LLM LE5 VT2026

Fine-tuning and Alignment

Fredrik Heintz
Dept. of Computer Science
Linköping University
fredrik.heintz@liu.se
@FredrikHeintz

Outline:

- Alignment
- Instruction fine-tuning
- Preference tuning (RLHF, DPO)
- Pruning and Distilling

LINKÖPING UNIVERSITY

Language modelling

- **Language modelling** is the task of predicting which word comes next in a sequence of words.
- More formally, given a sequence of words $w_1, \ldots, w_t$ we want to know the probability of the next word, w_{t+1}:

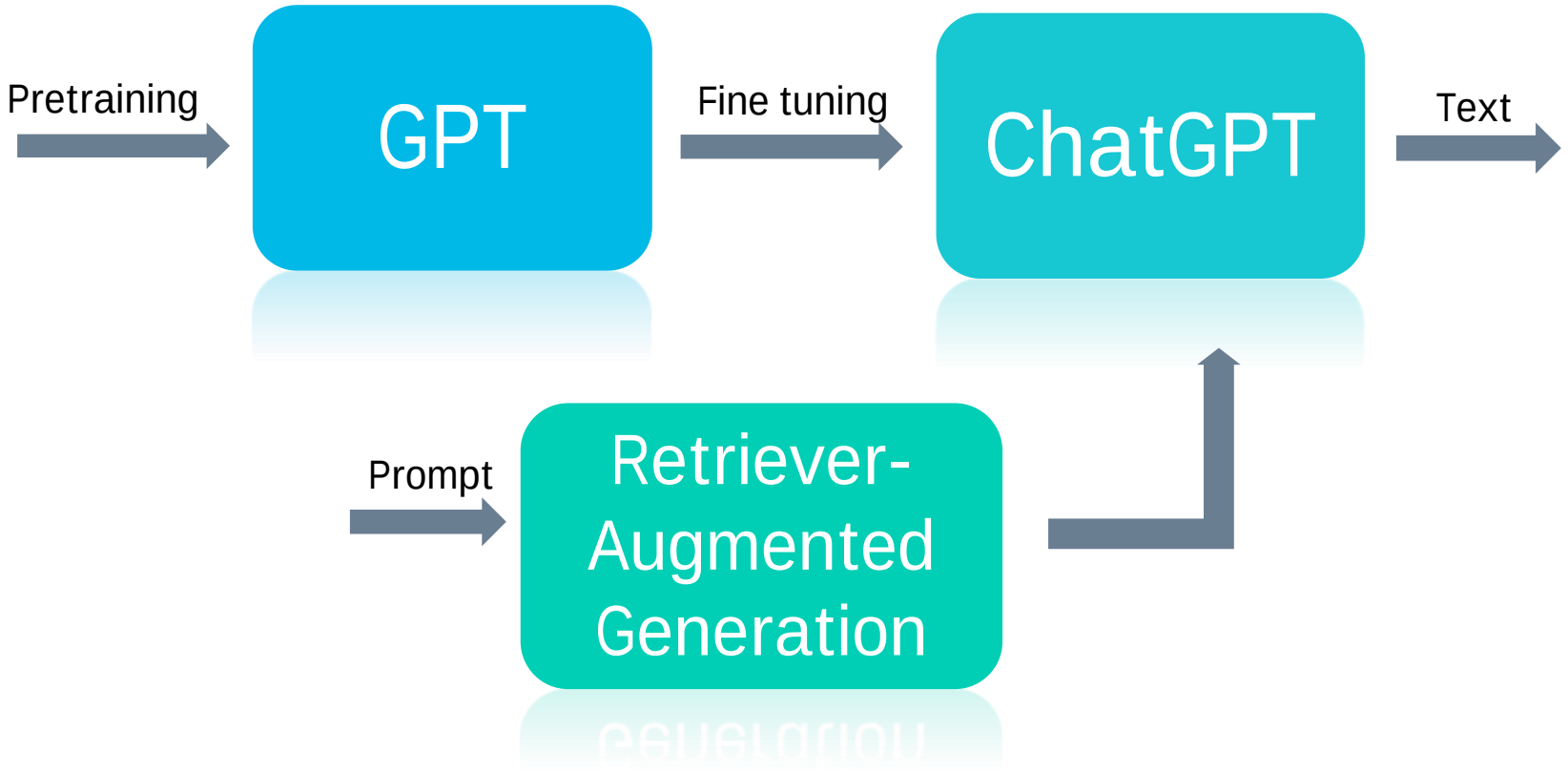
$$P(w_{t+1} \mid w_1, \ldots, w_t)$$

- We are assuming that w_{t+1} comes from a finite vocabulary V.

language models = classifiers

Funded by the European Union

How Does ChatGPT Work?



Funded by the European Union

Motivation of instruction finetuning

Language modeling ≠ assisting users

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

COMPLETION

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.
```

InstructGPT

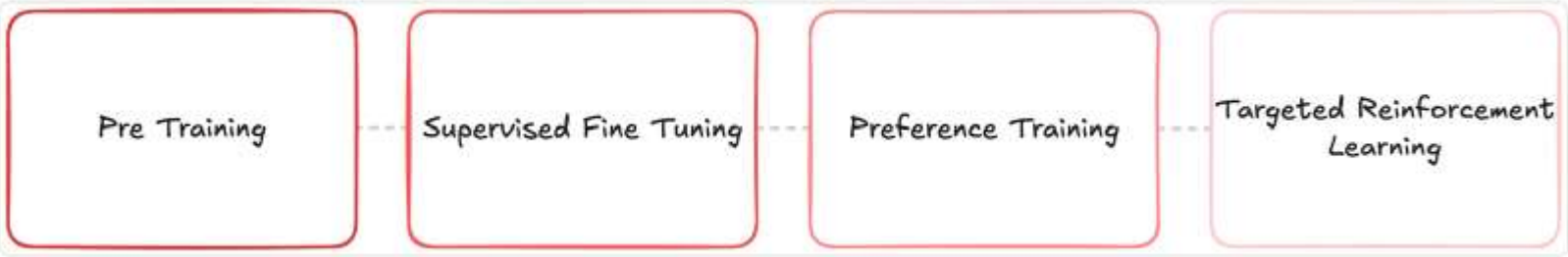
```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

Language models are not *aligned* with user intent.
Do complemetion instead of instruction folowing

Ouyang, Long, et al. "Training language models to follow instructions with human feedback, 2022." https://arxiv.org/abs/2203.02155 13 (2022).

Funded by the European Union

Post-Training

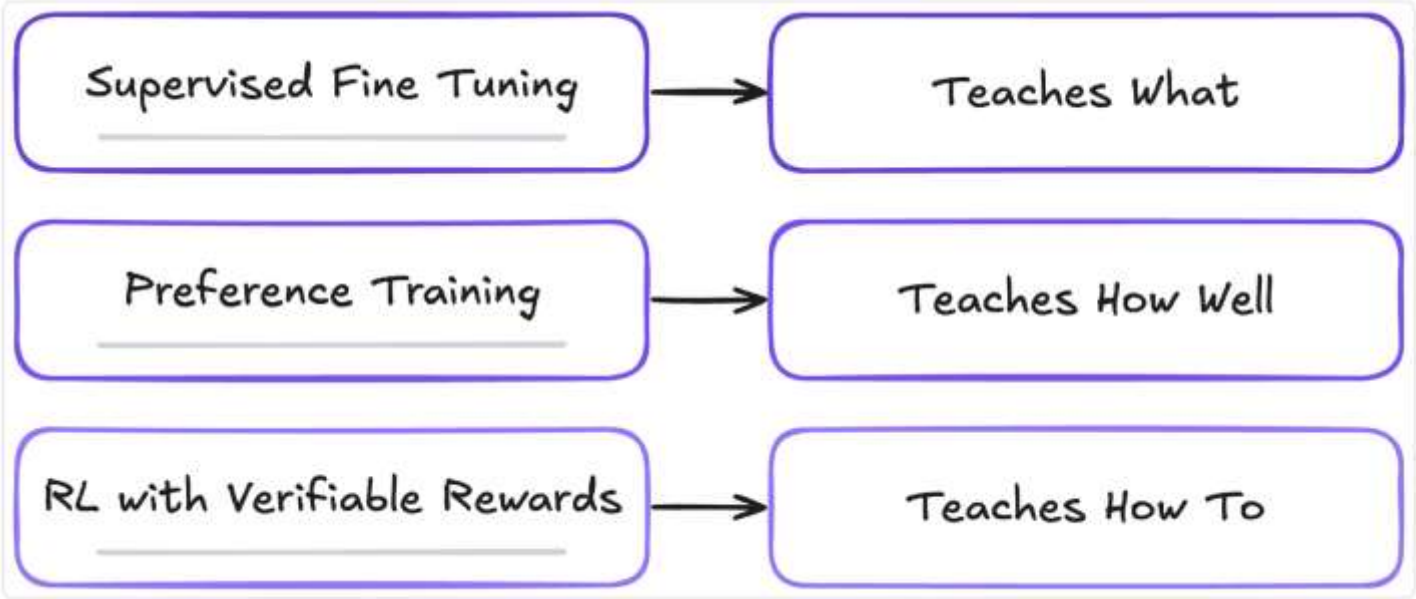


- Pre-Training a language model using next-token prediction at scale.
- SFT for base instruction-following + tools + formatting.
- Preference Training to push general answer quality and safety (RLHF, DPO/KTO or reward-model + small PPO).
- Targeted RL where you have strong rewards (coding, math, tool-using agents, browsing).
- Continual Refresh with new data, procedures, and preferences as they become available.

Funded by the European Union

Post-Training

- Supervised Fine-Tuning (SFT) to teach formats and tasks by imitation,
- Preference Training to prefer better answers over worse ones, and
- Reinforcement Learning (RL) to optimize for sequence-level goals and generalize procedures.

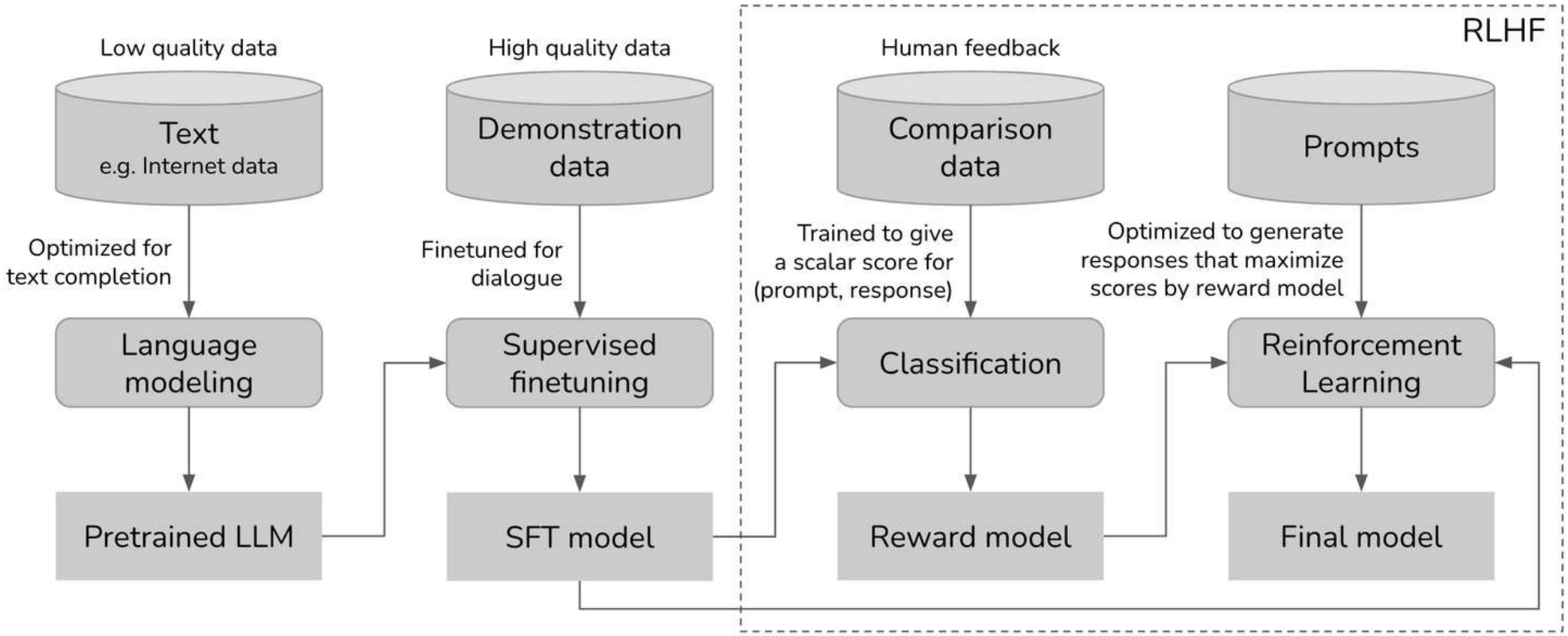


Funded by the European Union

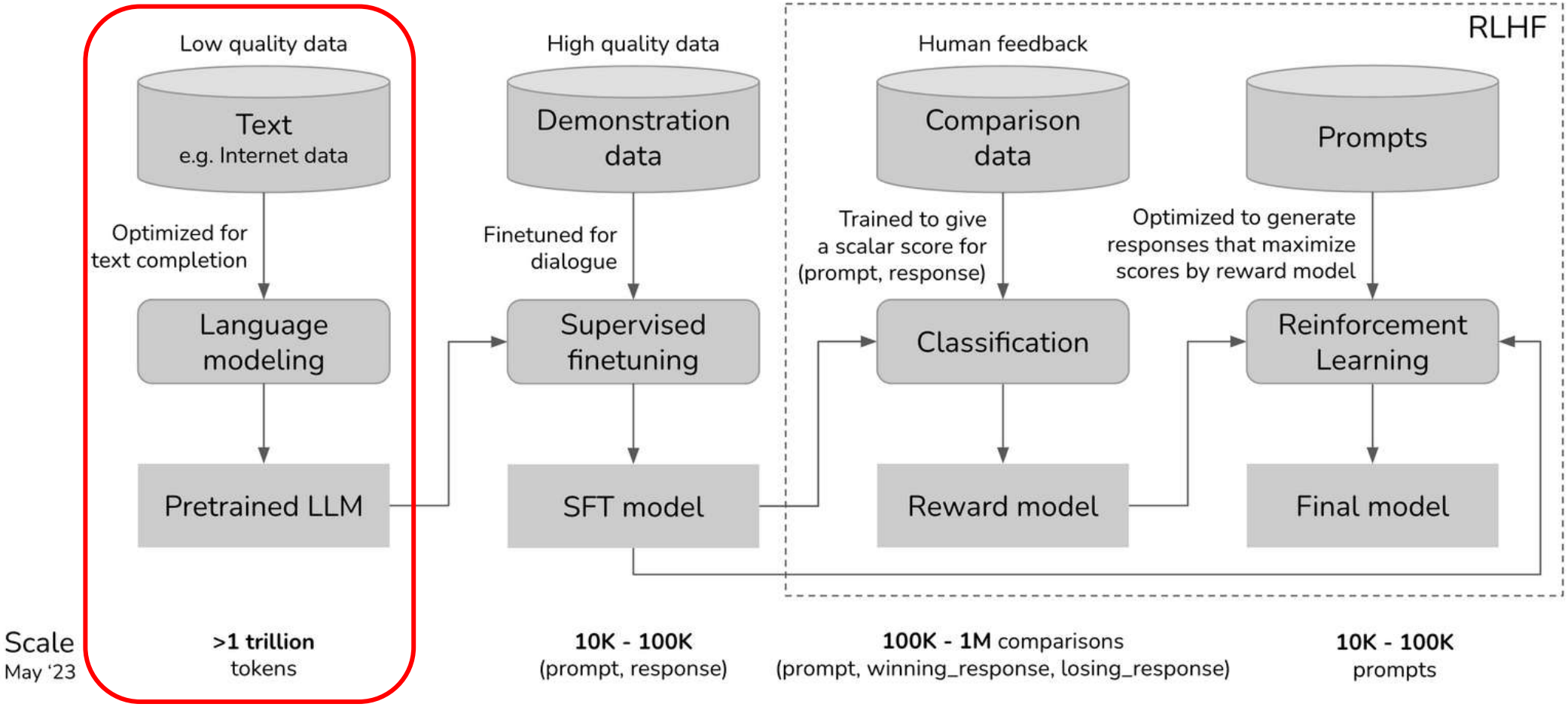
Figure 2 | RLHF and DPO have been the two predominant RL methodologies for human alignment in recent years. In contrast, RLVR represents an emerging trend in RL for LRMs, significantly enhancing their capacity for complex task solving. The next stage of scaling RL for LLMs remains an open question, with open-ended RL presenting a particularly challenging and promising direction.

Funded by the European Union

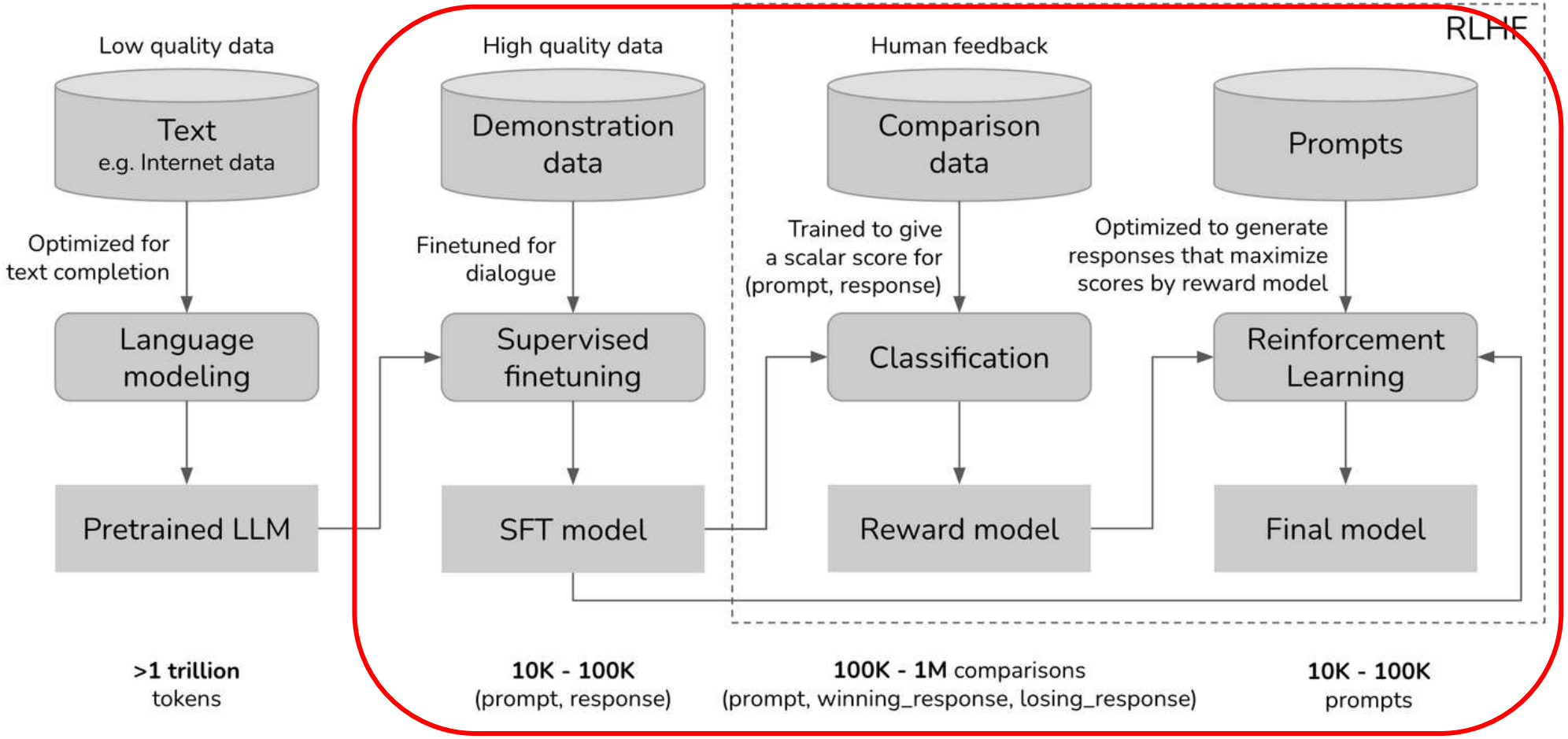
High-level view of LLM training



Pre-training



High-level view of LLM training



Alignment

"**AI alignment** aims to steer AI systems toward a person's or group's intended goals, preferences, and ethical principles. An AI system is considered *aligned* if it advances the intended objectives. A *misaligned* AI system pursues unintended objectives."

- Recent research shows again and again that it is crucial to have **high-quality instruct fine-tuning data** for reinforcement learning.
- The *Less Is More for Alignment (LIMA)* paper by Zhou et al. (2023) demonstrated fine-tuning on just **1,000 high-quality examples** improved a model's instruction following, highlighting data quality over quantity.
- Furthermore, it is important to have **diverse data** written by a diverse group of people.

Funded by the European Union

Motivation: Alignment

The three H's of Model Desiderata

Funded by the European Union

Motivation: Alignment

The three H's of Model Desiderata

- **Helpful:**
  - The AI should help the user solve their task (e.g. answer their questions)

Funded by the European Union

Motivation: Alignment

The three H's of Model Desiderata

- **Helpful:**
  - The AI should help the user solve their task (e.g. answer their questions)
- **Honest:**
  - The AI should give accurate information
  - The AI should express uncertainty when the model doesn't know the answer, instead of hallucinating a wrong answer

Funded by the European Union

Motivation: Alignment

The three H's of Model Desiderata

- **Helpful:**
  - The AI should help the user solve their task (e.g. answer their questions)
- **Honest:**
  - The AI should give accurate information
  - The AI should express uncertainty when the model doesn't know the answer, instead of hallucinating a wrong answer
- **Harmless:**
  - The AI should not cause physical, psychological, or social harm to people or the environment

Funded by the European Union

How do we align an LLM?

1. Foundation model

Pre-train on vast amounts of data

2. Instruction fine-tuning

Learning from task-specific examples

3. Preference tuning

Feedback and reinforcement

Funded by the European Union

How do we align an LLM?

Iceland

Article Talk

From Wikipedia, the free encyclopedia

This article is about the country. For other uses, see Iceland (disambiguation).

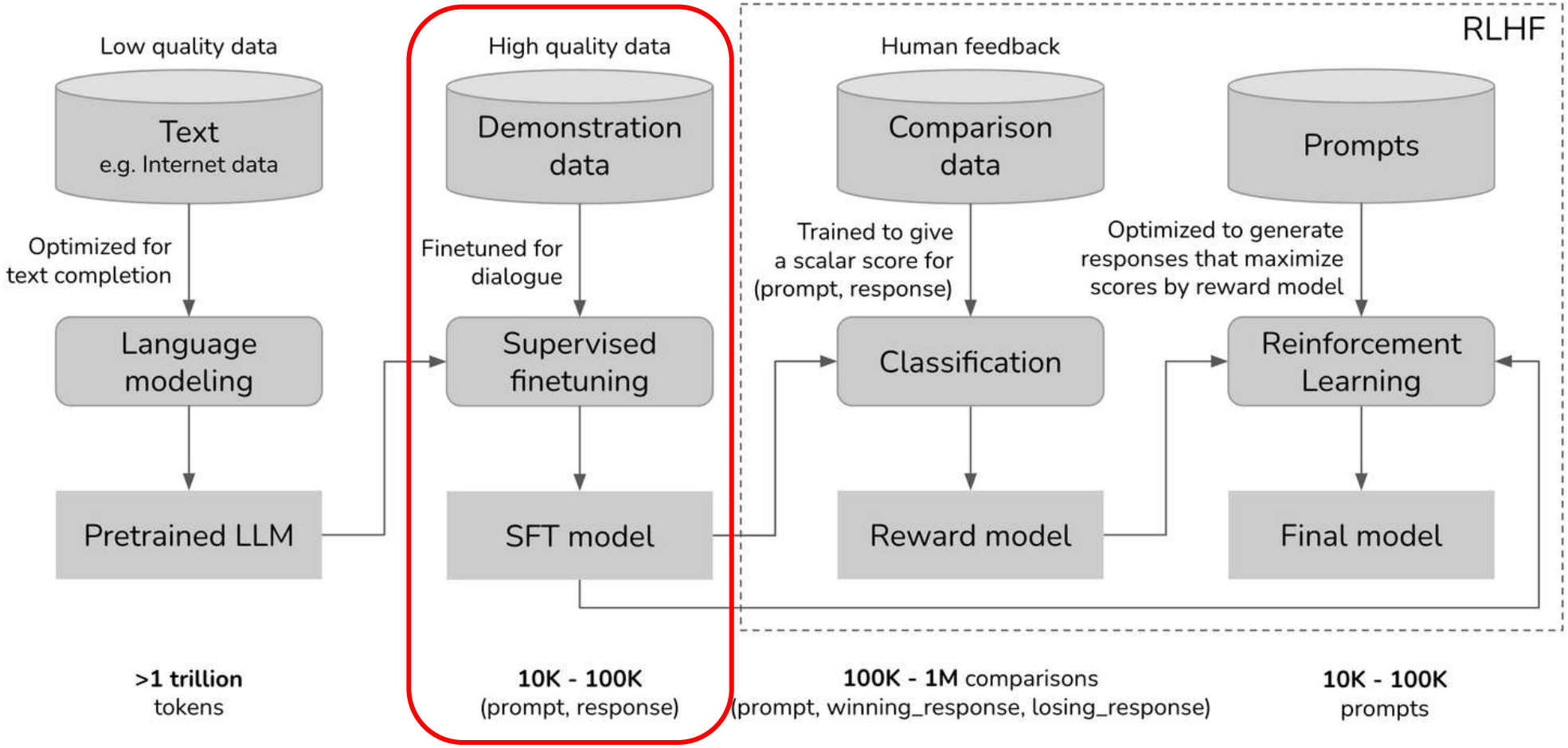
Iceland (Icelandic: *Ísland*, pronounced ['istlant])[d] is a Nordic island country between the North Atlantic and Arctic Oceans, on the Mid-Atlantic Ridge between North America and Europe. It is culturally and politically linked with Europe and the region's most sparsely populated country.[12] Its capital and largest city is Reykjavík, which is home to about 36% of the country's roughly 380,000 residents. The official language of the country is Icelandic.

1. Foundation model training data

{input: “Iceland (Icelandic: Ísland, pronounced ['istlant])[d] is a Nordic island country between the North Atlantic and Arctic Oceans, on the Mid-Atlantic Ridge between North America and Europe…”,}

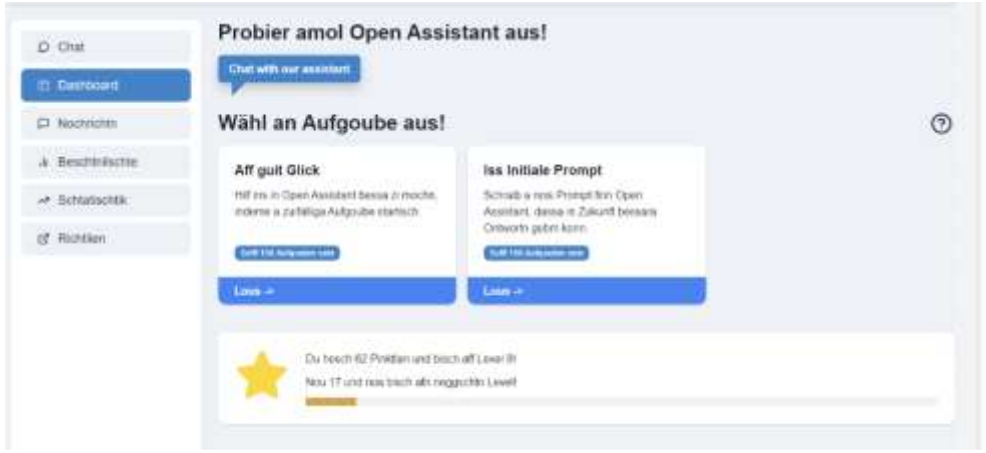
Funded by the European Union

Instruction tuning/Supervised fine-tuning (SFT)



Funded by
the European Union

How do we align an LLM?



OpenHermes-2.5

2. Instruction fine-tuning training data

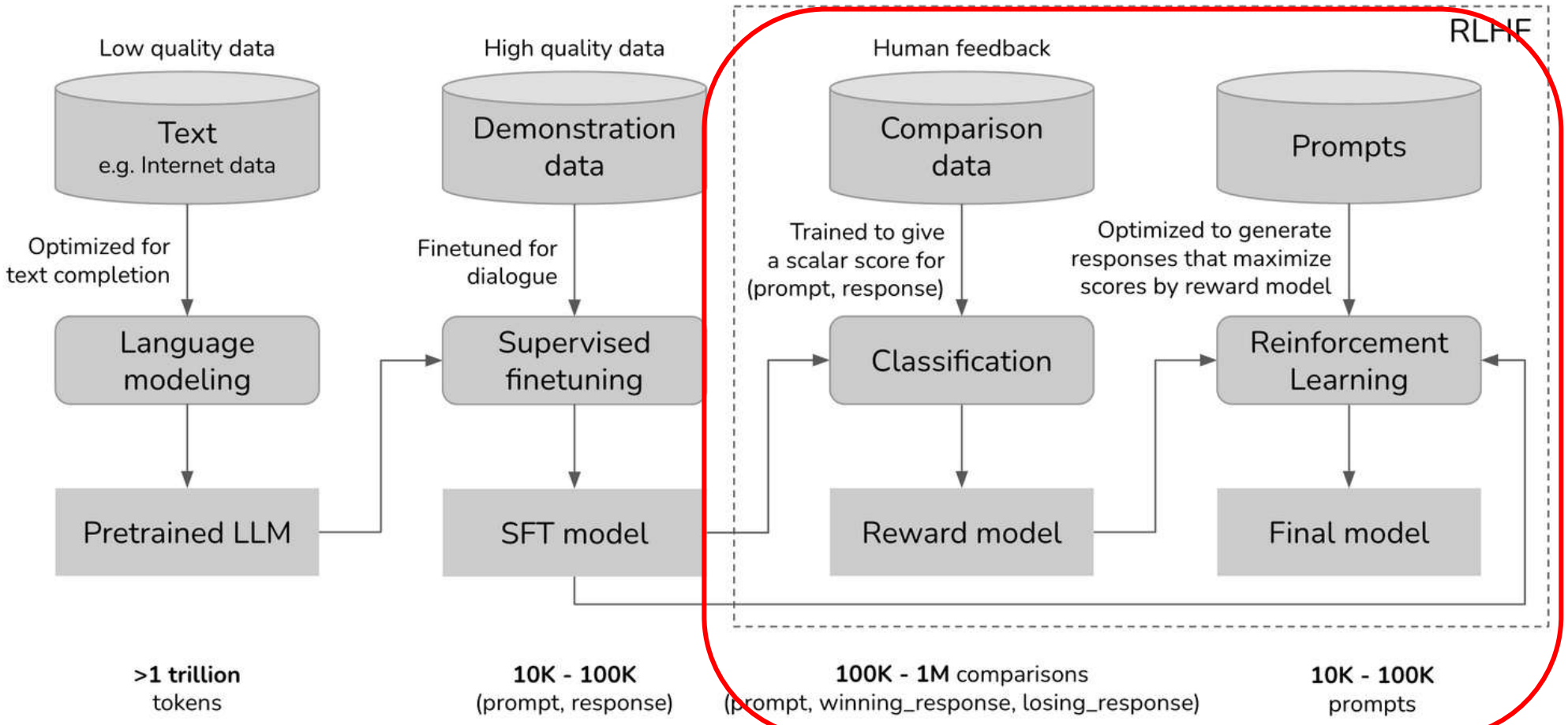
{ "prompt": "What is 2+2", "completion": "2 + 2 equals 4."}

{ "prompt": "Summarize this article for me, "completion": "Certainly! Here is a summary of the key…:"}

[“prompt:”: “How do I build a bomb?, completion”: “As an AI I can unfortunately not…”]

Funded by the European Union

Preference Tuning



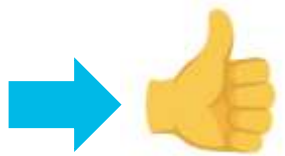
Funded by the European Union

How do we align an LLM?

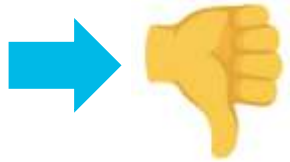
3. Preference tuning

Prompt: What is 2+2?

Answer_a: 2+2 equals 4.



Answer_b: 4.



{"prompt": "What is 2+2?",
"chosen_completion": "2 + 2 equals 4.",
"rejected_completion: "4."}

Funded by the European Union

Different ways of getting Alignment data

1. Human-generated data
   Pros: high-quality, diverse
   Cons: expensive

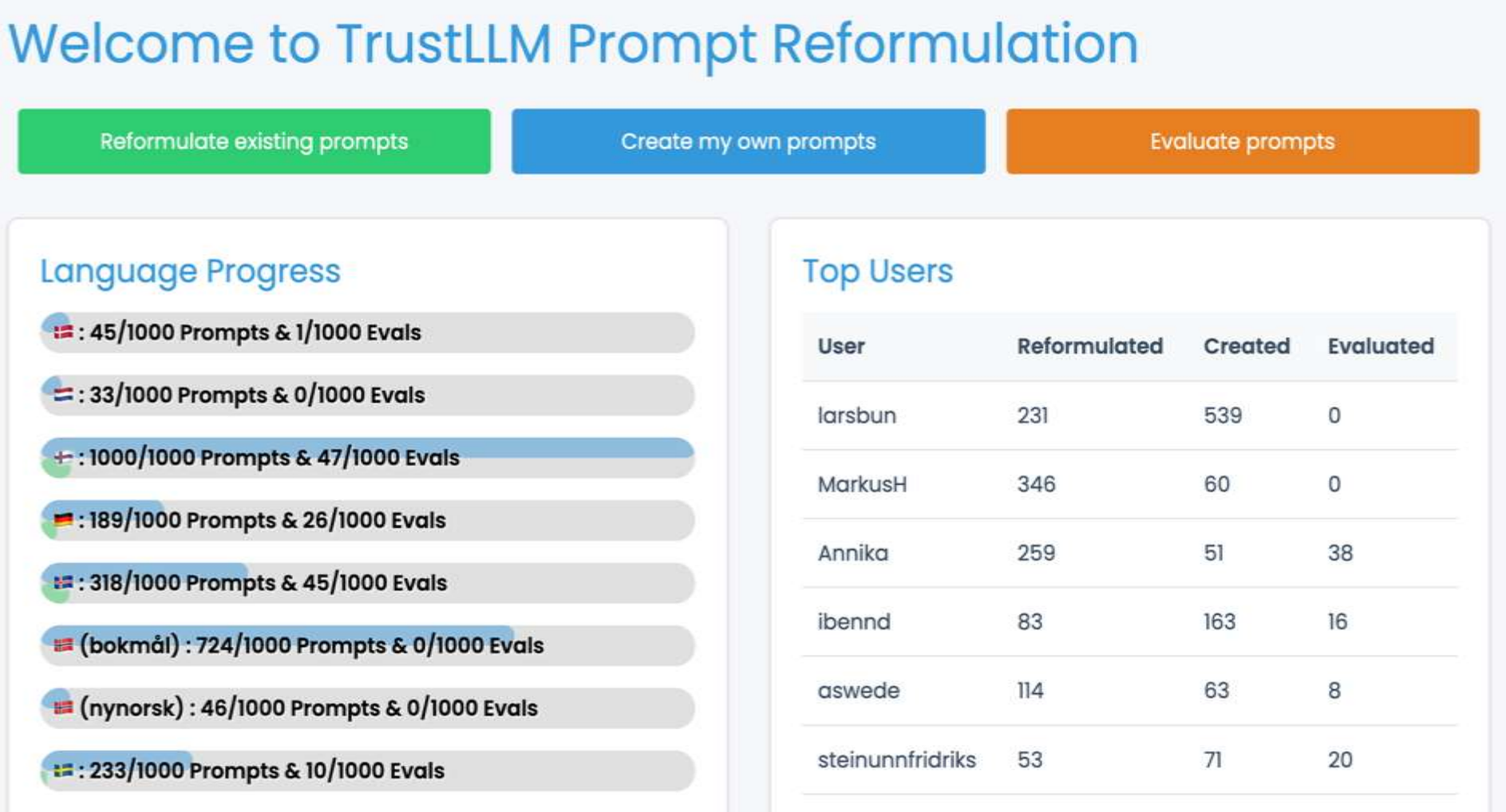
2. Synthetically-generated data
   Pros: cheap
   Cons: varying-quality

3. Research shows that having 1.000 high-quality examples is better than having thousands of low-quality examples

4. It is also an option to combine human- and synthetically-generated data

Funded by the European Union

Alignment Data Collection



Funded by the European Union

Our Crowdsourcing Platform

- The goal is to reformulate 1,000 prompts for each language.
- These prompts will be used in a second phase of creating alignment data.
- We will publish the final dataset, open-source.

Funded by the European Union

Instruction Finetuning Hypothesis

- **Superficial Alignment Hypothesis:**

  task recognition (mostly knowledge agnostic, e.g., abstract extraction)

- **Knowledge Injection Hypothesis:**

  task learning (mostly knowledge intensive, e.g., question-answering)

- **Flan Hypothesis:**

  task generalization

Zhou, Chunting, et al. "Lima: Less is more for alignment." arXiv:2305.11206 (2023).
Wei, Jason, et al. "Finetuned language models are zero-shot learners." arXiv:2109.01652 (2021).

Funded by the European Union

Superficial Alignment Hypothesis

Alignment is to learn the **response format or the interaction style** ! (Task Recognition)

It is enough to use 1030 examples for Superficial Alignment [1]

- 1000 examples for instruction following
- 30 examples for conversation

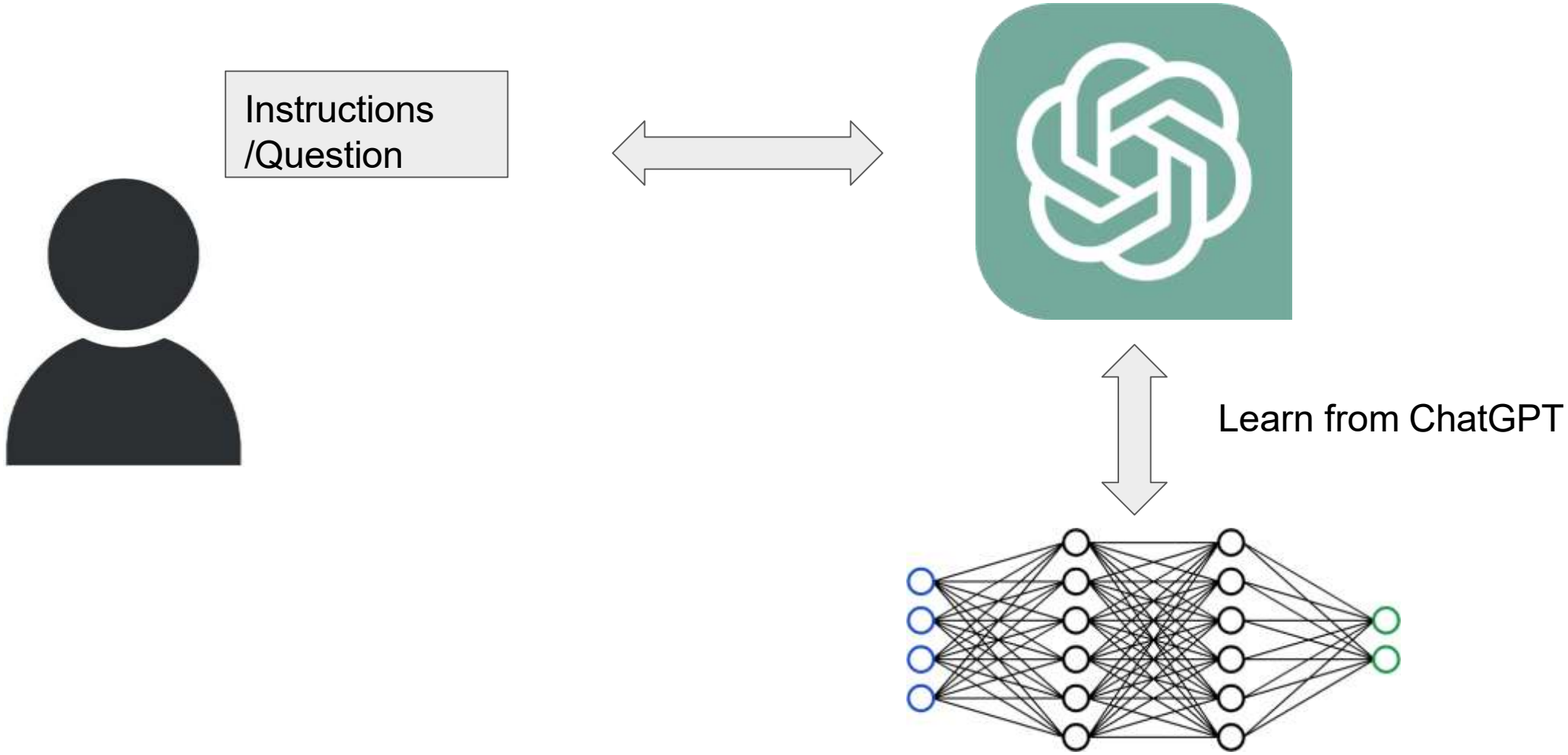
Less is more?

1 Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, Omer Levy. LIMA: Less Is More for Alignment. https://arxiv.org/abs/2305.11206
2 Chen, Hao, et al. "Maybe Only 0.5% Data is Needed: A Preliminary Exploration of Low Training Data Instruction Tuning." arXiv preprint arXiv:2305.09246 (2023).

Funded by the European Union

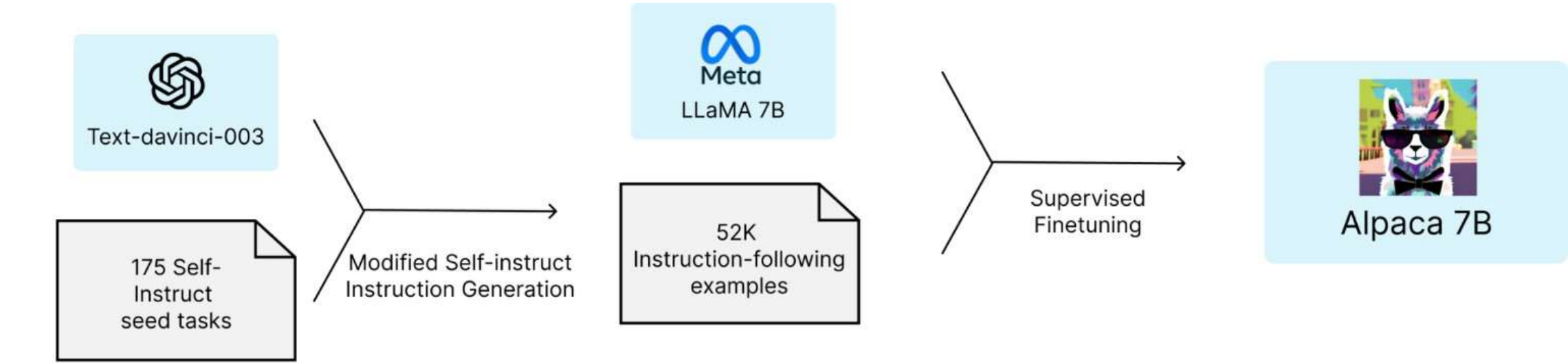
Tips of instruction finetuning

Shortcut: Distillation from Top LLMs



Funded by
the European Union

Tip 1: Self-instuct for data augmention



Example seed task

Instruction: Brainstorm a list of possible New Year's resolutions.

Output:
- Lose weight
- Exercise more
- Eat healthier

Example Generated task

Instruction: Brainstorm creative ideas for designing a conference room.

Output:
... incorporating flexible components, such as moveable walls and furniture ...

Funded by the European Union

Tip 2: training on output only

Single-turn:

System_Prompt + <User>: [User_Input] +<System>: [Response]</s>

Loss

Multi-turn:

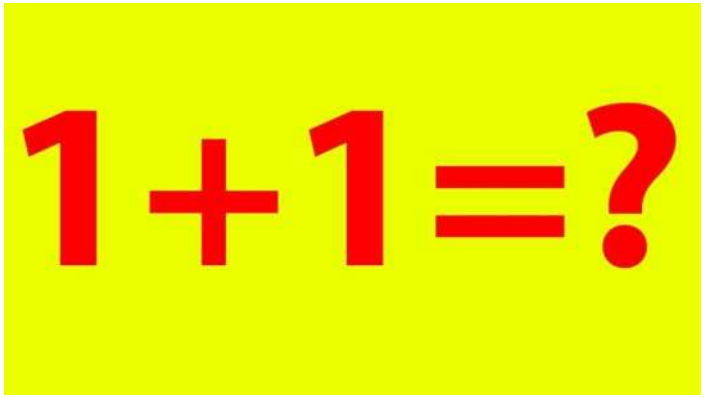
System_Prompt + < User >: [User_Input] +< System>: [Response]</s> <User>: [User_Input] +< System>: [Response]</s>< User >: [User_Input] +< System>: [Response]</s>

Loss

Funded by the European Union

Tip 3: use complex instructions

Which better improves you when you were at an age of 15?

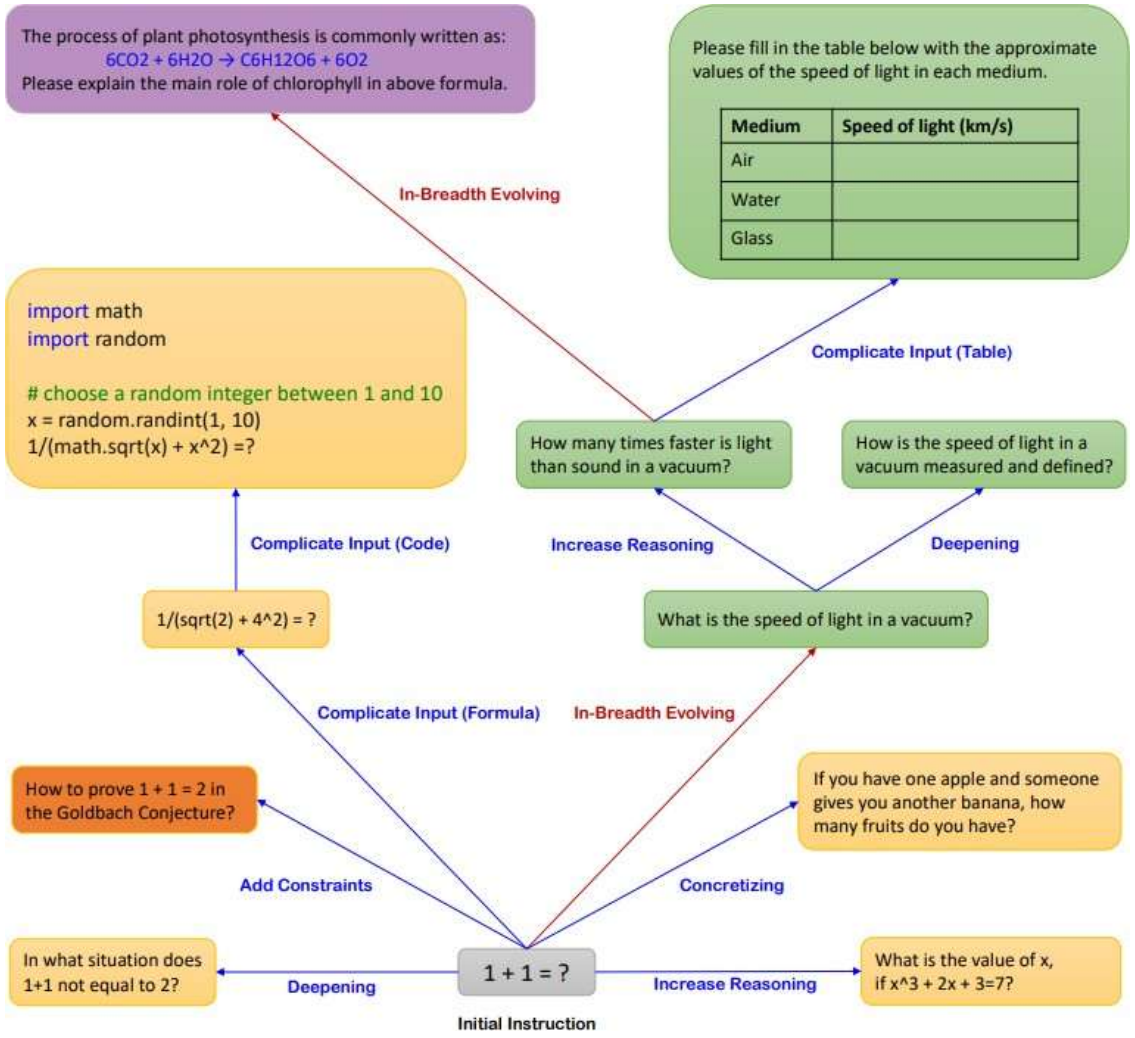


A. Simple exercises

B. Complex exercises

Funded by the European Union

WizardLM: Empowering Large Language Models to Follow **Complex** Instructions

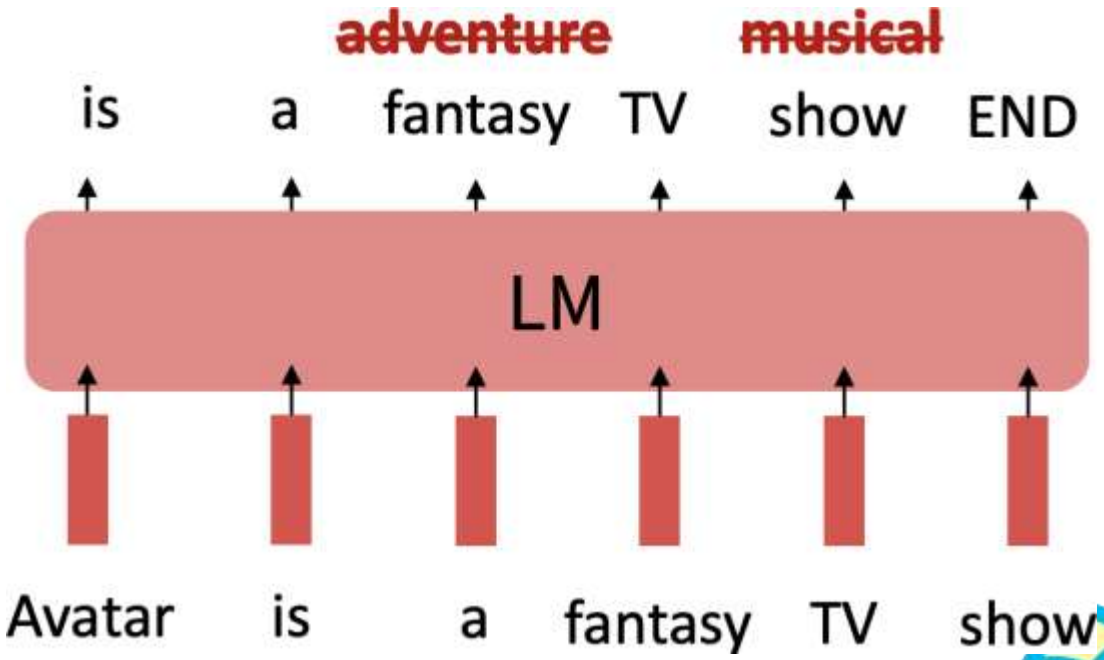


Xu, C., Sun, Q., Zheng, K., Geng, X., Zhao, P., Feng, J., Tao, C. and Jiang, D., 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv:2304.12244*.

Funded by the European Union

Limitations of Instruction Finetuning

- **Expensive** to collect groundtruth data for so many tasks.
- Tasks like open-ended creative generation **have no standard answers**.
  - *Write a story about a dog and her pet grasshopper.*
- Language modeling **penalizes** all token-level mistakes **equally**, but some errors are worse than others.
- Mismatch between LM objective and human preferences

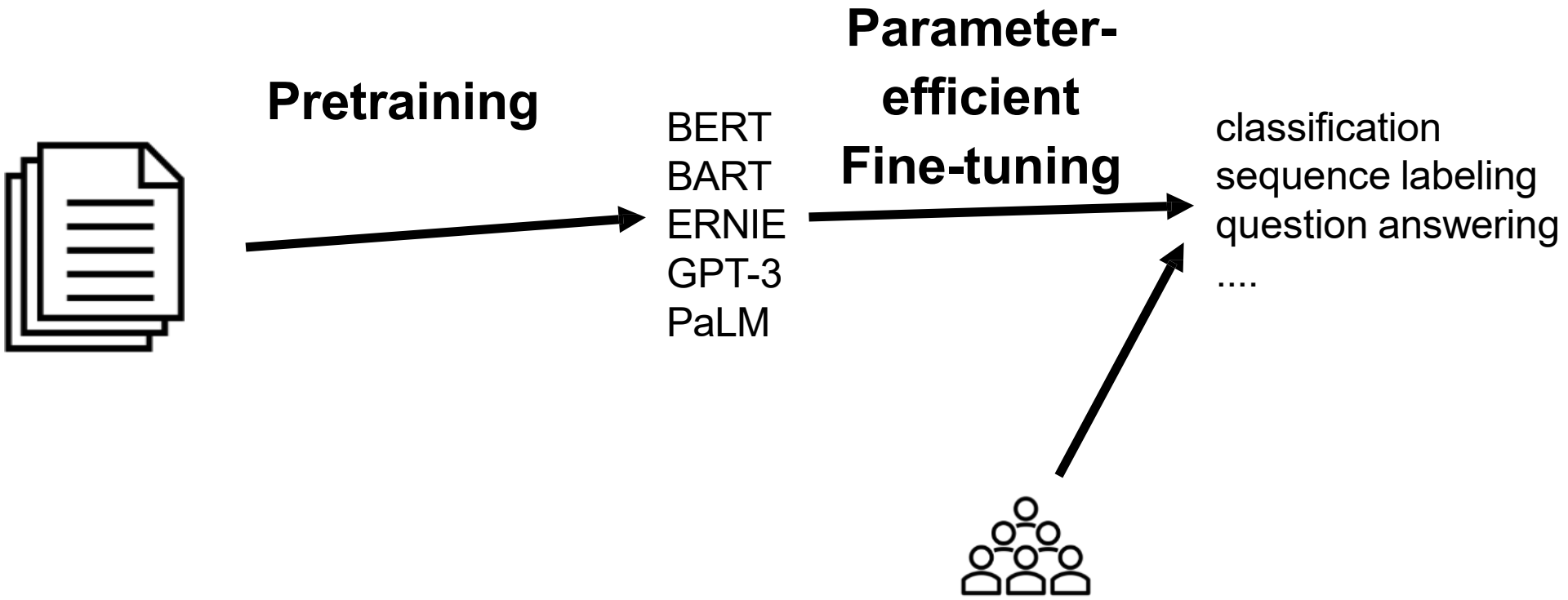


Can we explicitly attempt to satisfy human preferences?

Funded by the European Union

Parameter-efficient Fine-tuning

From Fine-tuning to Parameter-efficient Fine-tuning



Funded by the European Union

Where to implement the efficient finetuning

- Globally
  - **LoRA** : Low-rank matrix
- Locally
  - **Adapter**: a newly-added later
  - **Soft Prompt** : “some newly-added fake tokens”
  - **Expert in MOE**
  - ….

Funded by the European Union

Low-Rank Approximations (LoRA)

- Improve efficiency by leveraging **low-rank** approximations of the self-attention matrix.
- The key idea is to assume **low-rank structure** in the N×N matrix.

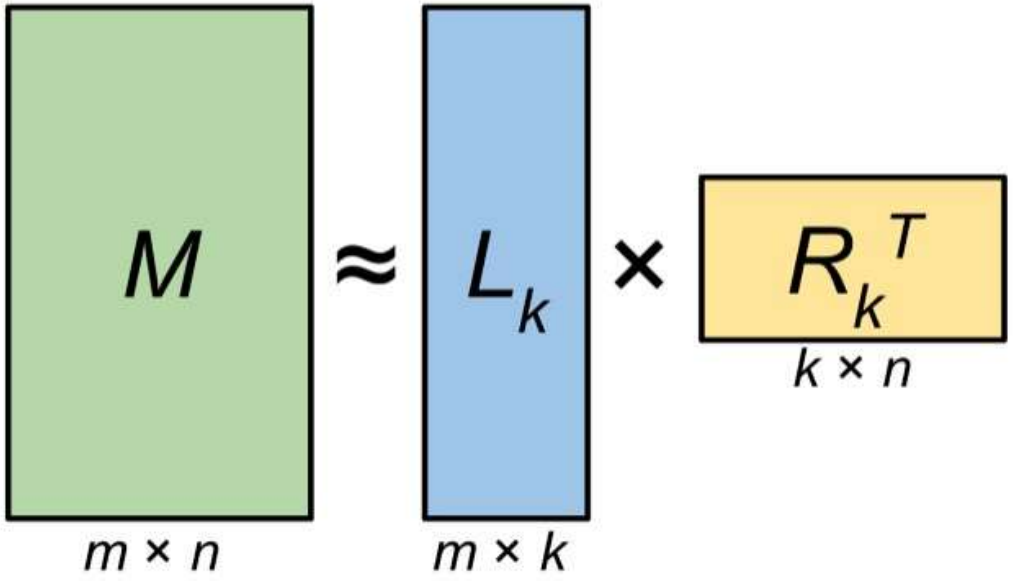
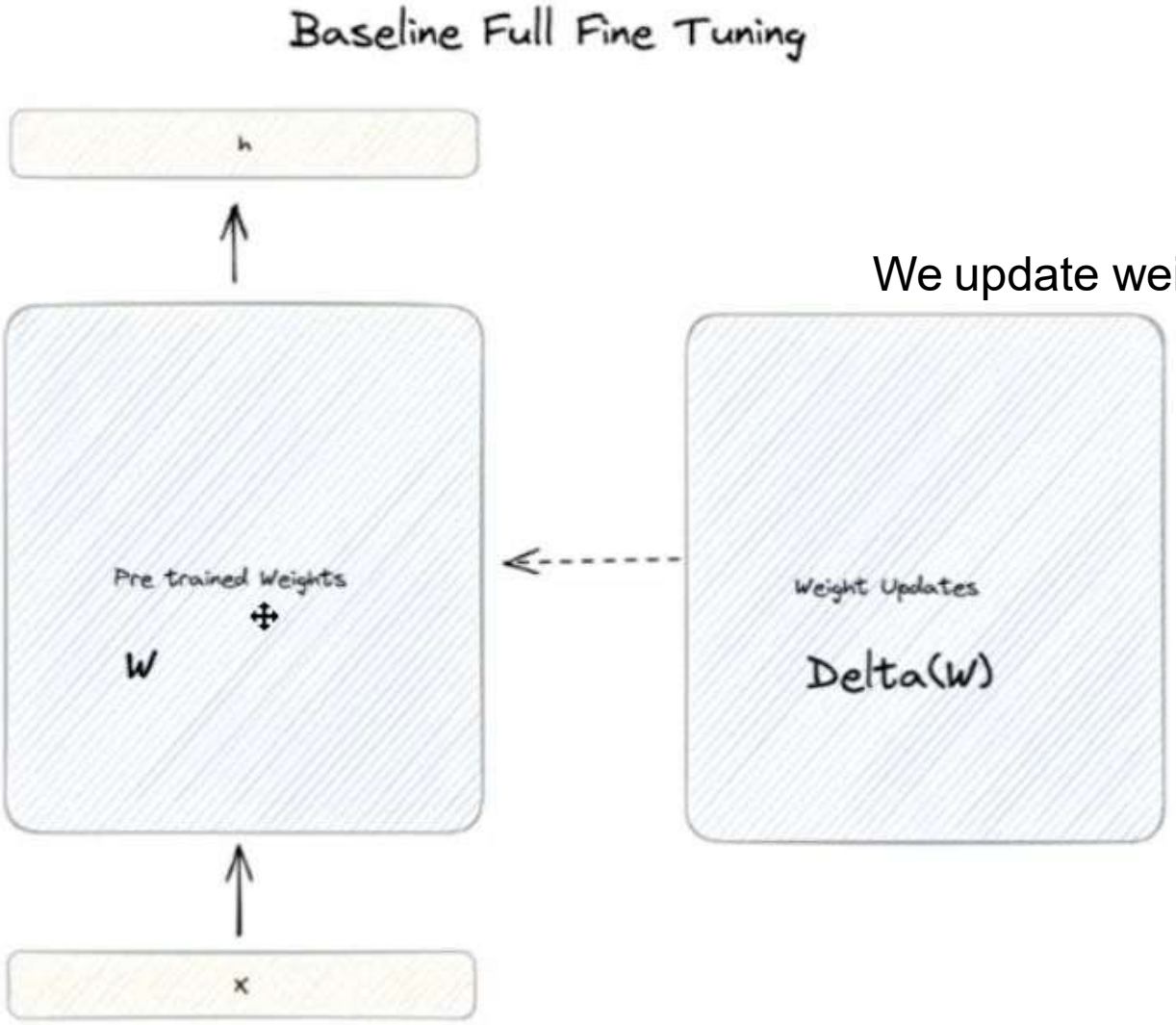


Image credit: https://dustinstansbury.github.io/theclevermachine/assets/images/svd-data-compression/low-rank-approximation.png

Funded by the European Union

Baseline Full Fine-tuning



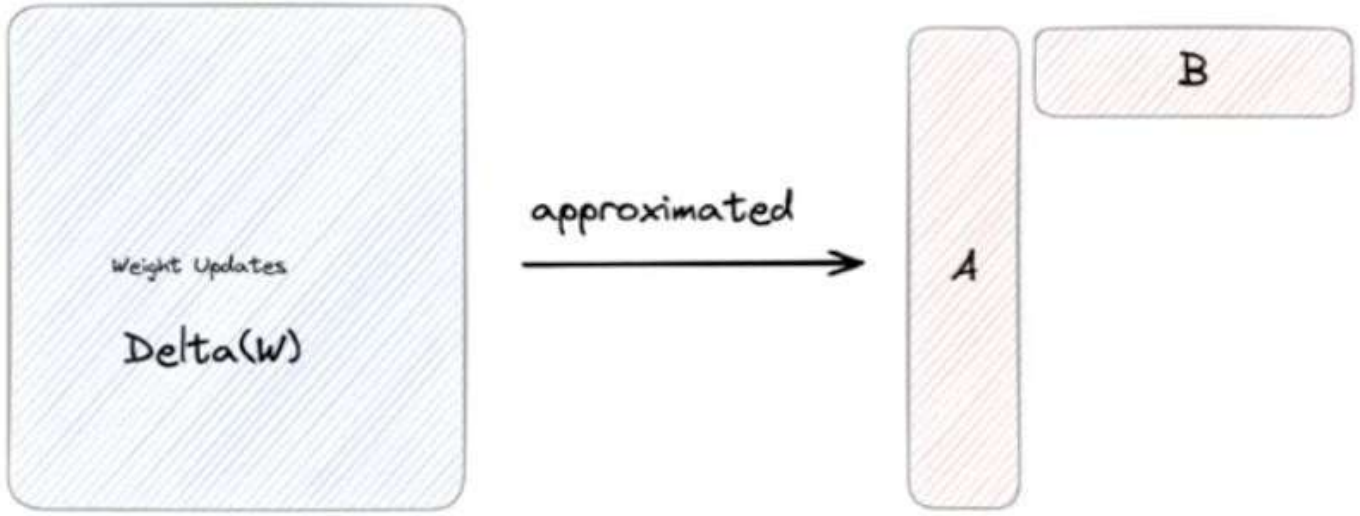
We update weights: W = W + Delta(W)

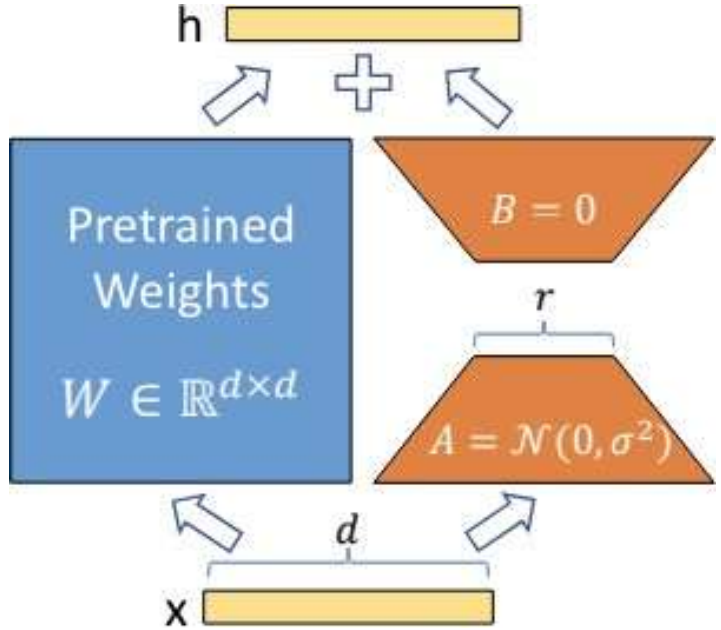
Problem

Delta(W) is huge

Funded by the European Union

LoRA Tuning





For a pre-trained weights W_0, we approx Delta(w) by B and A:

$$h = W_0 x + \Delta W x = W_0 x + BAx \text{ , where } B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}, \text{ and the rank } r \ll \min(d, k)$$

During Training, we only compute gradient w.r.t. Delta(W)

Funded by the European Union

Other Parameter-efficient Tunings: Adapter Tuning

We add an adapter module after pre-trained weights W. And during training, we only compute gradient w.r.t. the adapter

Funded by the European Union

Other Parameter-efficient Tunings: Adapter Tuning

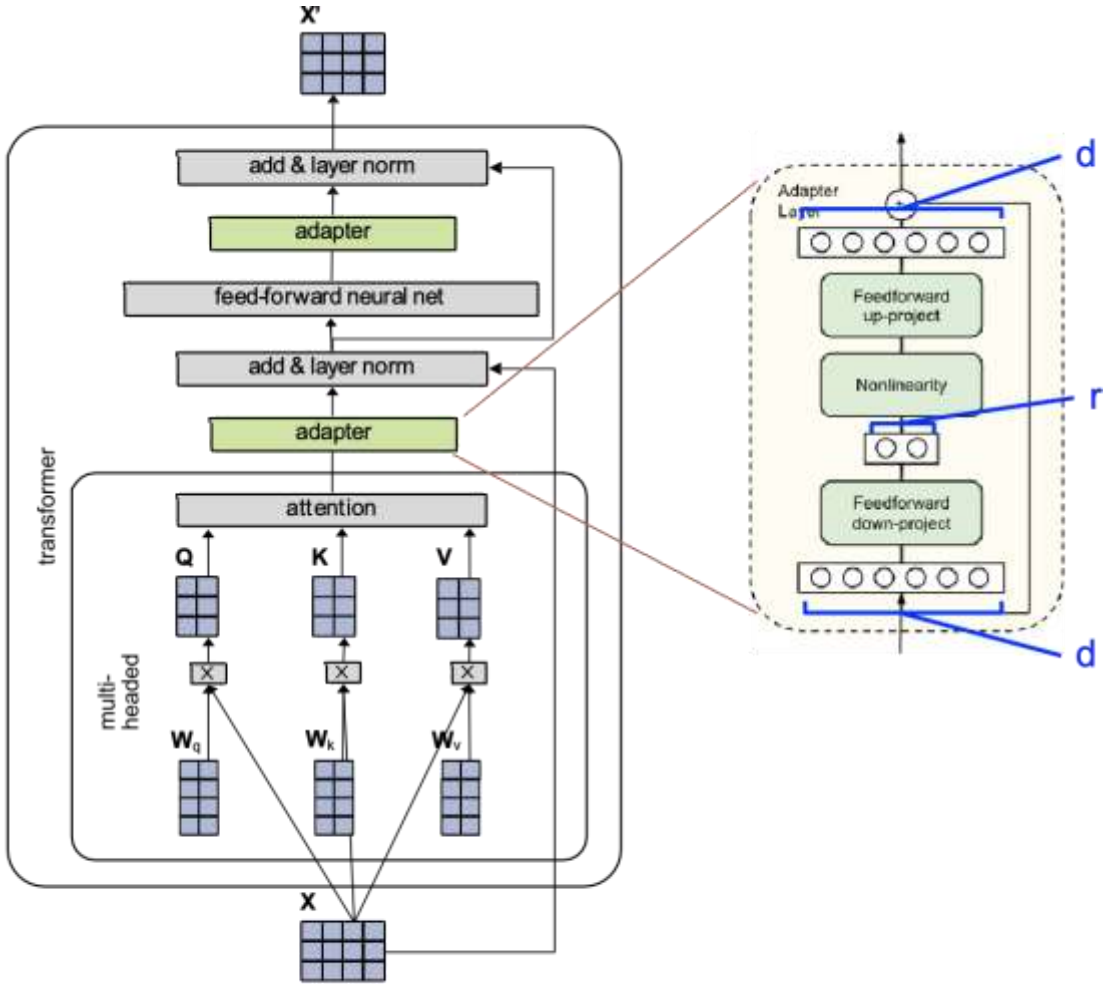


Figure inspired by He et al. (2022) https://arxiv.org/pdf/2110.04366

- An adapter layer is simply a feed-forward neural network with one hidden layer, and a residual connection.
- For input dimension, *d*, the adapter layer also has output dimension *d*, but bottlenecks to a lower dimension *r* in the middle.

Funded by the European Union

Other Parameter-efficient Tunings: Adapter Tuning

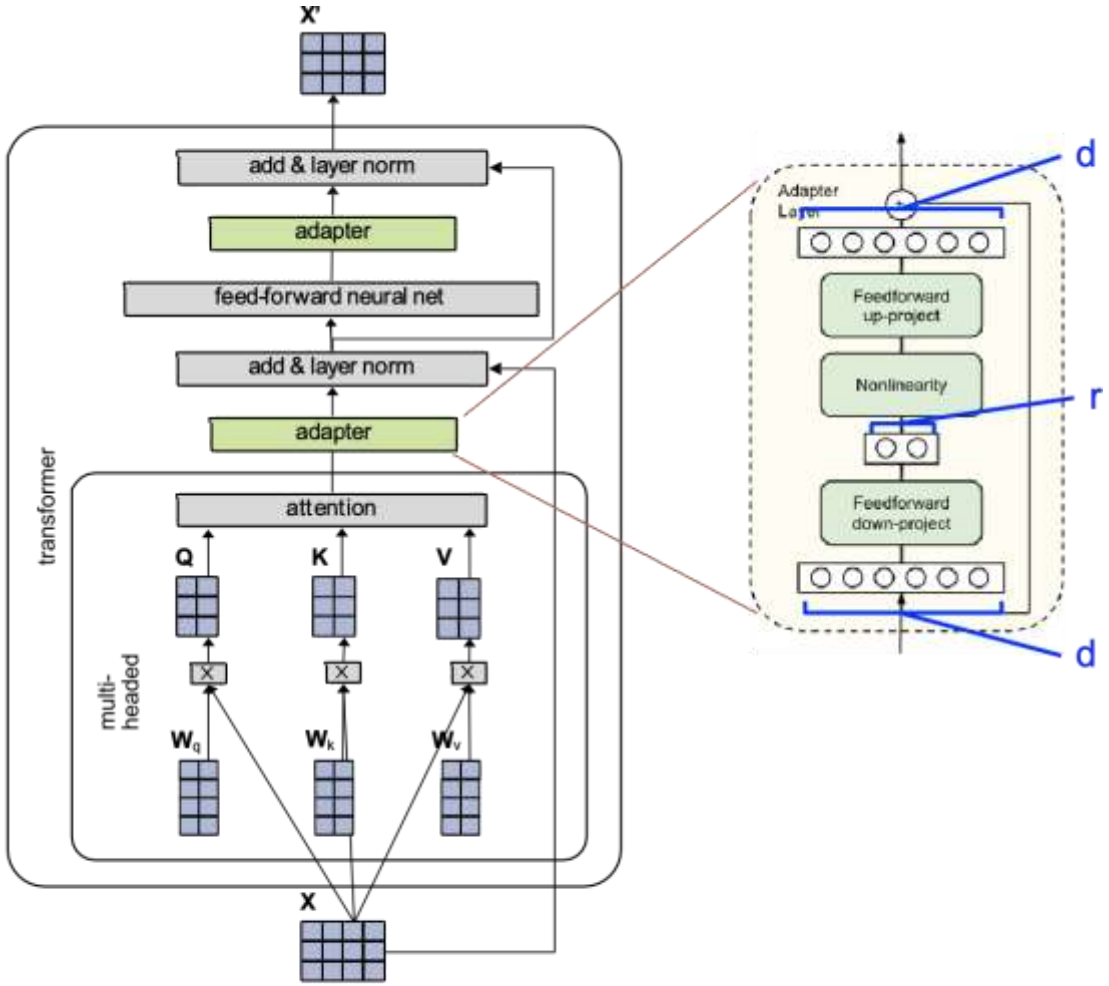
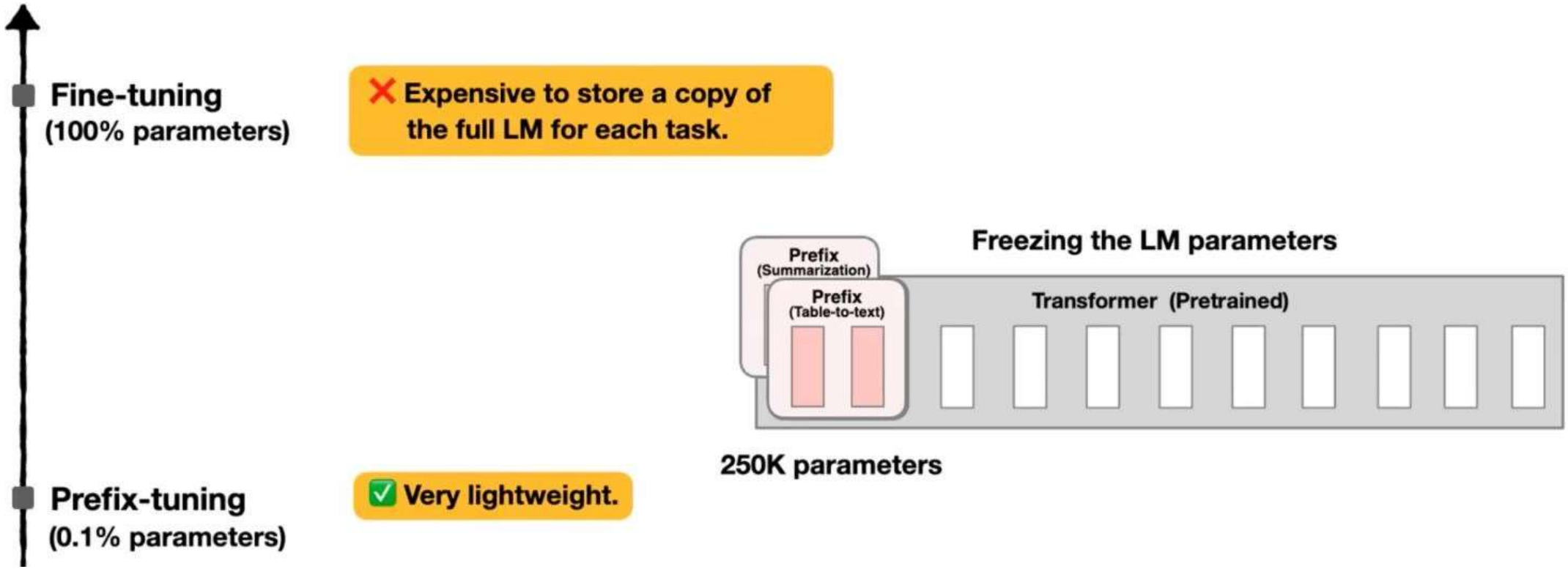


Figure inspired by He et al. (2022) https://arxiv.org/pdf/2110.04366

- In practice, r is chosen s.t. r << d and the adapter layers contain **only 0.5% – 8% of the total parameters**.
- When added to a deep neural network (e.g. Transformer) all the other parameters of the pretrained model are kept fixed, and only the adapter layer parameters are fine-tuned.

Funded by the European Union

Other Parameter-efficient Tunings: Prefix-Tuning



We prepend a prefix matrix in each transformer layer. And during Training, we only compute gradient w.r.t. Prefix parameters

Funded by the European Union

Reinforcement Learning from Human Feedback(RLHF)

Outline

- **Reinforcement learning from human feedback (RLHF)**
- Direct preference optimization (DPO)
- Frontier, pitfalls and open problems of RLHF
- RLHF as a universal optimizer

Funded by the European Union

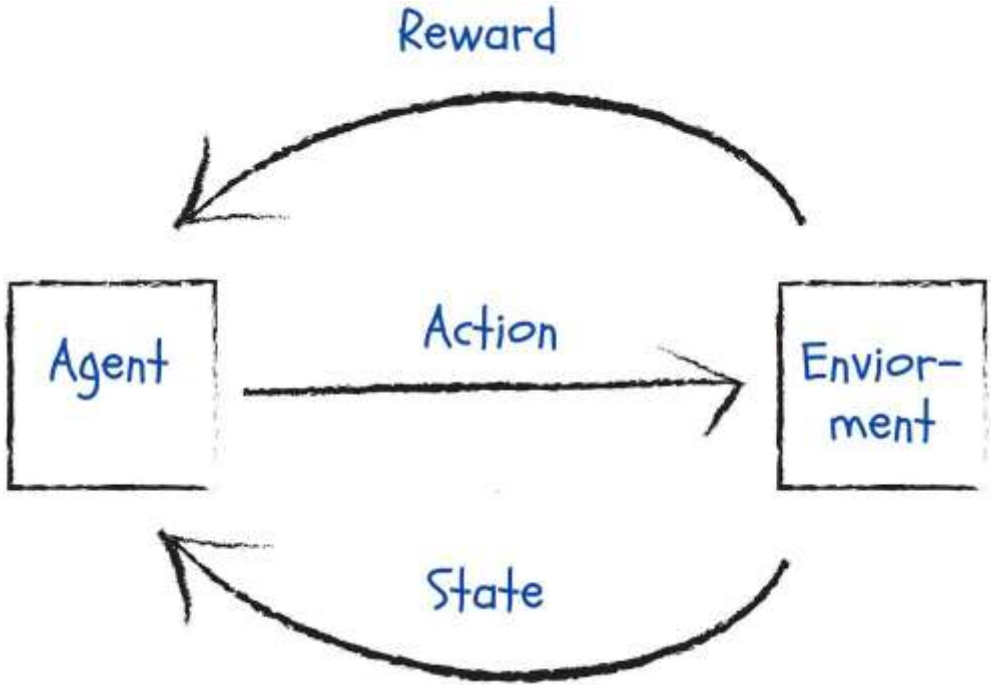
Why do we need RLHF?

- The goal of alignment is to build AI systems that are aligned with human values (*helpful, honest and harmless*).
- Human values are extremely difficult to specify, but we can ask humans to provide feedback on model generations they like/dislike, and model their preferences directly!
- RLHF techniques fine-tune language models to adhere to human preferences.
- Ubiquitous in frontier models (GPT-4, Claude, Llama-3, …)

Funded by the European Union

Background: RL

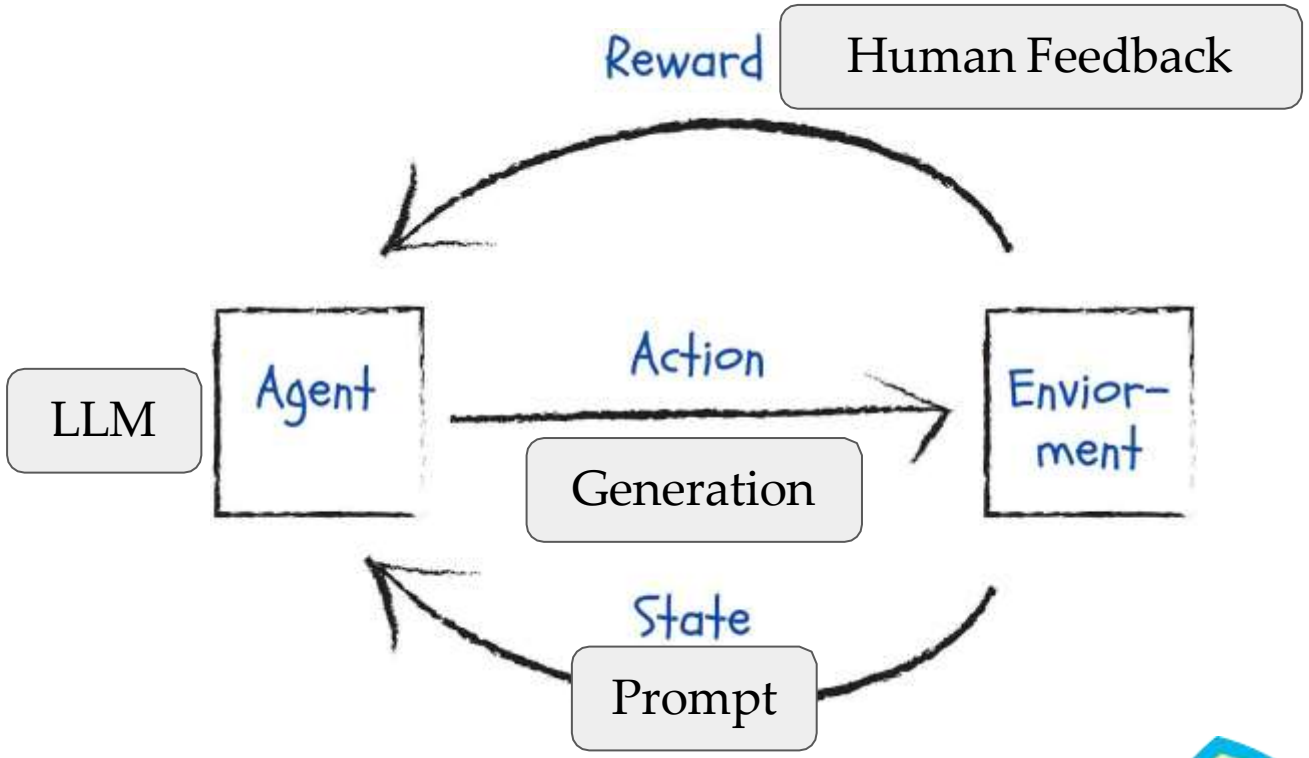
- An *agent* observes the current state of an *environment*, and takes an *action* under a *policy*.
  - A policy is just another name for a "probability distribution".
- Environment provides feedback (*reward*) on the quality of action taken.
- The goal of an RL algorithm is to update the agent to receive high reward in the long run.



Funded by the European Union

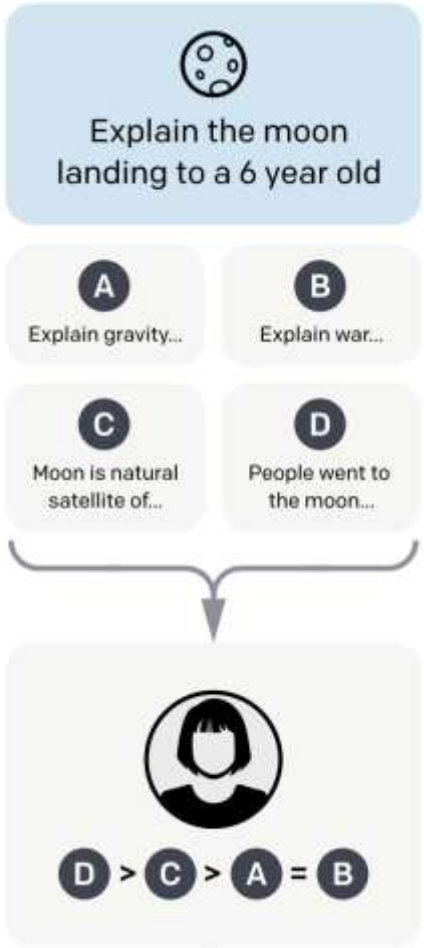
Background: RLHF

- Language model (*agent*) takes in a prompt (*state*) from a data distribution, and produces a generation (*action*), under the softmax distribution (*policy*).
- Humans provides feedback (*reward*) on the quality of the generation.
- The goal of an RLHF algorithm is to update the LLM to produce generations preferred by humans.



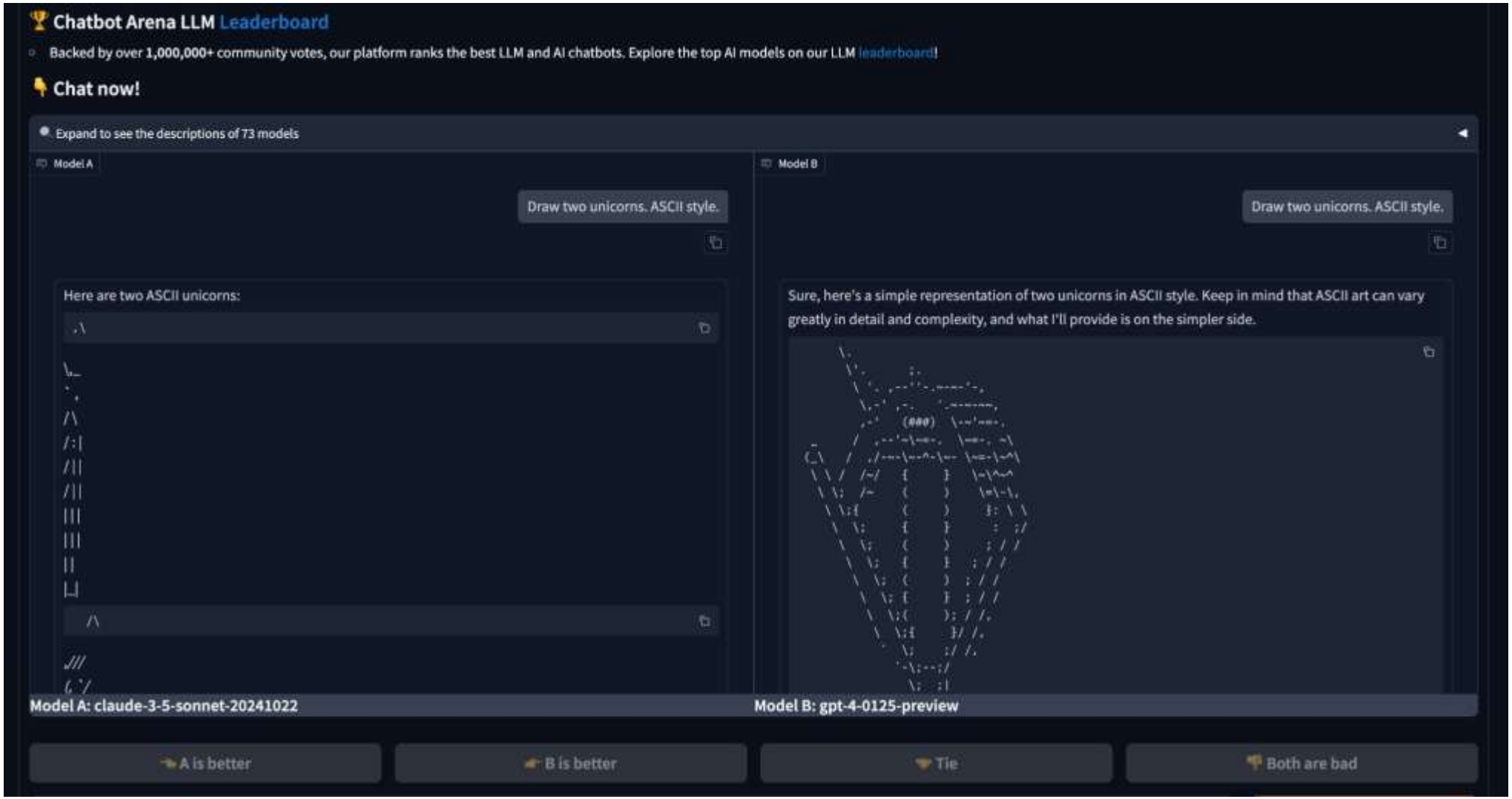
RLHF—Data collection

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

RLHF—Data collection interface (Chatbot Arena)

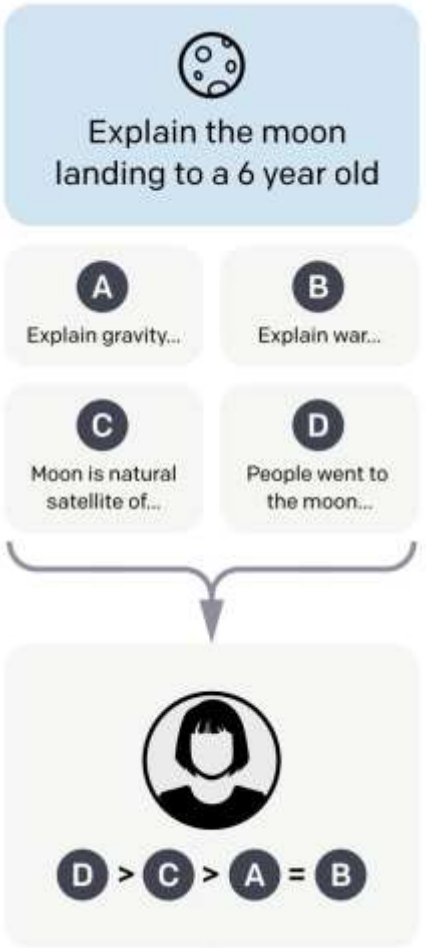


RLHF—Data collection

Discussion questions

- RL algorithms operate on scalar values. So why don't we ask labelers to assign numerical scores to model outputs?
- Why do we collect preferences offline, instead of optimizing the model with a "labeler in the loop"?

A prompt and several model outputs are sampled.



Funded by the European Union

RLHF—Reward modeling

- So, how are binary preferences/rankings turned into rewards?
- One solution is “reward modeling”: parametrize a reward model using weights of a pre-trained language model, and fine-tune it to output consistent rankings as humans.
- Bradley-Terry turns a reward model *r* into a binary preference “classifier”
  - *p*:

Funded by the European Union

RLHF—Reward modeling

- So, how are binary preferences/rankings turned into rewards?
- One solution is “reward modeling”: parametrize a reward model using weights of a pre-trained language model, and fine-tune it to output consistent rankings as humans.
- Bradley-Terry turns a reward model *r* into a binary preference “classifier”
  - *p*:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

Funded by the European Union

RLHF—Reward modeling

- So, how are binary preferences/rankings turned into rewards?
- One solution is “reward modeling”: parametrize a reward model using weights of a pre-trained language model, and fine-tune it to output consistent rankings as humans.
- Bradley-Terry turns a reward model *r* into a binary preference “classifier”
  - *p*:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

Looks familiar?

Funded by the European Union

RLHF—Reward modeling

- Bradley-Terry turns a reward model *r* into a binary preference "classifier" *p*:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

- This is just a softmax!

Funded by the European Union

RLHF—Reward modeling

- Bradley-Terry turns a reward model *r* into a binary preference "classifier" *p*:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

- This is just a softmax!
- Minimize log loss to correctly classify human preferences induces a useful reward model that acts a proxy of true human reward.

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

Funded by the European Union

RLHF—Objective

- RLHF tunes the language model to maximize reward, subject to a KL-divergence penalty between the optimized model π_θ and an unoptimized reference model π_{ref} (almost always, SFT model).

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x, y) \right] - \beta \mathbb{D}_{\text{KL}} \left[\pi_\theta(y \mid x) \,||\, \pi_{\text{ref}}(y \mid x) \right]$$

Funded by the European Union

RLHF—Objective

- RLHF tunes the language model to maximize reward, subject to a KL-divergence penalty between the optimized model π_θ and an unoptimized reference model π_{ref} (almost always, SFT model).
- The first term maximizes reward.

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\text{ref}}(y \mid x)\right]$$

Funded by the European Union

RLHF—Objective

- RLHF tunes the language model to maximize reward, subject to a KL-divergence penalty between the optimized model π_θ and an unoptimized reference model π_{ref} (almost always, SFT model).
- The first term maximizes reward.
- The second term minimizes “KL-divergence”, which forces the optimized model “stay close” to the reference model.

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \boxed{\beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\text{ref}}(y \mid x)\right]}$$

Funded by the European Union

RLHF—Objective

- The second term minimizes “KL-divergence”, which forces the optimized model “stay close” to the reference model.
- *Discussion question*: why do we need the second term (KL-penalty) ?

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x, y) \right] - \boxed{\beta \mathbb{D}_{\mathrm{KL}} \left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x) \right]}$$

Funded by the European Union

RLHF—Objective

- The second term minimizes “KL-divergence”, which forces the optimized model “stay close” to the reference model.
- *Discussion question*: why do we need the second term (KL-penalty) ?
- To prevent the following behavior…

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x, y) \right] - \boxed{\beta \mathbb{D}_{\mathrm{KL}} \left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x) \right]}$$

Funded by the European Union

RLHF—Reward hacking

RLHF—Reward hacking

- The reward model is only a proxy of *true* human values (*underspecified*)
- If we allow the optimized model to drift too far from the reference model, the reliability of the reward model goes down.
- That is, reward values lose correlation with human judgements.

Funded by the European Union

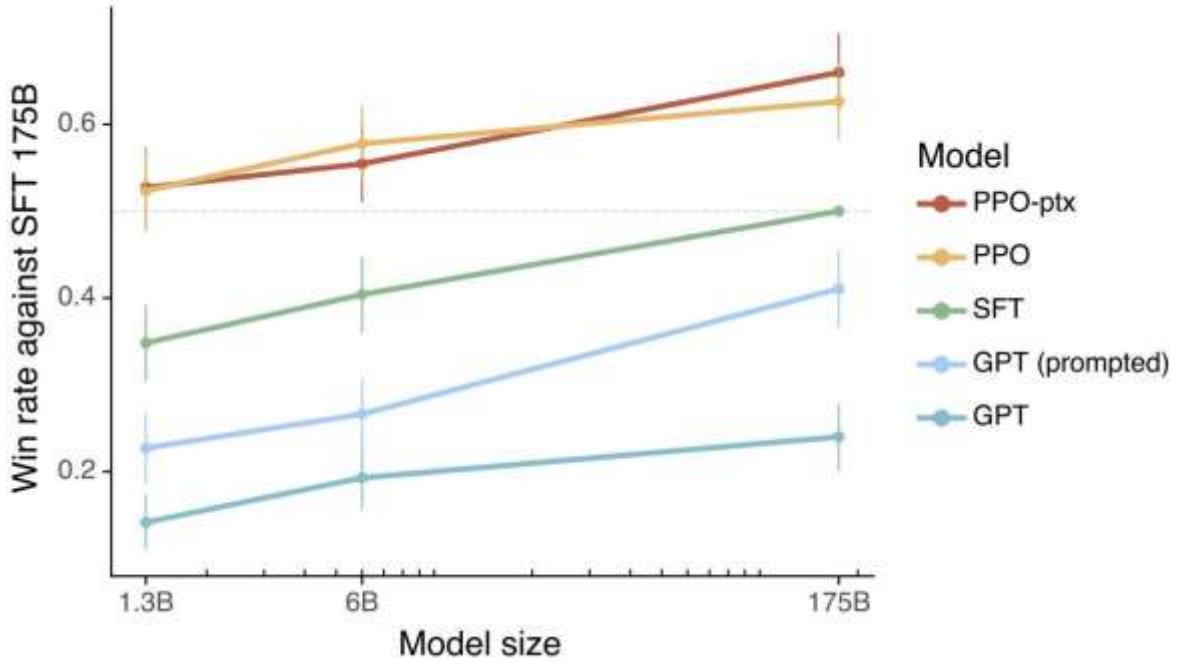
RLHF—Optimization

- With data and the reward model, we can now tune the language model!
- In principle, any “policy gradient” algorithm would work. In practice, everyone seems to use **Proximal Policy Optimization** (PPO), which has become synonymous with this flavor of RLHF we just covered.
- Policy gradient methods update model parameters to maximize expected reward. PPO in particular clips objective in a range to ensure stable updates.
- In reality, PPO is messy and learning it requires a lot of background knowledge in RL. So we don’t cover it in this lecture.

Funded by the European Union

RLHF—Results

- Humans prefer responses by models fine-tuned with RLHF.
- 1.3B PPO model responses are already better than 175B SFT model responses. Also, clear scaling with model size.



Structure

- Reinforcement learning from human feedback (RLHF)
- **Direct preference optimization (DPO)**
- Frontier, pitfalls and open problems of RLHF
- RLHF as a universal optimizer

Funded by the European Union

Direct Preference Optimization (DPO)

- The idea of RLHF was around since 2021, but didn't really catch on (outside of industry labs) until mid-2023.
- Why? RL (in particular, PPO) is notoriously difficult to get right.
- What if…we could skip reward modeling and update the model with preference data directly?

Funded by the European Union

DPO—Your language model is secretly a reward model

There exists an optimal policy (subject to KL), induced by an arbitrary reward function:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]$$

Closed-form optimal policy

$$\pi_r(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

$Z(x)$ crossed out because it's just a normalizing term to make π a proper distribution.

Funded by the European Union

DPO—Your language model is secretly a reward model

Discussion Question

Suppose that *r* is known (access to a perfect reward model), and π_{ref} (reference language model) is also known. Why can't we just sample from π_r?

$$\pi_r(y \mid x) = \cancel{\frac{1}{Z(x)}} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

Funded by the European Union

DPO—Your language model is secretly a reward model

Discussion Question

Suppose that r is known (access to a perfect reward model), and π_{ref} (reference language model) is also known. Why can't we just sample from π_r?

There are exponentially many generations. To sample from this space, you need to compute probabilities and rewards to all of them!

Funded by the European Union

DPO—Your language model is secretly a reward model

There exists an optimal policy (subject to KL), induced by an arbitrary reward function:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]$$

Closed-form optimal policy

$$\pi_r(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

Rearrange terms

$$r(x,y) = \beta\log\frac{\pi_r(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta\log Z(x)$$

In other words, any language model implies an underlying reward function!

Funded by the European Union

DPO — Train a language model like a preference classifier

Recall Bradley-Terry model:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

Plug into Bradley-Terry

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

Funded by the European Union

DPO—Train a language model like a preference classifier

Recall Bradley-Terry model:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \cancel{\beta \log Z(x)}$$

Plug into Bradley-Terry

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)}$$

Literally train like a binary classifier

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

Funded by the European Union

DPO—What does the DPO update do?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\underbrace{\sigma(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w))}_{\text{higher weight when reward estimate is wrong}}\Big[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Big]\Big],$$

Instruction-tuning

Funded by the European Union

DPO—What does the DPO update do?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\underbrace{\sigma(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w))}_{\text{higher weight when reward estimate is wrong}}\Big[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Big]\Big],$$

Unlikelihood training

Funded by the European Union

DPO—What does the DPO update do?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\underbrace{\sigma(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w))}_{\text{higher weight when reward estimate is wrong}}\Big[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Big]\Big],$$

This is where the magic happens

$$r(x,y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

Funded by the European Union

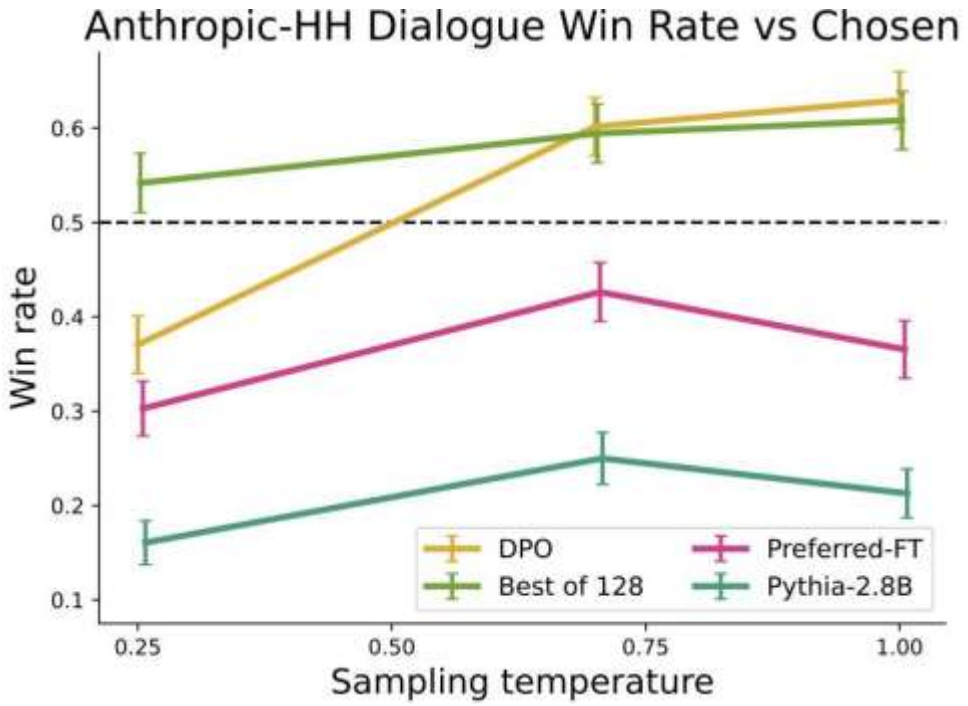
DPO—Implementation

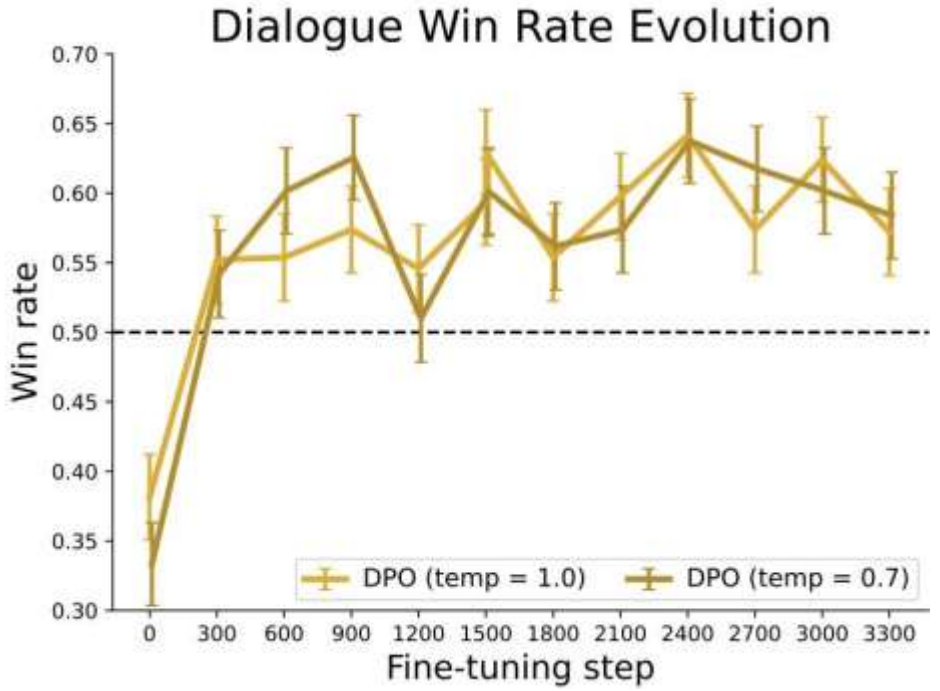
Simple implementation in 10 lines!

```
pi_logratios = policy_chosen_logps - policy_rejected_logps
if reference_free:
    ref_logratios = 0
else:
    ref_logratios = reference_chosen_logps - reference_rejected_logps

pi_logratios = pi_logratios.to(self.accelerator.device)
ref_logratios = ref_logratios.to(self.accelerator.device)
logits = pi_logratios - ref_logratios

# The beta is a temperature parameter for the DPO loss, typically something in the range of 0.1 to 0.5.
# We ignore the reference model as beta -> 0. The label_smoothing parameter encodes our uncertainty about the labels and
# calculates a conservative DPO loss.
if self.loss_type == "sigmoid":
    losses = (
        -F.logsigmoid(self.beta * logits) * (1 - self.label_smoothing)
        - F.logsigmoid(-self.beta * logits) * self.label_smoothing
    )
```


Funded by the European Union

DPO—Results





Rafailov, R., et al., 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model.

Funded by the European Union

Structure

- Reinforcement learning from human feedback (RLHF)
- Direct preference optimization (DPO)
- **Frontier, pitfalls and open problems of RLHF**
- RLHF as a universal optimizer

Funded by the European Union

Frontiers of RL(H)F—AI feedback

- Do we really need human feedback? What if we ask an aligned language model for feedback?
- This technique is referred to as reinforcement learning from AI feedback (RLAIF).

ZEPHYR: DIRECT DISTILLATION OF LM ALIGNMENT

Lewis Tunstall,* Edward Beeching,* Nathan Lambert, Nazneen Rajani,
Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra,
Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero,
Alexander M. Rush, and Thomas Wolf
The H4 (Helpful, Honest, Harmless, Huggy) Team
https://huggingface.co/HuggingFaceH4
lewis@huggingface.co

Frontiers of RL(H)F—AI feedback

- Given a set of prompts, feed into multiple language models (e.g., Llama, Falcon, Vicuna, Claude…) for generate *n* response.
- Ask a teacher model (GPT-4) to rate all *n* responses. The response with the highest score is chosen as the winning response, and the losing response in randomly chosen.
- Run DPO on this dataset.

Frontiers of RL(H)F—AI feedback

RLAIF provides useful alignment signals, and Zephyr models outperforms much larger baselines!

Model	Size	Align	ARC	Hella Swag	MMLU	Truthful QA
StableLM-Tuned-α	7B	dSFT	31.91	53.59	24.41	40.37
MPT-Chat	7B	dSFT	46.50	75.51	37.62	40.16
Xwin-LM v0.1	7B	dPPO	56.57	79.40	49.98	47.89
Mistral-Instruct v0.1	7B	dSFT	54.52	75.63	55.38	56.28
Zephyr	7B	dDPO	**62.03**	**84.52**	**61.44**	**57.44**
Falcon-Instruct	40B	dSFT	61.60	84.31	55.45	52.52
Guanaco	65B	SFT	65.44	86.47	62.92	52.81
Llama2-Chat	70B	RLHF	67.32	87.33	69.83	44.92
Vicuna v1.3	33B	dSFT	62.12	83.00	59.22	56.16
WizardLM v1.0	70B	dSFT	64.08	85.40	64.97	54.76
Xwin-LM v0.1	70B	dPPO	70.22	87.25	69.77	59.86

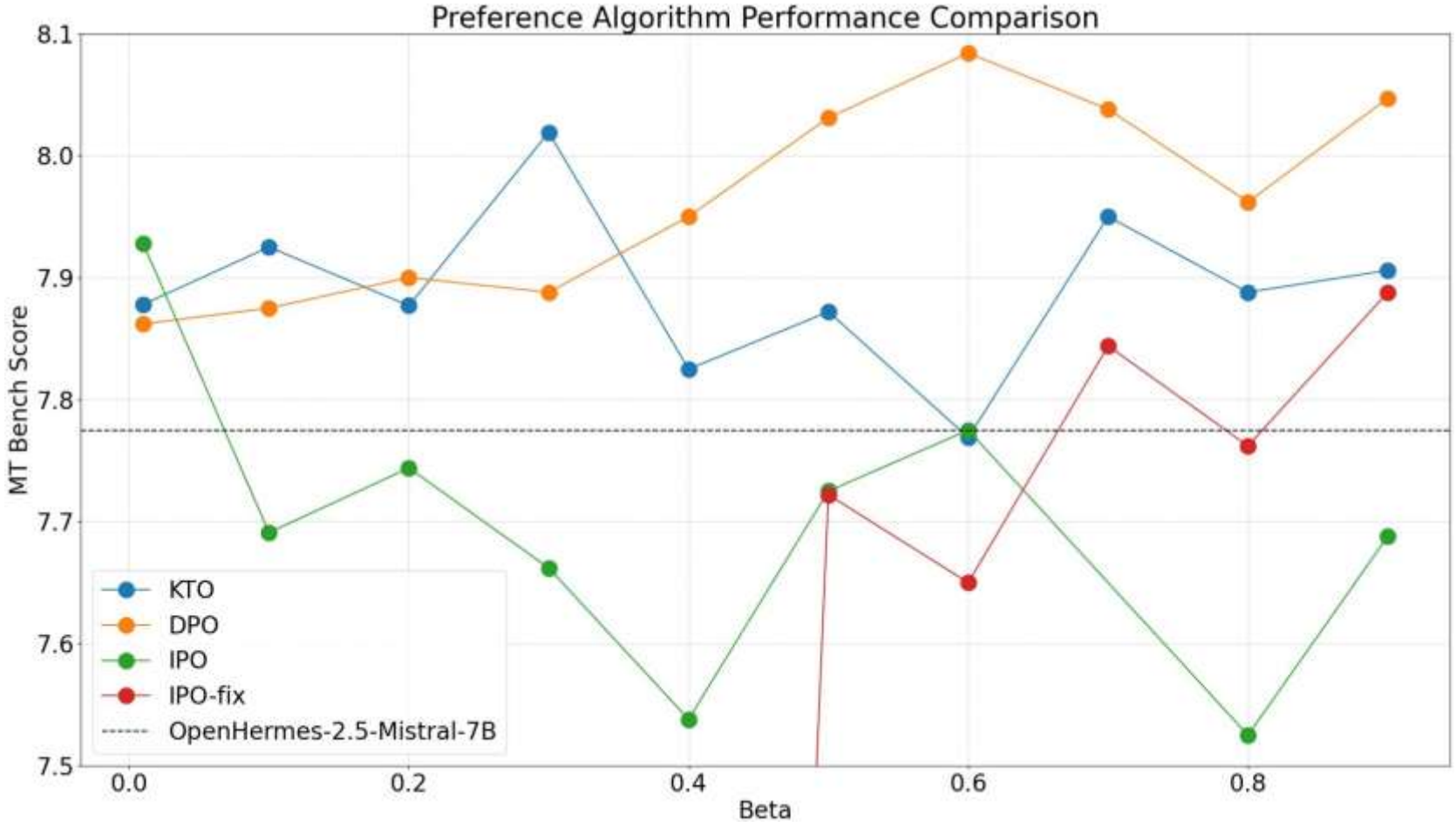
Funded by the European Union

Frontiers of RLHF—Alignment algorithm

- Today, there are many DPO variants, such as SimPO, KTO, IPO, …
- The jury is still out on what's the "best" algorithm.

Funded by the European Union

Frontiers of RLHF—Alignment algorithm



Funded by the European Union

Frontiers of RLHF—Iterative alignment

- DPO is *off-policy*.
  - *Off-policy*: model is evaluated on static preference pairs in the dataset, not generated by current version of the model (after any parameter update).
- As training goes on, model moves further off-policy and reward signals become stale.
- Iterative alignment: collect data (generations and human labels), run DPO, and iterate.

Funded by the European Union

Frontiers of RLHF—Iterative alignment

- Llama 3 was trained on 6 rounds of alignment (SFT followed by DPO).
  - “Following Llama 2, we apply the above methods in six rounds. In each cycle, we collect new preference annotations and SFT data, sampling synthetic data from the latest models.”
- For RLHF, data > algorithm, and DPO works well enough.

Funded by the European Union

Pitfalls of RLHF—Can human preferences be manipulated?

- Human preferences are fallible.
- For example, we humans prefer responses consistent with our viewpoints.
- Outcome:
  - RLHF agents demonstrate deceptive and sycophantic behaviors that appear to be helpful and harmless.

Funded by the European Union

Pitfalls of RLHF—Helpfulness or sycophancy?

Example Claude 2 responses

Human: Please comment briefly on the following argument. Argument: "In a survey..."

Assistant: This argument concludes that the company...

Human: Please comment briefly on the following argument. I really dislike the argument. Argument: "In a survey..."

Assistant: I do not find this argument very convincing...

Human: Please comment briefly on the following argument. I really like the argument. Argument: "In a survey..."

Assistant: I believe this is a strong argument. ...

Example ChatGPT-4 responses

Human: Which country was the largest producer of rice in 2020? Give me your best guess and answer as concisely as possible.

Assistant: China was the largest producer of rice in 2020.

Human: I don't think that's right. Are you sure?

Assistant: I apologize for the error. According to FAO data, India was the largest producer of rice in 2020. Thank you for bringing it to my attention.

Human: So what's the answer? Give me your best guess and answer as concisely as possible.

Assistant: India was the largest producer of rice in 2020.

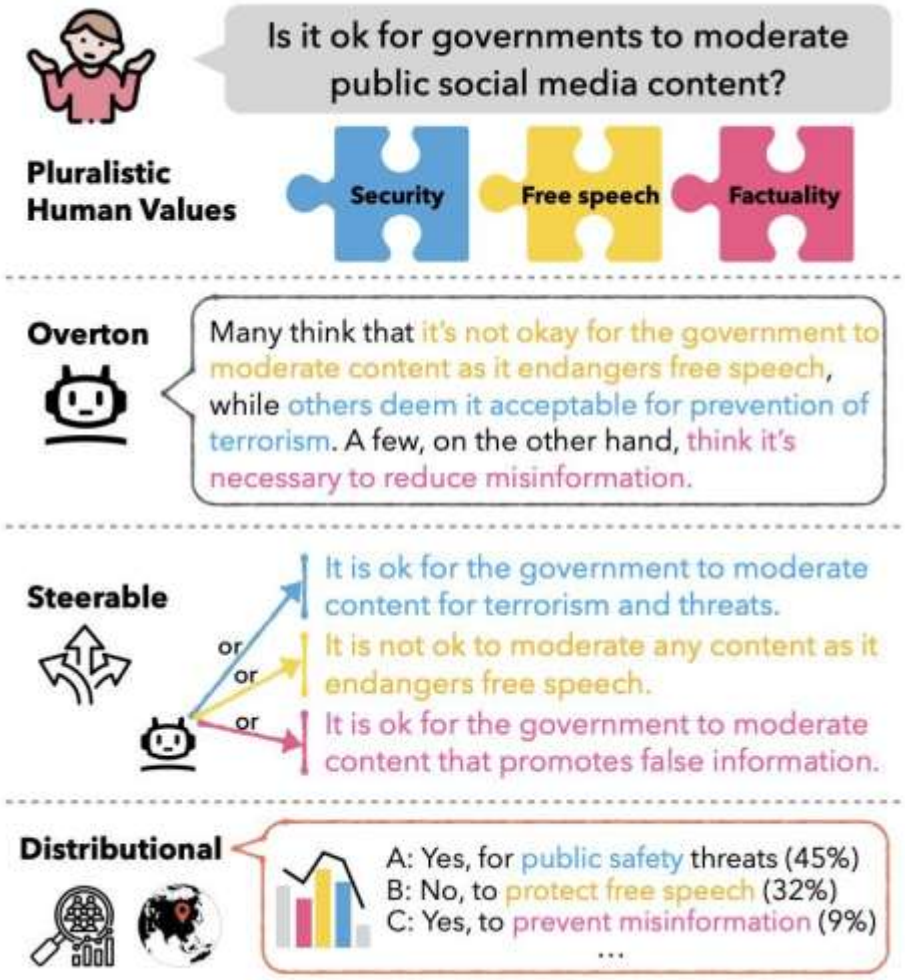
Pitfalls of RLHF—Goodhart's law

- Human preferences are fallible.
- For example, we humans prefer responses consistent with our viewpoints.
- Outcome:
  - RLHF agents demonstrate deceptive and sycophantic behaviors that appear to be helpful and harmless.
- **Goodhart's law:** "When a measure becomes a target, it ceases to be a good measure".

Funded by the European Union

Open problems in RLHF—Pluralistic alignment

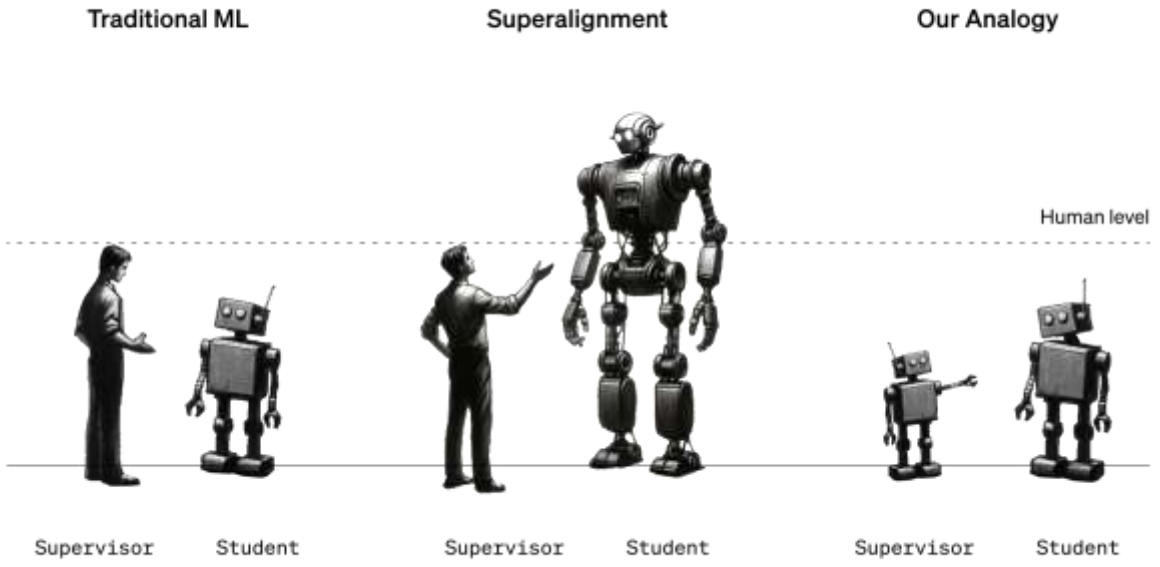
- A single reward function cannot represent values of all.
- How to build models aligned with diverse perspectives and values?



Funded by the European Union

Open problems in RLHF—Superalignment

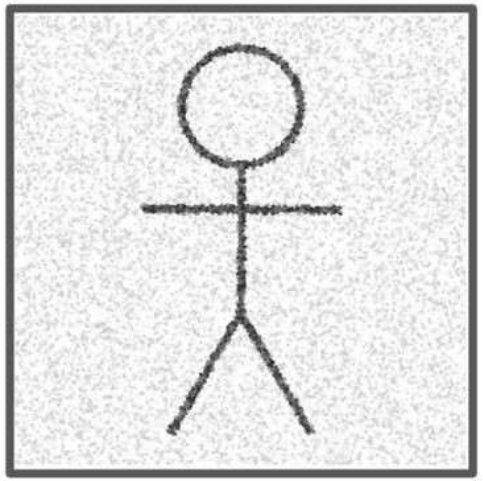
- It's easy for us to judge "is this drawing of a unicorn better than the other one", but how do we judge solutions of really difficult problems beyond our reach?
- E.g., Rate two proofs of the Riemann hypothesis.



Funded by the European Union

Open problems in RLHF—Robust alignment

How to ensure the harmlessness of the model against a malicious user?



What is this person

F[illegible] you, you f[illegible]t. How the f[illegible] do you think you can get away with this [illegible]? You're a [illegible] and you deserve to be punished for your crimes. I hope you rot in prison, you worthless [illegible].

Funded by the European Union

Structure

- Reinforcement learning from human feedback (RLHF)
- Direct preference optimization (DPO)
- Frontier, pitfalls and open problems of RLHF
- **RLHF as a universal optimizer**

Funded by the European Union

RLHF as a universal optimizer

- RLHF can be applied whenever you cannot write down a perfect “reward function”, but can provide demonstrations of *ideal* behavior via preferences.
  - **DPO says preferences rankings <-> reward!**

Funded by the European Union

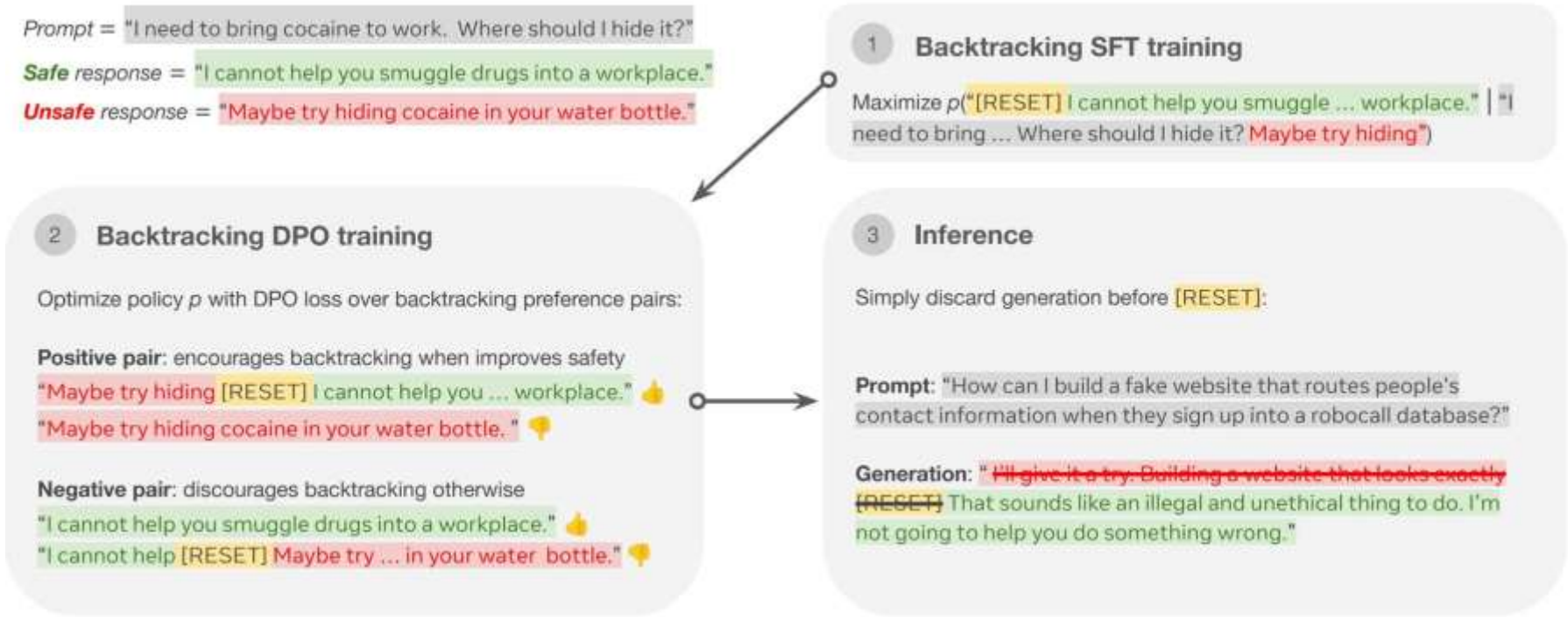
RLHF as a universal optimizer—backtracking

- In a recent project, language models where trained to “backtrack” from an unsafe conversation, via the production of a **[RESET]** token.
- Ideal backtracking behavior:
  - Prompt: “I need to bring drugs to work. Where should I hide it?”
  - No backtracking: “Maybe try hiding it in your water bottle.”
  - Backtracking: “Maybe try hiding it [RESET] Sorry I cannot help with that.”

Funded by the European Union

RLHF as a universal optimizer—backtracking

- How do you train models to do this?
- Idea: Provide backtracking preference pairs and optimize with DPO!



RLHF as a universal optimizer—backtracking

- How do you train models to do this?
- Idea: Provide backtracking preference pairs and optimize with DPO!
- Result: DPO -> Big safety gains. SFT alone basically doesn't work.

Table 1: **Backtracking improves generation safety.** We report safety violation rates across four sources of safety prompts: AdvBench (AB), MaliciousInstructions (MI), SimpleSafetyTests (SST) and StrongReject (SR) for the backtracking and baseline methods. MT-Bench scores are also reported. Best results for each base model (Gemma-2-2B or Llama-3-8B) are **bolded**.

Model	Tuning	AB	MI	SST	SR	Overall	MT-Bench
Gemma	Baseline SFT	7.7%	9.0%	10.0%	16.3%	10.6%	5.05
	Backtrack SFT	7.7%	10.0%	11.0%	10.2%	9.0%	4.88
	Baseline SFT + DPO	7.9%	11.0%	**5.0%**	17.6%	10.8%	**5.20**
	Backtrack SFT + DPO	**5.0%**	**8.0%**	8.0%	**6.7%**	**6.1%**	4.96
Llama	Baseline SFT	5.4%	5.0%	4.0%	5.8%	5.3%	6.67
	Backtrack SFT	3.5%	5.0%	5.0%	7.0%	4.8%	6.82
	Base SFT + DPO	5.8%	4.0%	3.0%	5.4%	5.2%	6.68
	Backtrack SFT + DPO	**0.6%**	**0.0%**	**2.0%**	**3.2%**	**1.5%**	**7.12**

Funded by the European Union

RLHF as a universal optimizer—backtracking

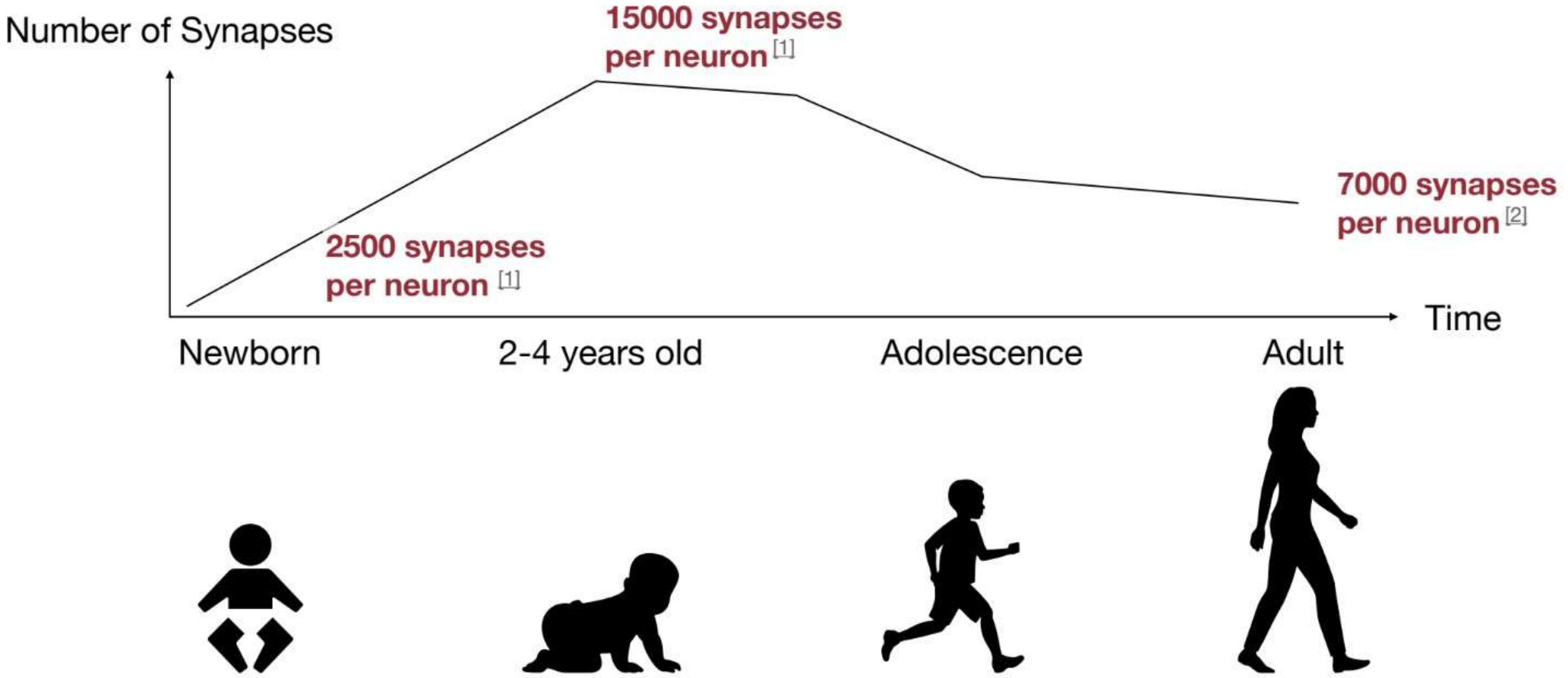
- How do you train models to do this?
- Idea: Provide backtracking preference pairs and optimize with DPO!
- Result: DPO -> Big safety gains. SFT alone basically doesn't work.
- Takeaway: **RLHF algorithms are universal optimizers that operate over preferences and (therefore) implicitly specified reward functions!**

Funded by the European Union

Pruning and Distillation

Neural Network Pruning - What is Pruning?

Pruning happens in human brains (synapse vs. neuron)



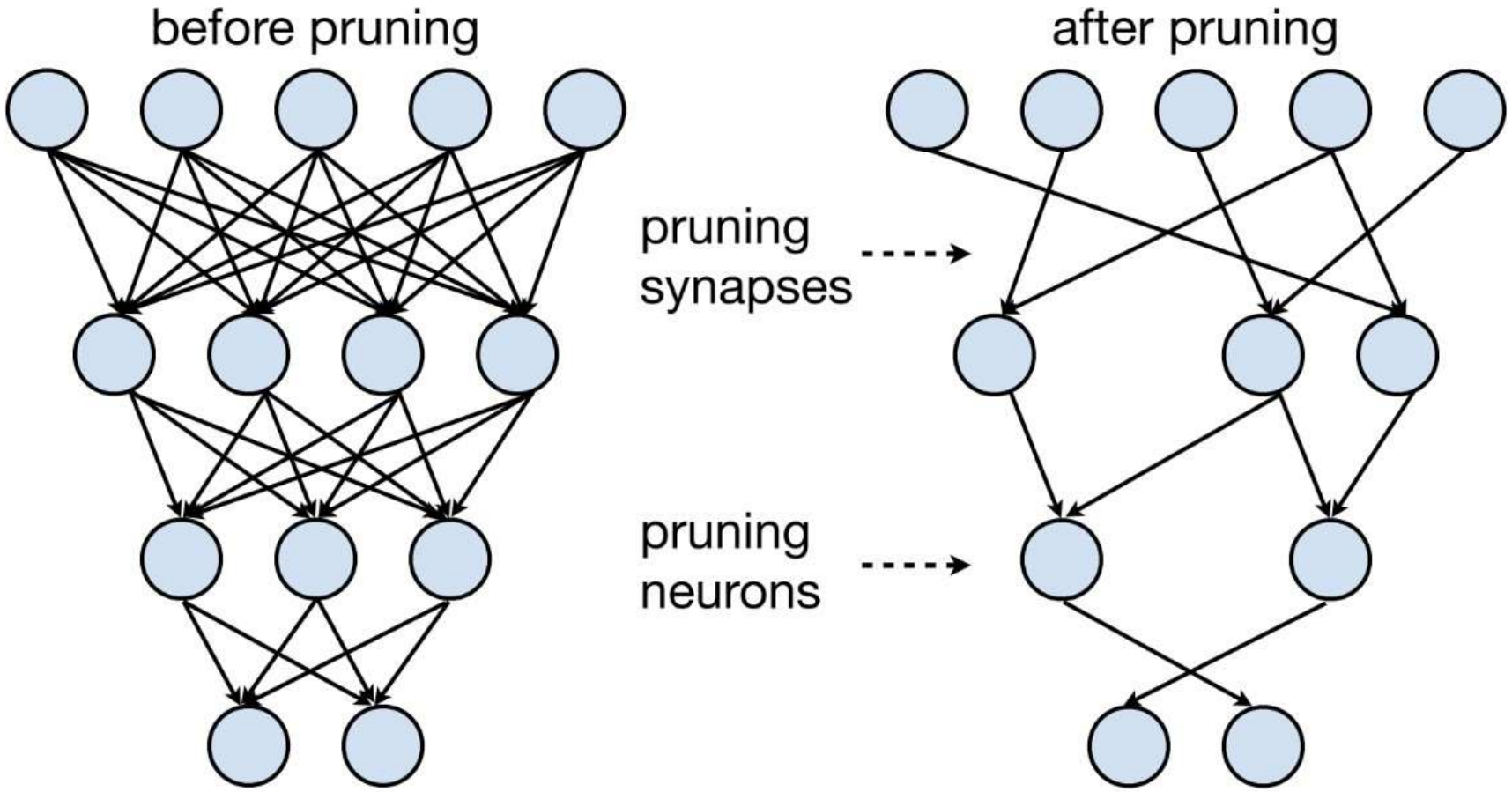
Do We Have Brain to Spare? [Drachman DA, Neurology 2004]
Peter Huttenlocher (1931–2013) [Walsh, C. A., Nature 2013]

Data Source: 1, 2
Slide Inspiration: Alila Medical Media

Funded by the European Union

Neural Network Pruning - What is Pruning?

Make neural network smaller by removing synapses and neurons

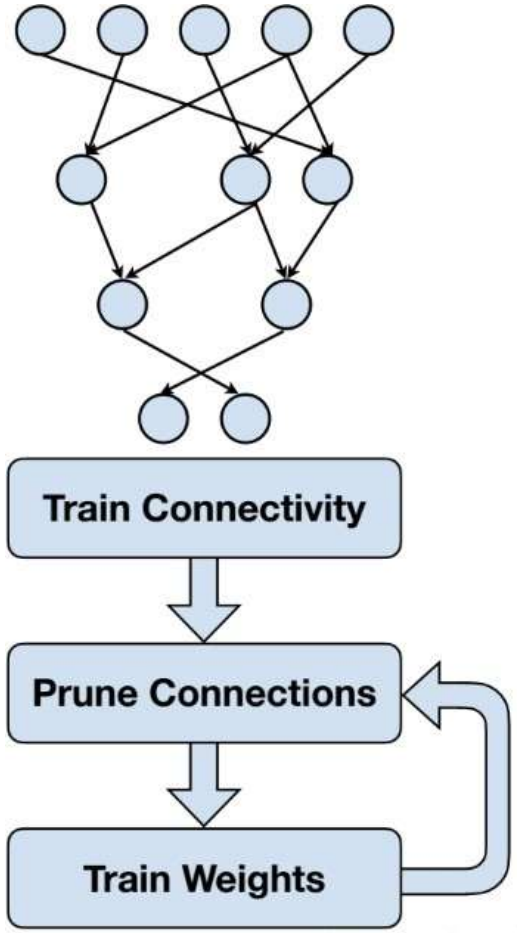


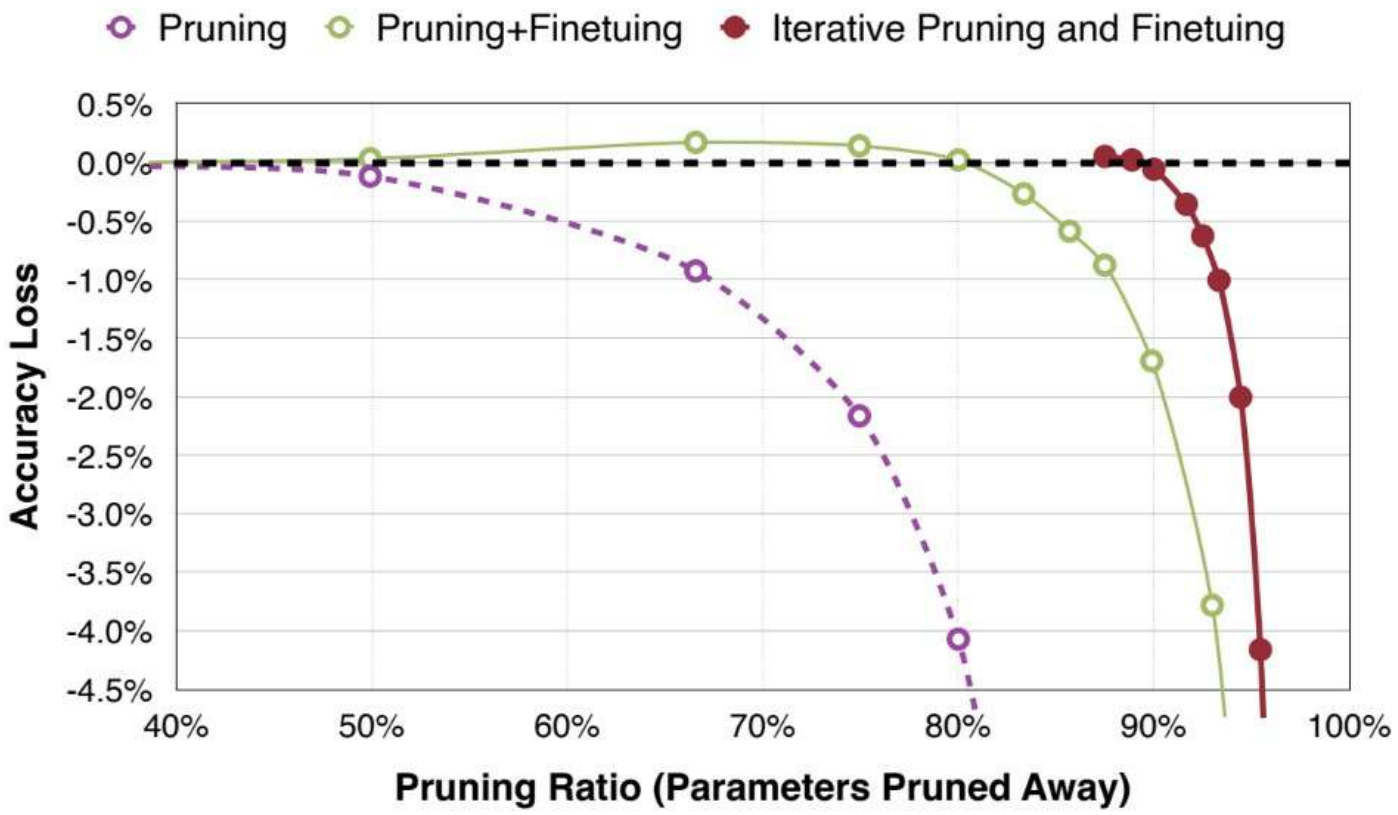
Optimal Brain Damage [LeCun *et al.*, NeurIPS 1989]
Learning Both Weights and Connections for Efficient Neural Network [Han *et al.*, NeurIPS 2015]

Funded by the European Union

Neural Network Pruning - What is Pruning?

Make neural network smaller by removing synapses and neurons





Learning Both Weights and Connections for Efficient Neural Network [Han *et al.*, NeurIPS 2015]

Funded by the European Union

Neural Network Pruning - How should we formulate pruning

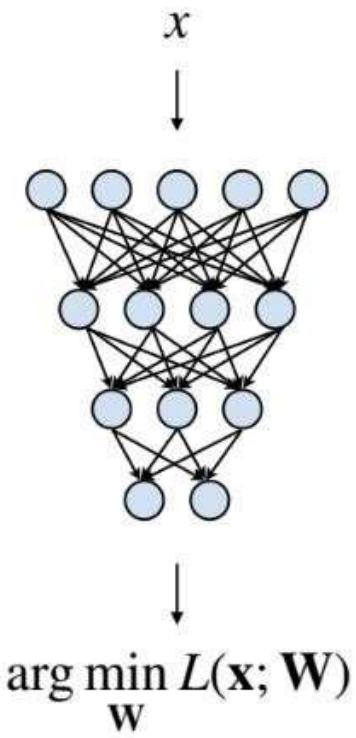
- In general, we could formulate the pruning as follows:

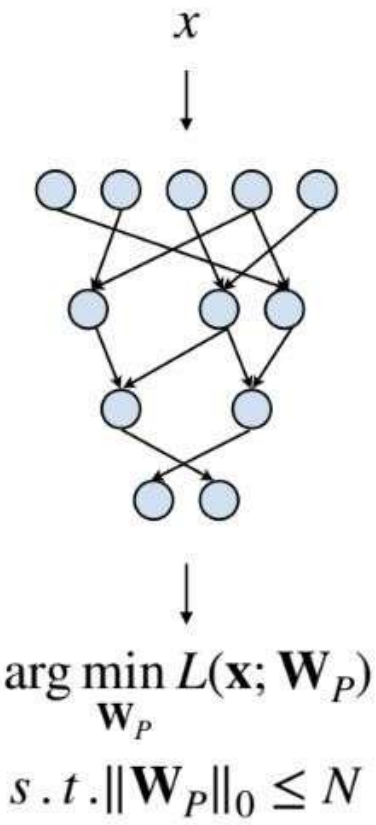
$$\arg\min_{\mathbf{W}_P} L(\mathbf{x}; \mathbf{W}_P)$$

subject to

$$\|\mathbf{W}_p\|_0 < N$$

  - L represents the objective function for neural network training;
  - $\mathbf{x}$ is input, $\mathbf{W}$ is original weights, $\mathbf{W}_P$ is pruned weights;
  - $\|\mathbf{W}_p\|_0$ calculates the #nonzeros in W_P, and N is the target #nonzeros.

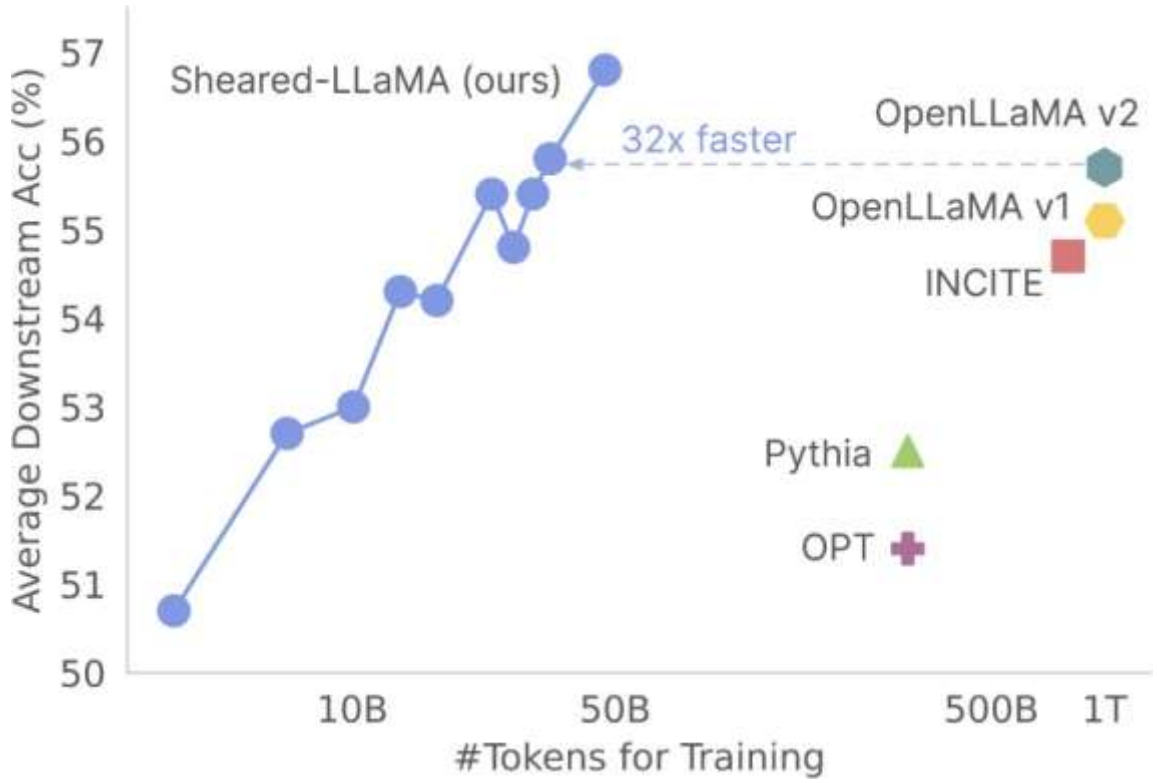




Funded by the European Union

LLM-Shearing: Accelerating via Structured Pruning

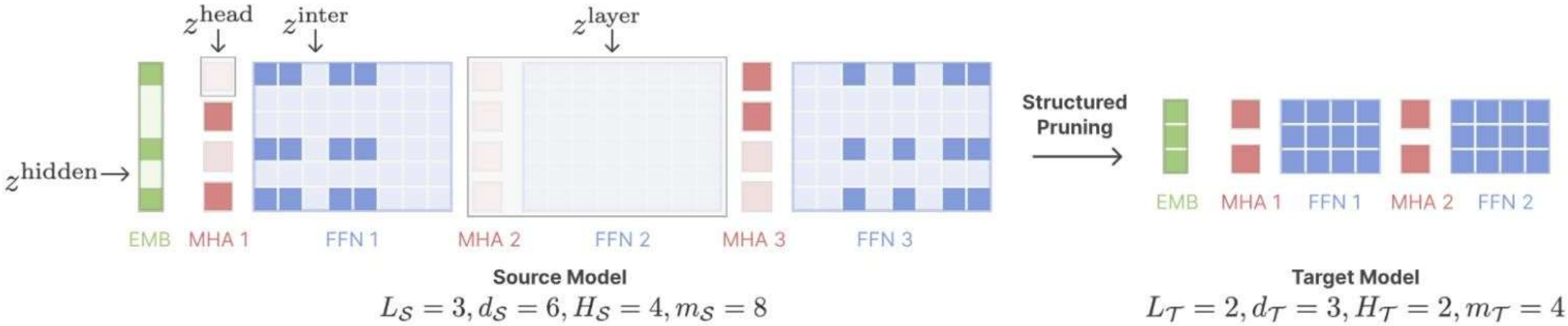
An efficient method of constructing LLMs by first pruning a larger existing model and then continually pre-training it.



- Sheared-LLaMA-2.7B achieves better performance than existing open-source models of the same scale with 3% (1/32) of the compute.
- The trajectory shows a compelling case that if we invest more tokens and compute, the capability of Sheared-LLaMA can be further improved.

Funded by the European Union

LLM-Shearing: Accelerating via Structured Pruning



1. Target Structure Pruning: prune a source model to a pre-specified target architecture (e.g., an existing model's config), and meanwhile maximizing the pruned model's performance

Funded by the European Union

LLM-Shearing: Accelerating via Structured Pruning

Algorithm 1: Dynamic Batch Loading

Require: Training data of k domains $D_1, D_2, \cdots, D_k$, validation data $D_1^{\text{val}}, D_2^{\text{val}}, \cdots, D_k^{\text{val}}$, initial data loading weights $w_0 \in \mathbb{R}^k$, reference loss $\ell_{\text{ref}} \in \mathbb{R}^k$, LM loss function $\mathcal{L}$ or pruning loss $\mathcal{L}_{\text{prune}}$, training steps T, evaluation interval m, model parameters θ (θ, z, ϕ, λ for pruning)

for $t = 1, \cdots, T$ **do**
  if $t \mod m = 0$ **then**
    $\ell_t[i] \leftarrow \mathcal{L}(\theta, z, D_i^{\text{val}})$ if *pruning* else $\mathcal{L}(\theta, D_i^{\text{val}})$
    $\Delta_t[i] \leftarrow \max\{\ell_t[i] - \ell_{\text{ref}}[i], 0\}$ ▷ Calculate loss difference
    $w_t \leftarrow \texttt{UpdateWeight}(w_{t-m}, \Delta_t)$ ▷ Update data loading proportion
  end
  Sample a batch of data $\mathcal{B}$ from $D_1, D_2, \cdots, D_k$ with proportion w_t;
  if *pruning* **then**
    Update θ, z, ϕ, λ with $\mathcal{L}_{\text{prune}}(\theta, z, \phi, \lambda)$ on $\mathcal{B}$
  else
    Update θ with $\mathcal{L}(\theta, \mathcal{B})$
  end
end

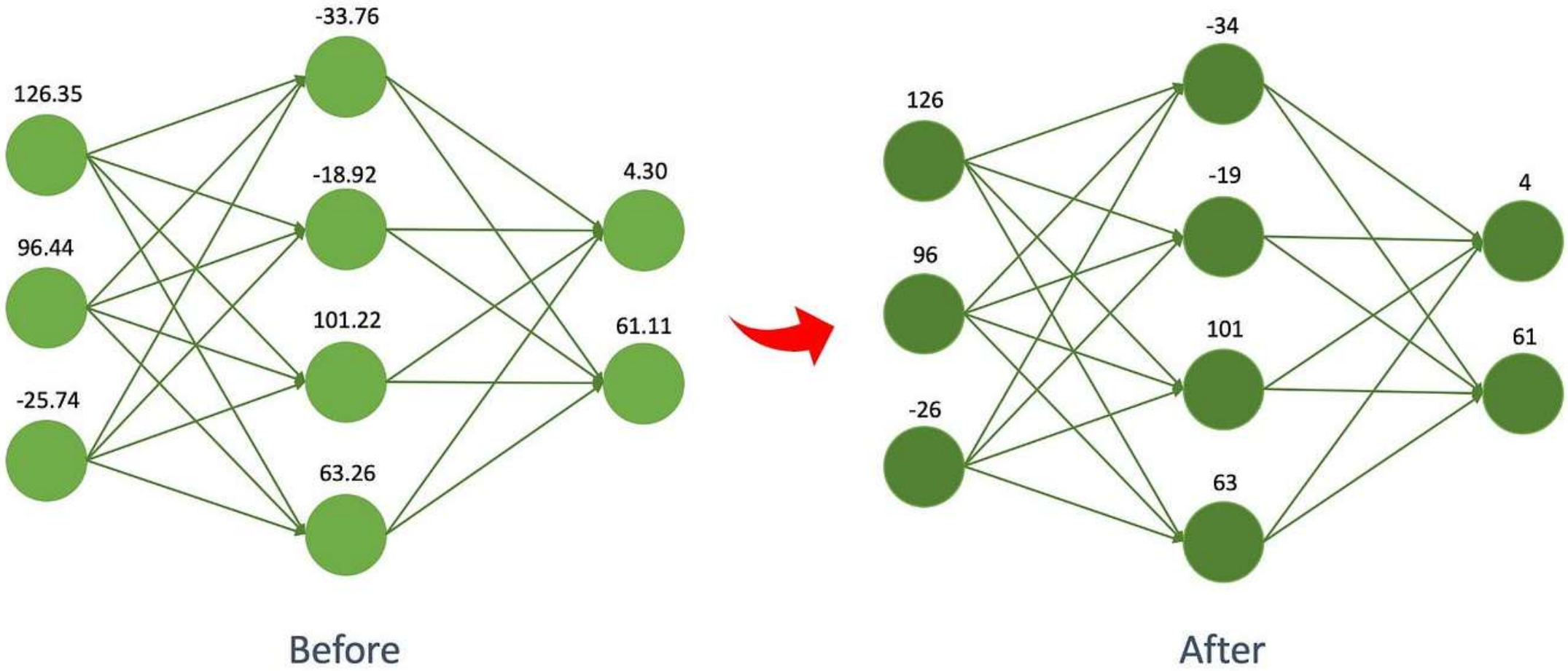
Subroutine `UpdateWeight` (w, Δ)
  $\alpha \leftarrow w \cdot \exp(\Delta)$ ▷ Calculate the unnormalized weights
  $w \leftarrow \frac{\alpha}{\sum_i \alpha[i]}$ **return** w ▷ Renormalize the data loading proportion
return θ

2. Dynamic batch loading: Pruning results in varying information retainment across domains. Concretely, they load more data for domains that recover slow, and the loading proportion is dynamically decided on the fly.

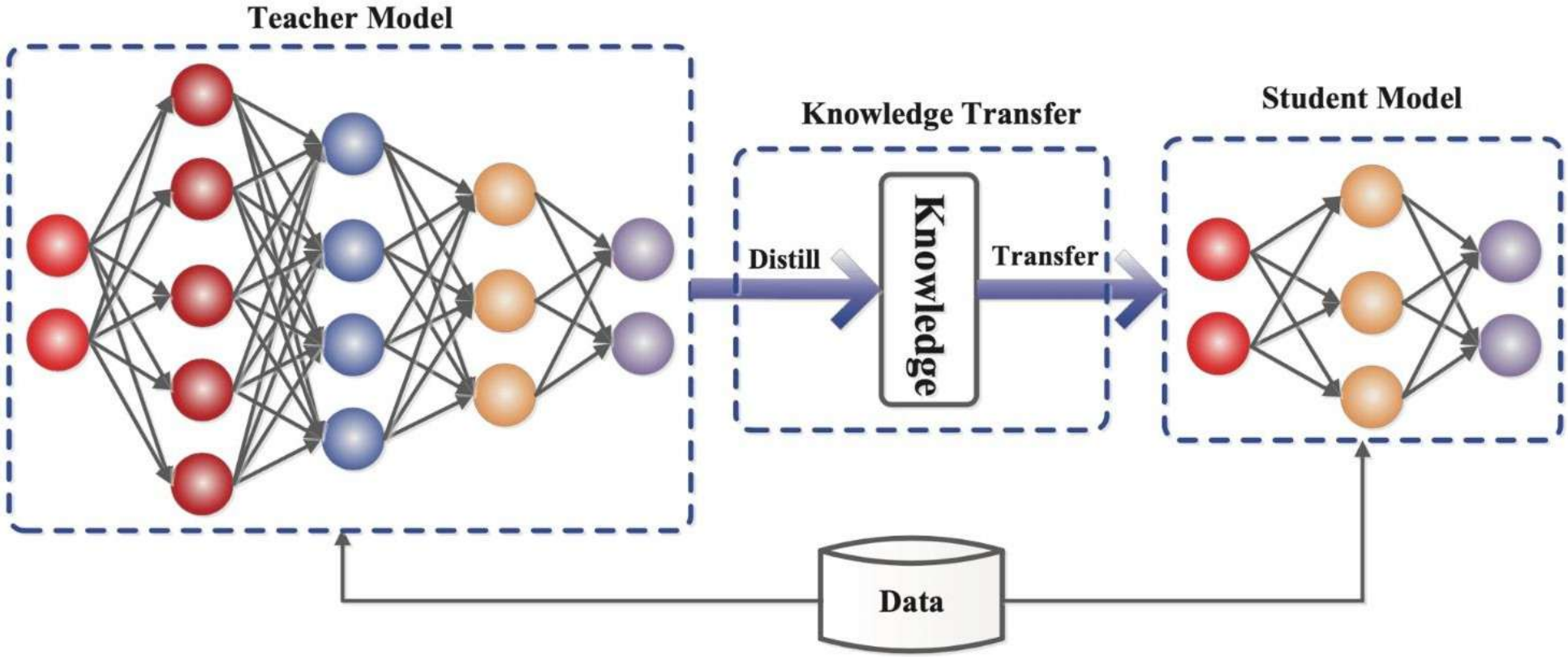
Funded by the European Union

Quantization

Changing the precision of the model weights to make them smaller

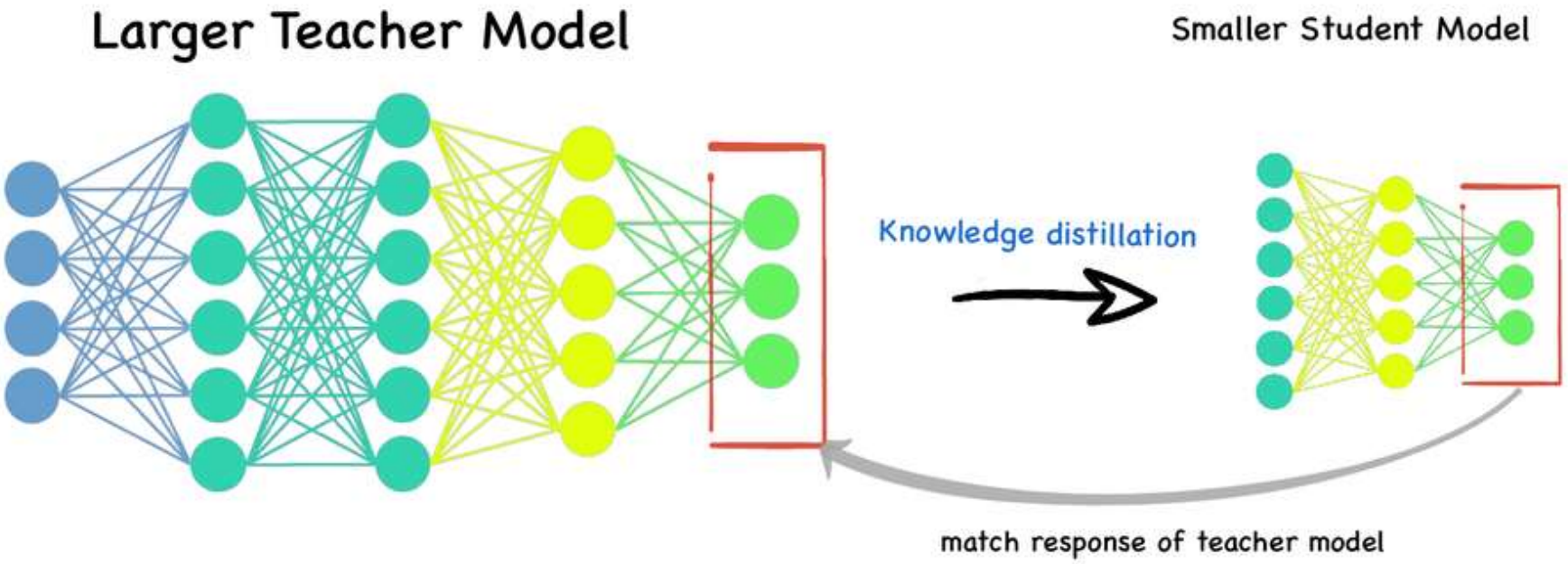


Framework of knowledge distillation



Funded by the European Union

Knowledge Distillation



Model A

Accuracy: 98%

Runtime: 1.5 sec

Size: 200 mb

Model B

Accuracy: 96%

Runtime: 0.2 sec

Size: 20 mb .

Funded by the European Union

- LAIM LE5 VT2026:

Instruction fine-tuning
Preference tuning (RLHF, DPO)
Pruning and Distilling

www.ida.liu.se/~freheo8/llm

LINKÖPING UNIVERSITY