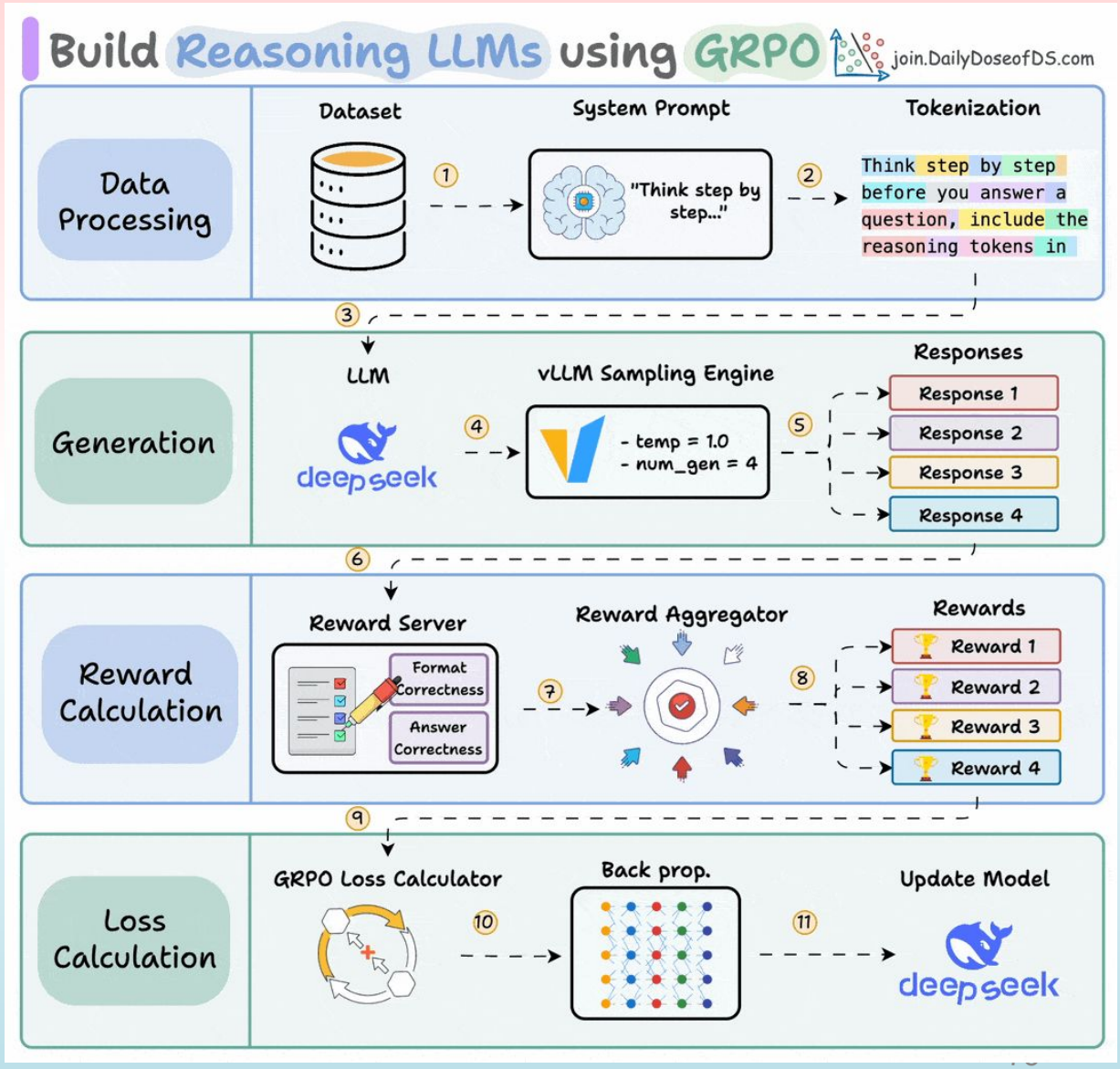
- We already know which answer humans like
- Just directly make those more likely than the bad ones.

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)}\left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \log\frac{\pi_\theta(y^-|x)}{\pi_{\text{ref}}(y^-|x)}\right)\right)\right]$$



Group Relative Policy Optimization

- Do not need a learned value function / critic to estimate advantages (simple)
- Sample groups of responses (actions) from a given prompt (state) and compute relative advantages across that group. (stable)
- Given a prompt, generate multiple candidate responses and compute statistics (mean, std) over the group.
- Then define each response's advantage as something like how much better it performed relative to its peers.
- The function could be very simple like the length of sentence.



Further discussion

- There are still many unsolved problem in the LLM
- For example,
  - Some aspects in the reward function cannot be clearly defined: Helpfulness vs Safety.
  - It would be hard for user to identify the better answer
  - Lacks stability: if you change your prompt, you will get a very different answer
  - ……

Thank you

Lin.1.Gui@kcl.ac.uk

www.kcl.ac.uk/people/lin-gui