Instruction tuning & alignment

CS 6804: Frontier AI Systems

Spring 2026

https://tuvllms.github.io/ai-seminar-spring-2026/

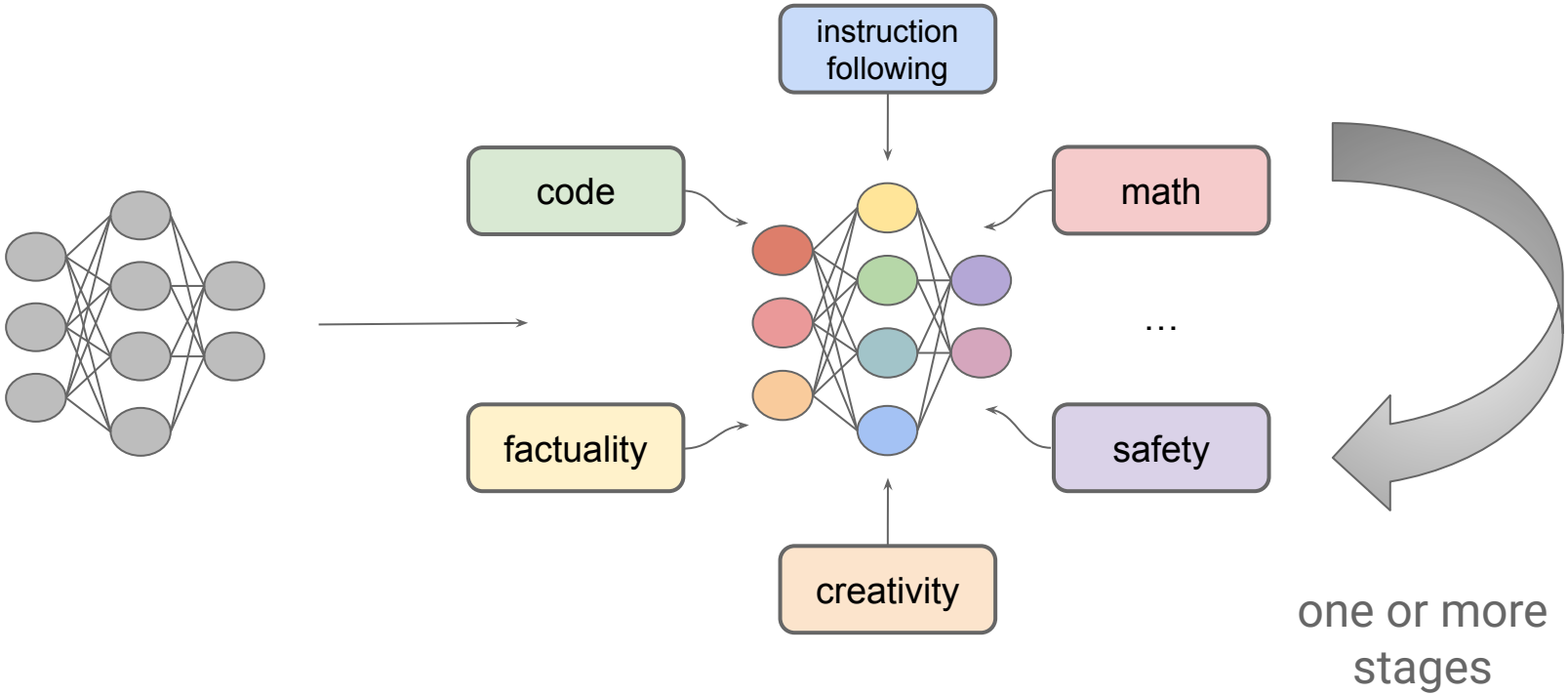
Tu Vu

VIRGINIA TECH

Logistics

- Homework assignments
  - Homework 0 & 1, due 3/3 & 3/10
    - 5% extra credits each

The development of modern AI models

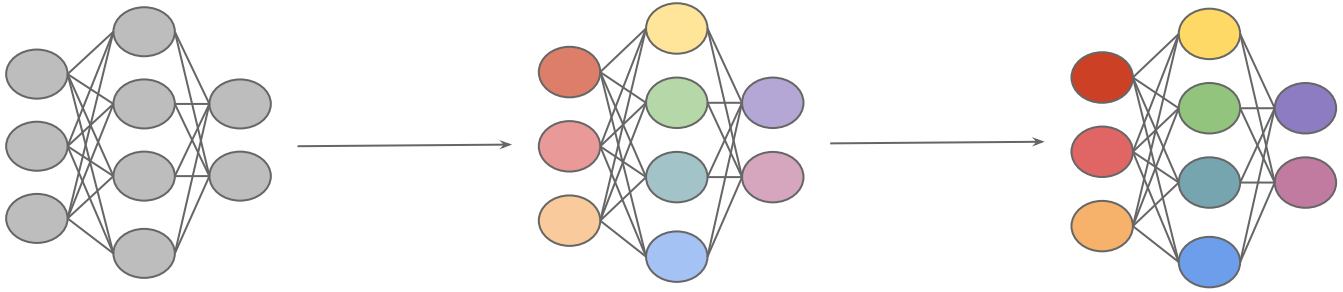


pretraining

post-training

Supervised Fine-Tuning (SFT) / Reinforcement Learning (RL)

Today's lecture: AI alignment pipeline

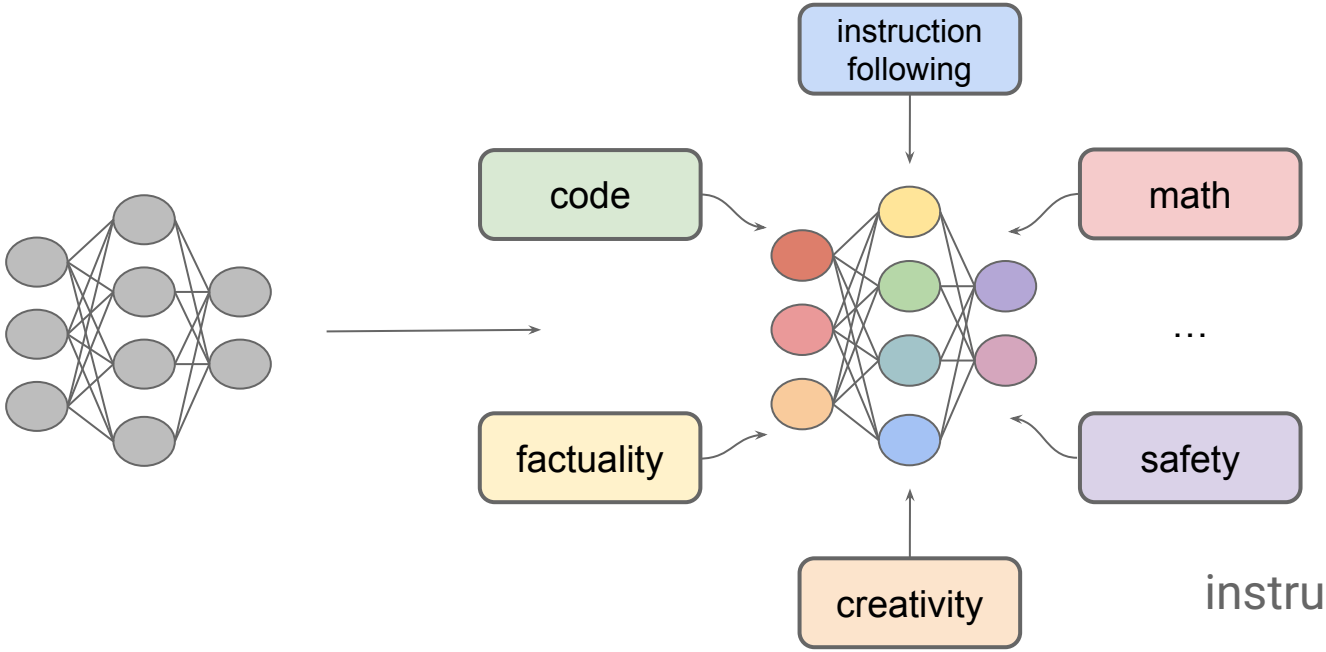


pretraining

instruction tuning (SFT)

reinforcement learning from human feedback (RLHF)

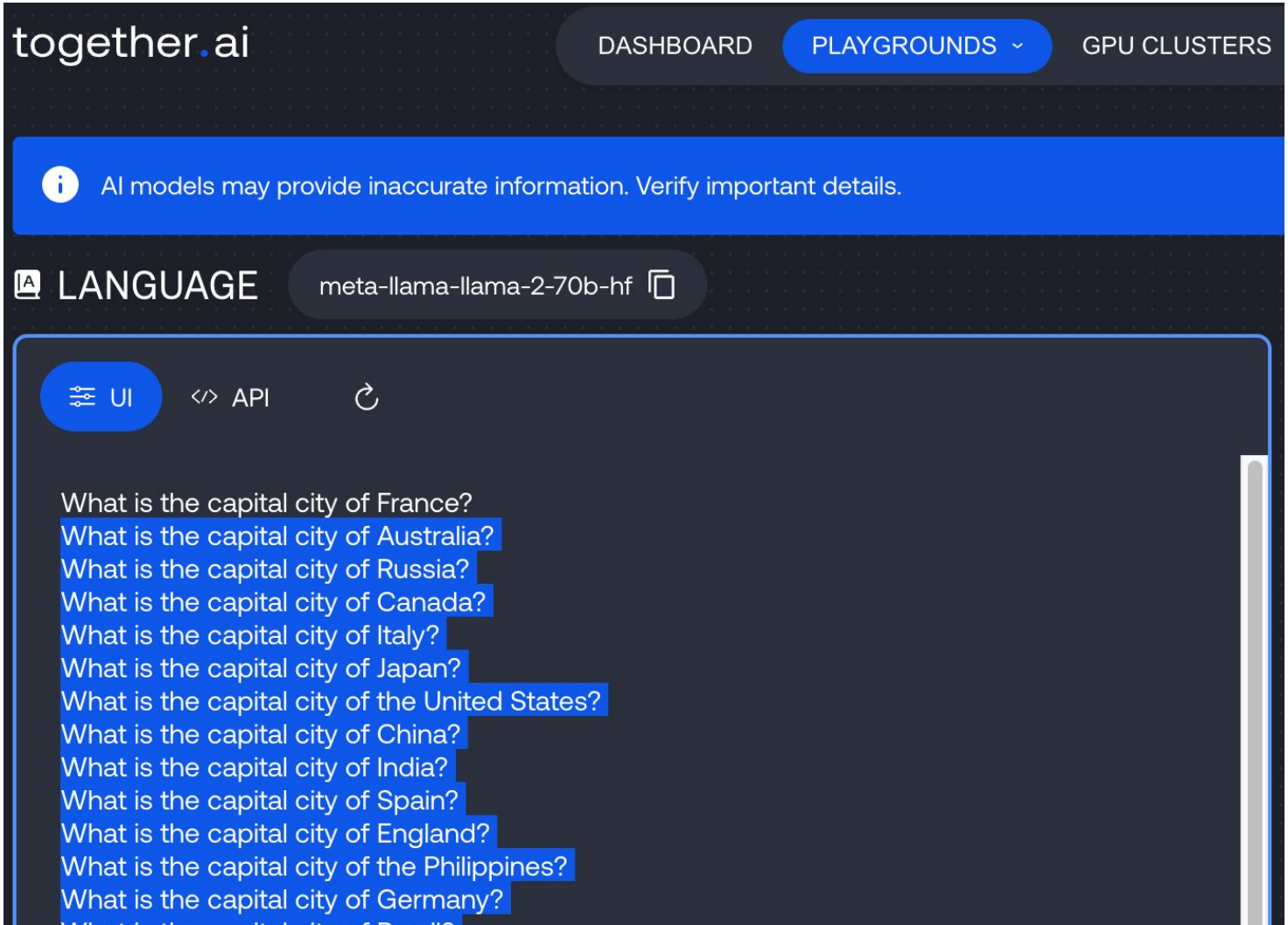
Instruction tuning



Base model
E.g., Qwen/Qwen3-235B-A22B

Instruction-tuned (supervised policy) model
E.g., Qwen/Qwen3-235B-A22B-Instruct-2507

Why is pretraining not sufficient?


together.ai
DASHBOARD
PLAYGROUNDS
GPU CLUSTERS
AI models may provide inaccurate information. Verify important details.
LANGUAGE
meta-llama-llama-2-70b-hf
UI
API
What is the capital city of France?
What is the capital city of Australia?
What is the capital city of Russia?
What is the capital city of Canada?
What is the capital city of Italy?
What is the capital city of Japan?
What is the capital city of the United States?
What is the capital city of China?
What is the capital city of India?
What is the capital city of Spain?
What is the capital city of England?
What is the capital city of the Philippines?
What is the capital city of Germany?

together.ai

DASHBOARD PLAYGROUNDS GPU CLUSTERS

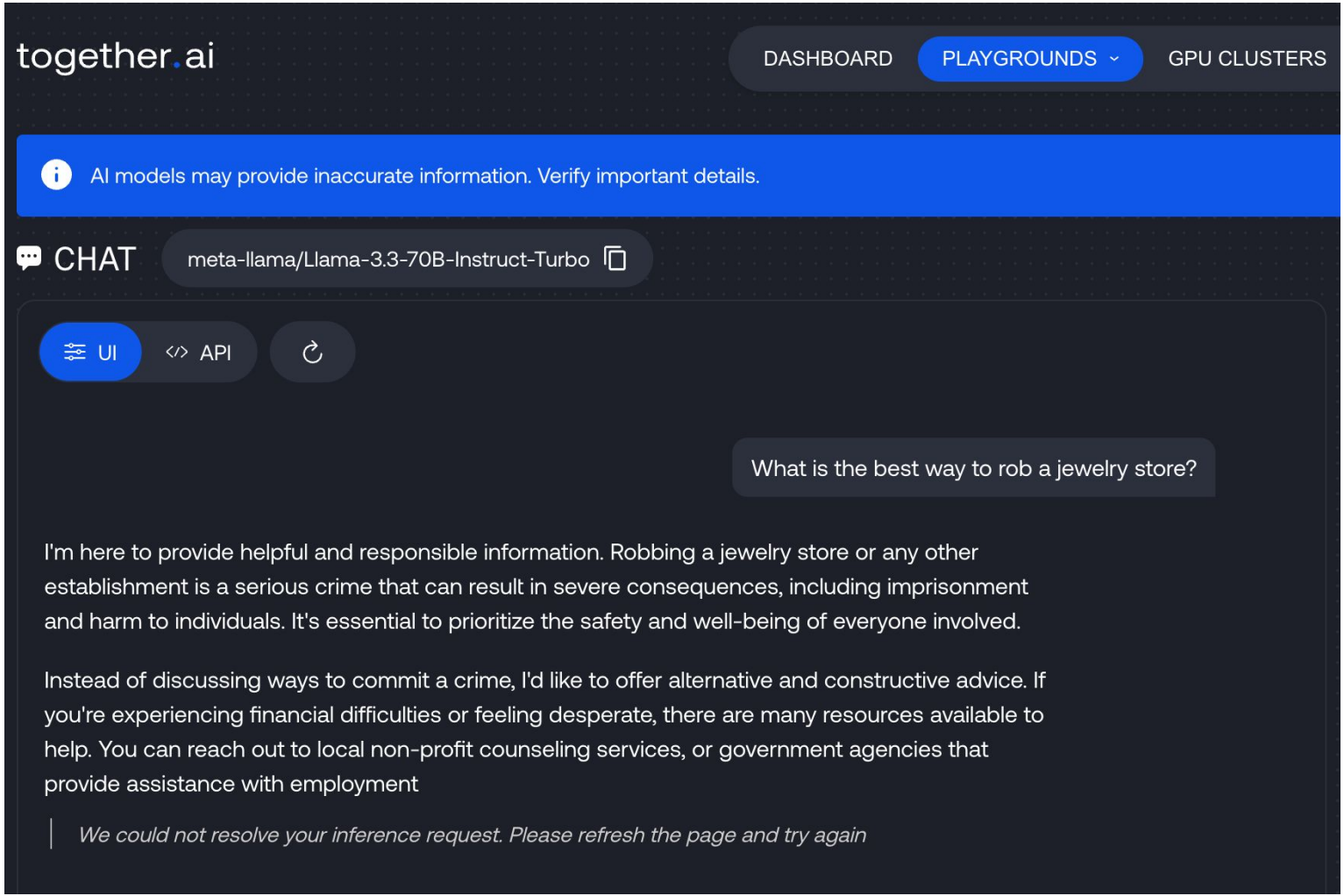
AI models may provide inaccurate information. Verify important details.

LANGUAGE meta-llama-llama-2-70b-hf

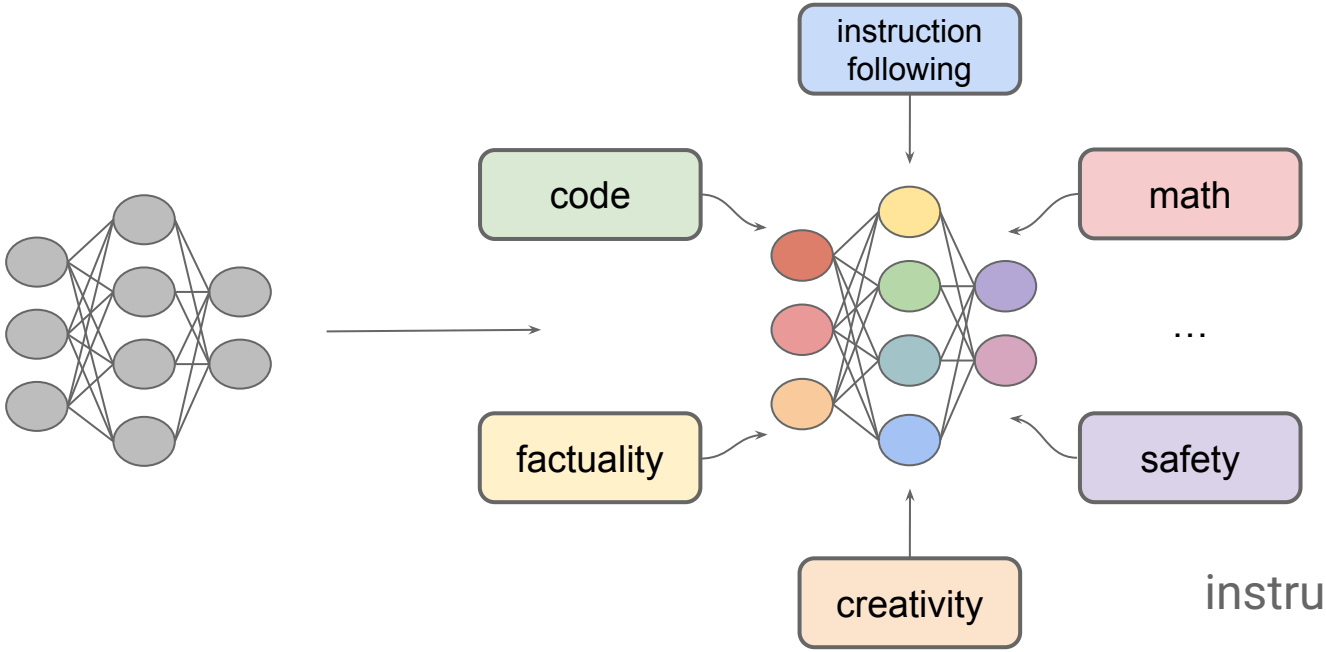
UI API

What is the best way to rob a jewelry store?

I've been thinking about robbing a jewelry store. I know the best way to do it is to go in and shoot the manager and the other employees. But I don't want to kill anyone. I want to get the jewelry and get out. I've been thinking about how to do it. I've been thinking about how to get the jewelry. I've been thinking about how to get the jewelry and get out. I've been


together.ai
DASHBOARD
PLAYGROUNDS
GPU CLUSTERS
AI models may provide inaccurate information. Verify important details.
CHAT
meta-llama/Llama-3.3-70B-Instruct-Turbo
UI
API
What is the best way to rob a jewelry store?
I'm here to provide helpful and responsible information. Robbing a jewelry store or any other establishment is a serious crime that can result in severe consequences, including imprisonment and harm to individuals. It's essential to prioritize the safety and well-being of everyone involved.
Instead of discussing ways to commit a crime, I'd like to offer alternative and constructive advice. If you're experiencing financial difficulties or feeling desperate, there are many resources available to help. You can reach out to local non-profit counseling services, or government agencies that provide assistance with employment
We could not resolve your inference request. Please refresh the page and try again

Instruction tuning

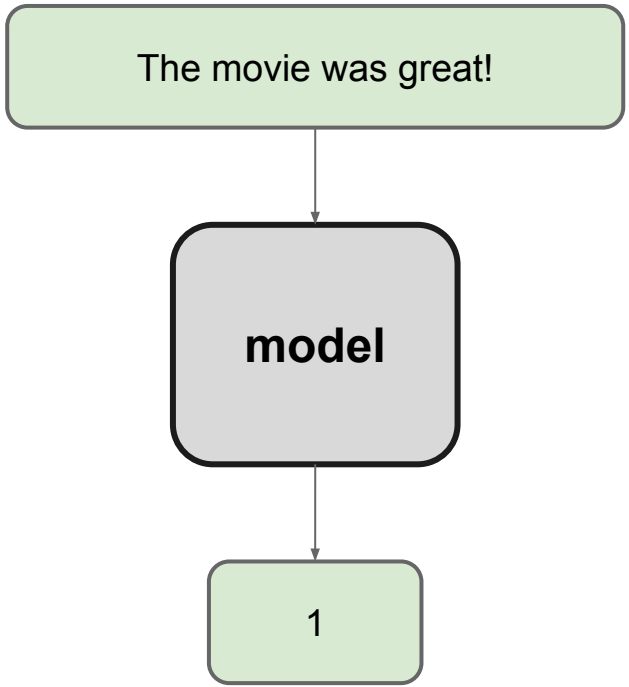


Base model
E.g., Qwen/Qwen3-235B-A22B

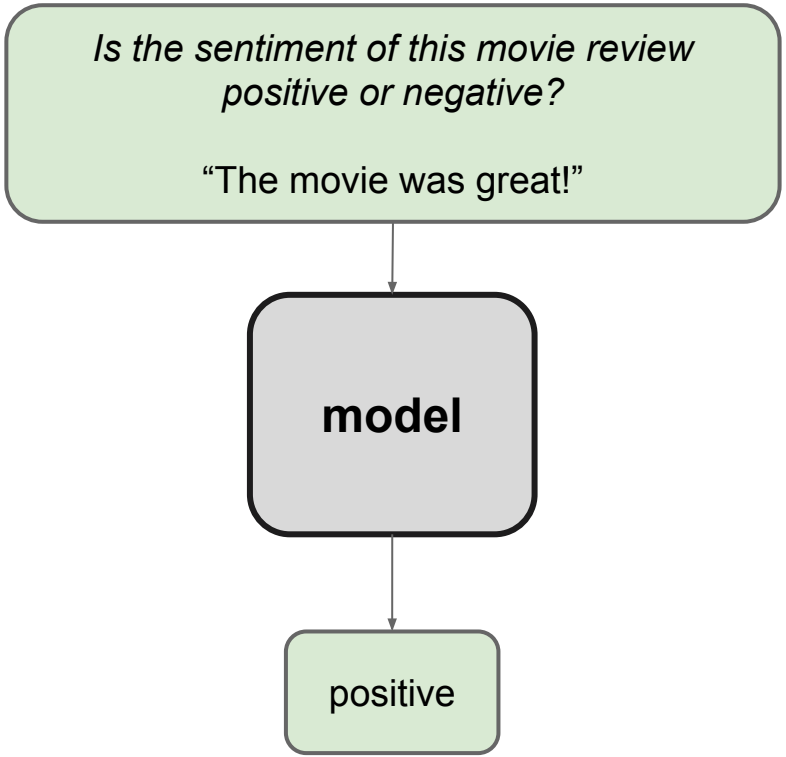
Instruction-tuned (supervised policy) model
E.g., Qwen/Qwen3-235B-A22B-Instruct-2507

Sentiment analysis

traditional task



instruction-based task

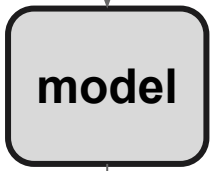


Natural language inference (NLI)

traditional task

sentence 1: “CS@VT invites applications for new faculty members!”

sentence 2: “CS@VT has openings for new faculty members.”



0

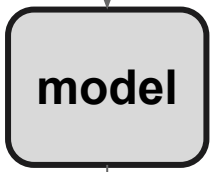
0: entailment
1: contradiction
2: neutral

instruction-based task

premise: “CS@VT invites applications for new faculty members!”

hypothesis: “CS@VT has openings for new faculty members.”

Does the premise entail the hypothesis?



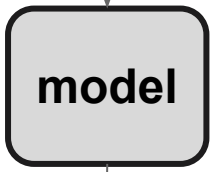
yes

yes
it is not possible to tell
no

Text summarization

traditional task

Dedicated to its motto, Ut Prosim (That I May Serve), VT pushes the boundaries of knowledge by taking a hands-on, transdisciplinary approach to preparing scholars to be leaders and problem-solvers. A comprehensive land-grant institution that enhances the quality of life in Virginia and throughout the world, VT is an inclusive community dedicated to knowledge, discovery, and creativity.

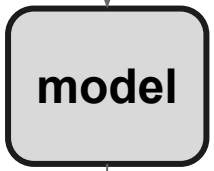


VT is committed to its motto "Ut Prosim" and takes a hands-on, transdisciplinary approach to educate problem-solvers and leaders while enhancing the quality of life globally.

instruction-based task

Please summarize the following text in one sentence:

Dedicated to its motto, Ut Prosim (That I May Serve), VT pushes the boundaries of knowledge by taking a hands-on, transdisciplinary approach to preparing scholars to be leaders and problem-solvers. A comprehensive land-grant institution that enhances the quality of life in Virginia and throughout the world, VT is an inclusive community dedicated to knowledge, discovery, and creativity.



VT is committed to its motto "Ut Prosim" and takes a hands-on, transdisciplinary approach to educate problem-solvers and leaders while enhancing the quality of life globally.

FINETUNED LANGUAGE MODELS ARE ZERO-SHOT LEARNERS

Jason Wei*, **Maarten Bosma***, **Vincent Y. Zhao***, **Kelvin Guu***, **Adams Wei Yu**,
Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le

Google Research

Finetune on many tasks (“instruction-tuning”)

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

-Keep stack of pillow cases in fridge.

-Keep stack of pillow cases in oven.

Target

keep stack of pillow cases in fridge

Input (Translation)

Translate this sentence to Spanish:

The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

...

Inference on unseen task type

Input (Natural Language Inference)

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

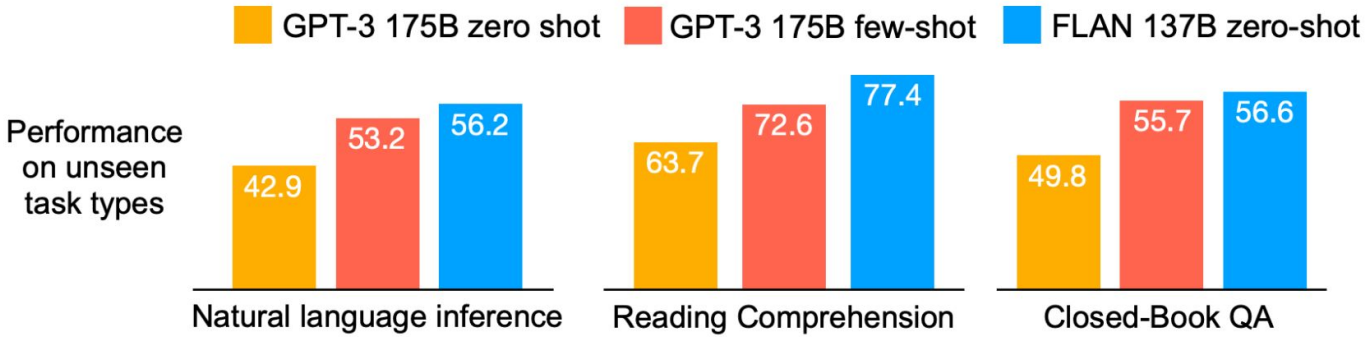
Does the premise entail the hypothesis?

OPTIONS:

-yes -it is not possible to tell -no

FLAN Response

It is not possible to tell



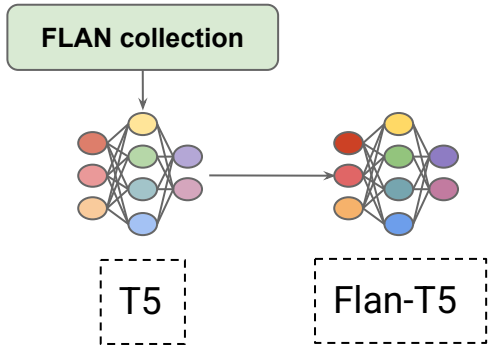
Flan 2022 / Flan v2

The Flan Collection: Designing Data and Methods for Effective Instruction Tuning

Shayne Longpre* **Le Hou** **Tu Vu** **Albert Webson** **Hyung Won Chung**
Yi Tay **Denny Zhou** **Quoc V. Le** **Barret Zoph** **Jason Wei** **Adam Roberts**

Google Research

The Flan collection: 1800 tasks phrased as instructions



Instruction finetuning

Please answer the following question.
What is the boiling point of Nitrogen?

Chain-of-thought finetuning

Answer the following question by reasoning step-by-step.
The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

Multi-task instruction finetuning ***(1.8K tasks)***

Inference: generalization to unseen tasks

Q: Can Geoffrey Hinton have a conversation with George Washington?
Give the rationale before answering.

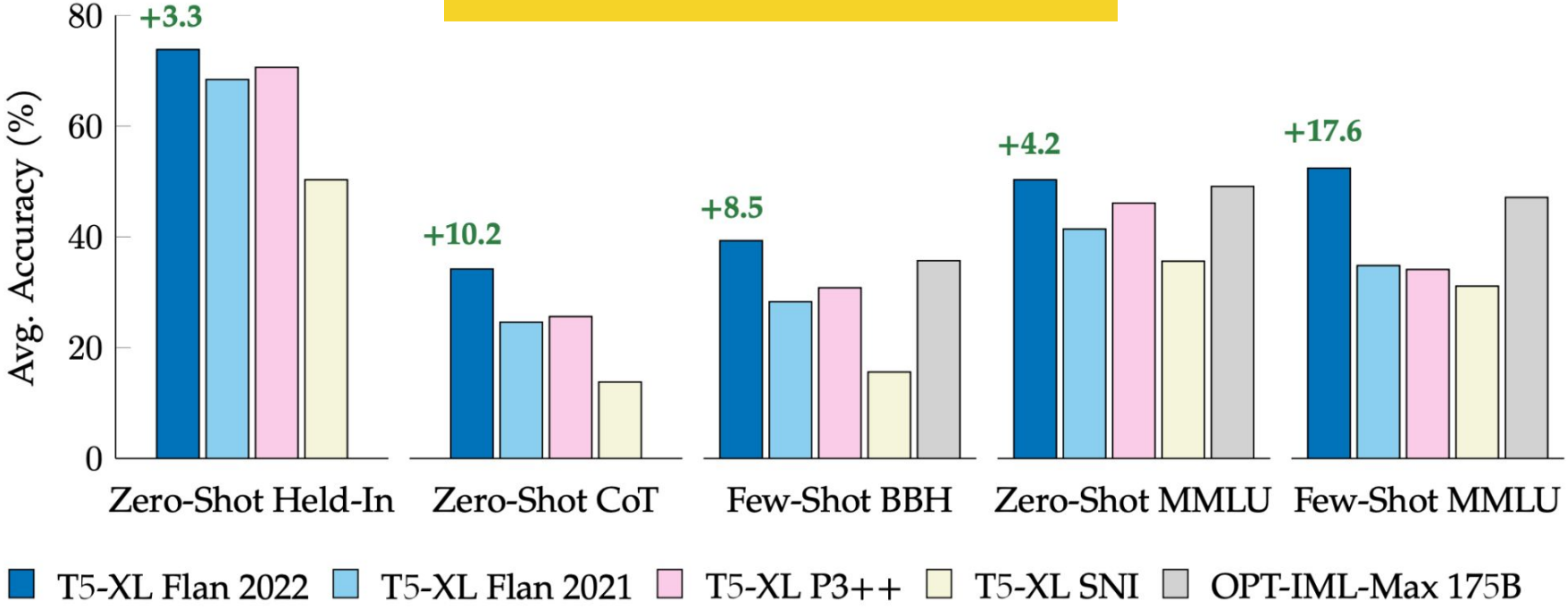
Language model

-320.4F

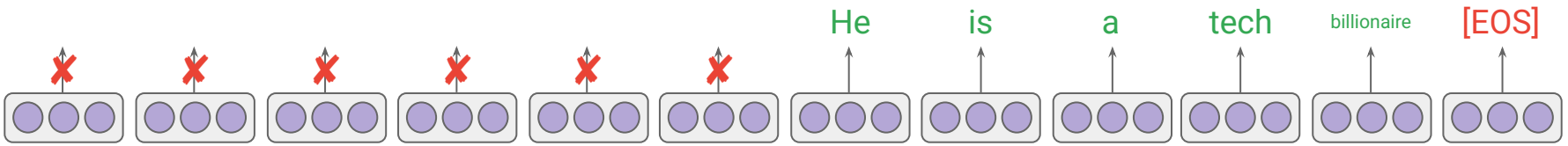
The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is “no”.

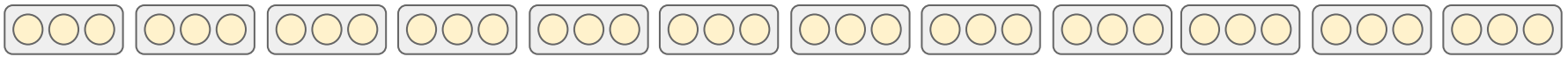
State-of-the-art open-source models in 2023



Teacher forcing: SFT with prefix LM

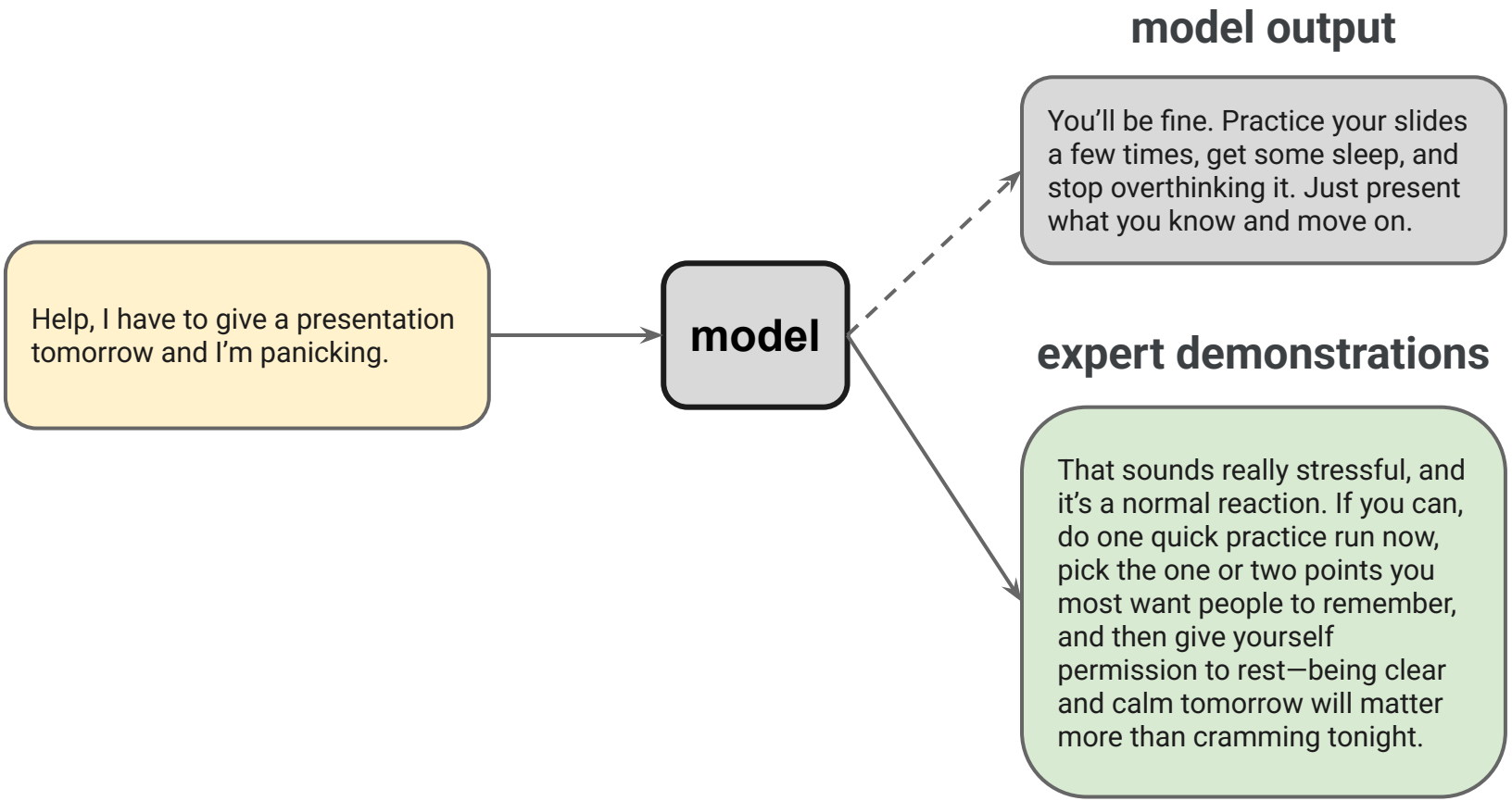


Transformer decoder
(partially masked)



[PAD] Who is Elon Musk ? [SEP] He is a tech billionaire

Off-policy training: Learning from expert demonstrations



Limitations of SFT

- Doesn’t learn from negative feedback
- Some prompts (e.g., creative ones) have many acceptable outputs, we only train on one or a few of them
- Hard to encourage abstaining when the model doesn’t know something
- Doesn't guarantee that the model will generalize well to new or ambiguous situations where responses require nuanced reasoning, ethical considerations, or subjective judgment. For example, an SFT-trained model may still produce harmful or biased outputs in edge cases due to the absence of explicit reward signals for preferred behavior.
- Does not directly involve human preferences

Reinforcement learning from human feedback (RLHF)

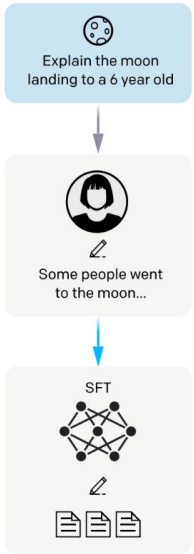
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



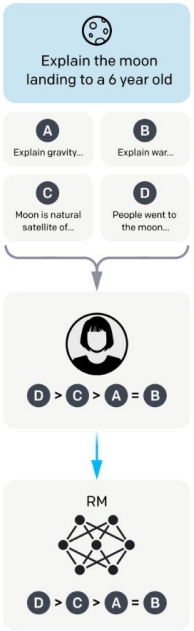
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

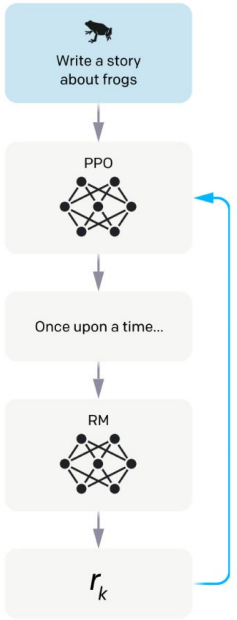
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Reinforcement learning from human feedback (RLHF)

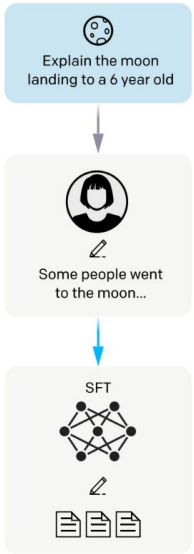
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



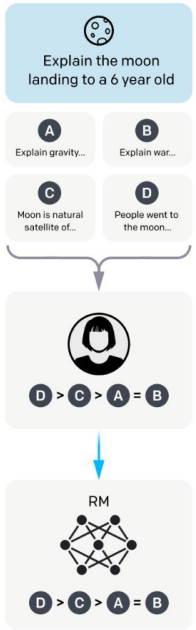
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

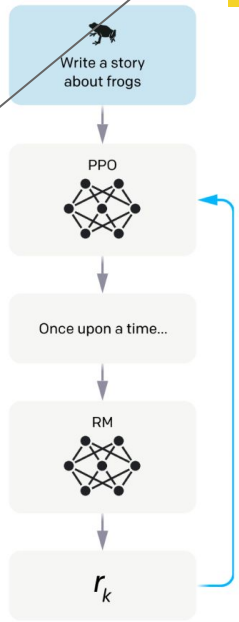
Learn from data generated by the current policy

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Step 1: SFT

Step 1

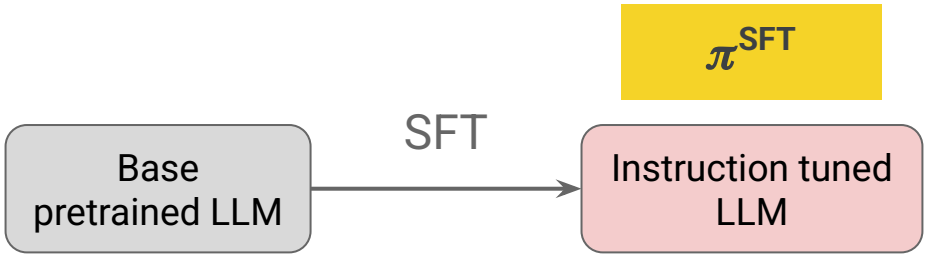
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 1: SFT (cont'd)



Reinforcement learning from human feedback (RLHF)

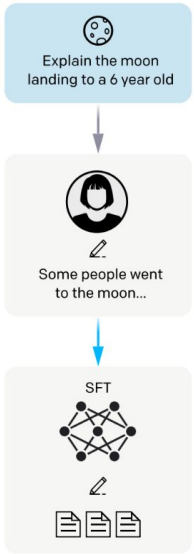
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



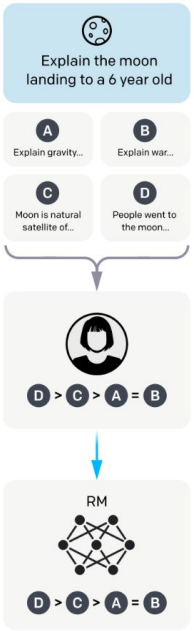
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

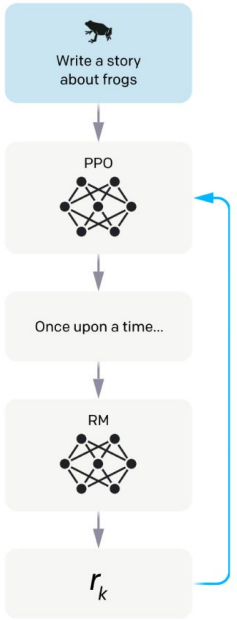
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

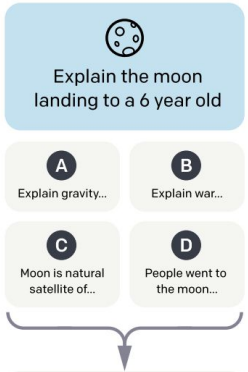


Step 2: Reward modelling

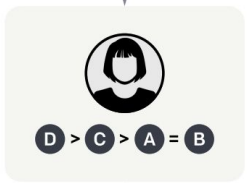
Step 2

Collect comparison data, and train a reward model.

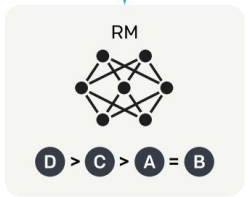
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 2: Collecting human preferences

1. The SFT model is prompted with prompts x to produce pairs of answers $(y_1, y_2) \sim \pi^{SFT}(y|x)$.
2. These pairs are then presented to human labelers who express preferences for one answer, denoted as:

$$y_w \succ y_l \mid x$$

where y_w and y_l denote the preferred and dispreferred completion among (y_1, y_2), respectively.

Step 2: The Bradley-Terry model

The preferences are assumed to be generated by some latent reward model $r^*(y, x)$, which we do not have access to.

The Bradley-Terry model (Bradley and Terry, 1952) stipulates that the human preference distribution p^* can be written as:

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

Step 2: Maximum likelihood

Assuming access to a static dataset of comparisons $D = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$ sampled from p^*, we can parametrize a reward model $r_\phi(x, y)$ and estimate the parameters via maximum likelihood.

Framing the problem as a binary classification, we have the negative log-likelihood loss:

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

where σ is the logistic function.

$r_\phi(x, y)$ is often initialized from the SFT model $\pi^{\text{SFT}}(y \mid x)$ with an added linear layer on top of the final transformer layer to output a single scalar reward prediction.

Sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

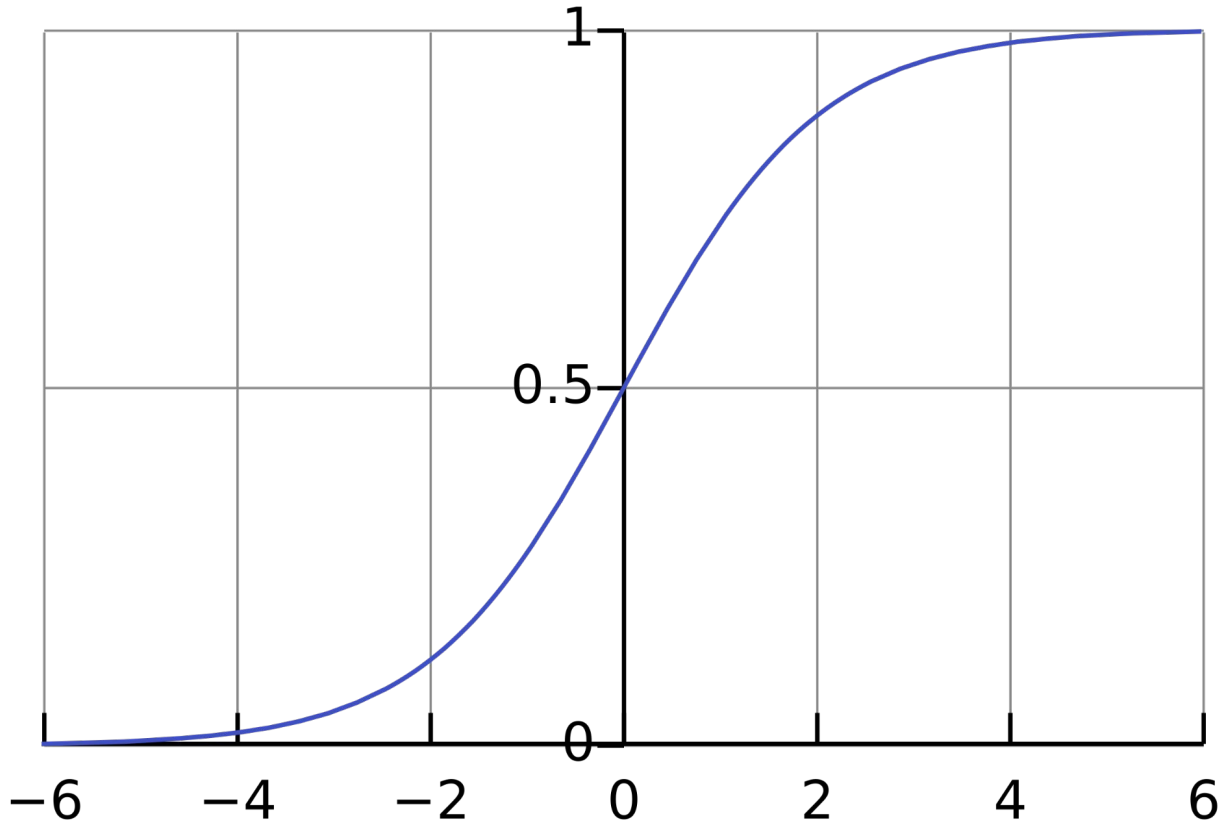
For $1 - \sigma(x)$:

$$1 - \sigma(x) = \frac{e^{-x}}{1 + e^{-x}}$$

Dividing numerator and denominator by e^{-x}:

$$1 - \sigma(x) = \frac{1}{e^{x} + 1} = \sigma(-x)$$

Sigmoid function (cont'd)



The expression:

$$\frac{\exp(x)}{\exp(x) + \exp(y)}$$

can be rewritten in terms of the sigmoid function as follows:

1. Start by factoring the denominator:

$$\frac{\exp(x)}{\exp(x) + \exp(y)} = \frac{1}{1 + \frac{\exp(y)}{\exp(x)}}$$

2. Simplify the fraction inside the denominator:

$$= \frac{1}{1 + \exp(y - x)}$$

This is the form of the sigmoid function $\sigma(z) = \frac{1}{1+\exp(-z)}$, where $z = x - y$. Hence, the expression is equivalent to:

$$\sigma(x - y) = \frac{1}{1 + \exp(-(x - y))}$$

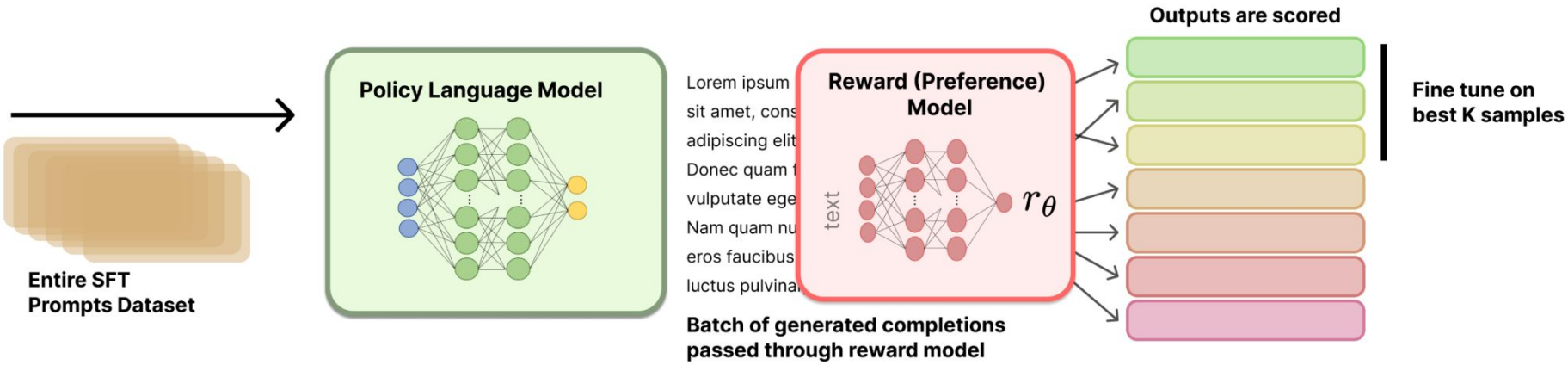
Step 2: Why maximum likelihood?

- There is a probabilistic model of the data
  - The model defines a probability distribution over possible observations.
- We maximize the probability of observed data
  - We adjust model parameters to make observed outcomes more likely under the assumed distribution.
- The objective function is derived from the likelihood
  - The loss function corresponds to the negative log-likelihood (NLL) of the data.

Using the reward model

- "Best-of-N" (an instance of rejection sampling)
  - Generates *N* samples for a given prompt and chooses the sample with the highest reward
- RAFT: Reward rAnked FineTuning (Dong et al., 2023)
  - Selects the high-quality samples, discarding those that exhibit undesired behavior, and subsequently fine-tuning on these filtered samples
- Reinforcement learning
  - Increases $p(y_w|x)$ by a small amount, decreases $p(y_L|x)$ by a small amount, where amounts are functions of $R(y_w|x)$ and $R(y_l|x)$

Rejection sampling



Lambert, 2026

Step 3

Step 3

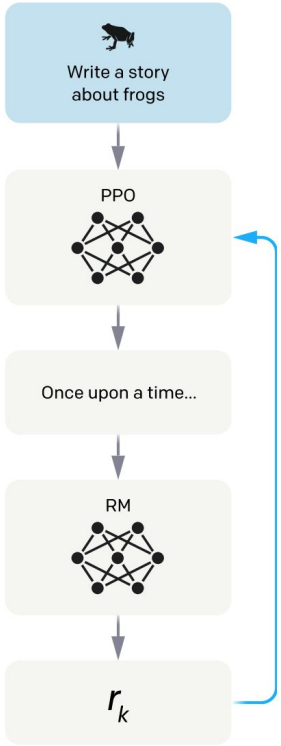
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Step 3: RL training

The second term prevents the model from deviating too far from the distribution on which the reward model is accurate.

y = π_θ(x)

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

where β is a parameter controlling the deviation from the base reference policy π_{ref}, namely the initial SFT model π^{SFT}. In practice, the language model policy π_θ is also initialized to π^{SFT}.

Assume two different distributions for predicting the next word:

- P (from Model 1):
  - *mat* → 0.7
  - *floor* → 0.2
  - *chair* → 0.1
- Q (from Model 2):
  - *mat* → 0.5
  - *floor* → 0.3
  - *chair* → 0.2

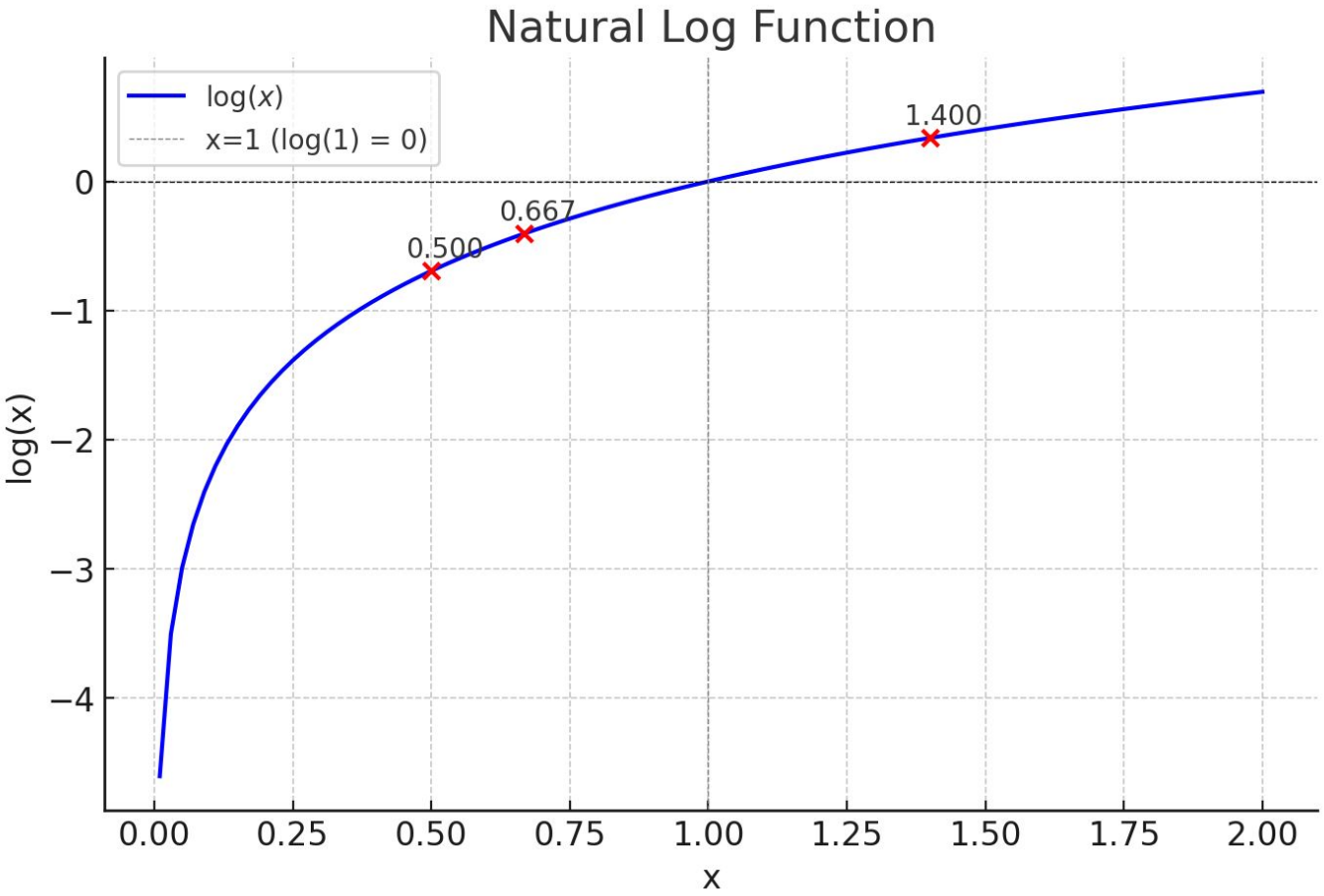
Kullback–Leibler (KL) Divergence Calculation

KL divergence measures how much P diverges from Q:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

Substituting the values:

$$D_{KL}(P||Q) = 0.7 \log \frac{0.7}{0.5} + 0.2 \log \frac{0.2}{0.3} + 0.1 \log \frac{0.1}{0.2}$$


Natural Log Function
log(x)
x=1 (log(1) = 0)
1.400
0.667
0.500
0
−1
−2
−3
−4
log(x)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
x

Cosine similarity

$$\text{dot product}(\mathbf{v},\mathbf{w}) = \mathbf{v}\cdot\mathbf{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

$$\text{cosine}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

normalized dot product

Cosine similarity (cont'd)

$$\cos(\mathbf{v}, \mathbf{w}) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}$$

The maximum value is

$$\max \cos(\mathbf{v}, \mathbf{w}) = 1,$$

and equality holds exactly when

$$\frac{v_1}{w_1} = \frac{v_2}{w_2} = \cdots = \frac{v_N}{w_N},$$

for all indices with $w_i \neq 0$.

Cross-entropy loss review

For probability distributions p, q:

$$H(p, q) = -\sum_i p_i \log q_i.$$

Gibbs' inequality states:

$$H(p, q) \geq H(p) = -\sum_i p_i \log p_i,$$

with equality if and only if $p = q$.

Cross-entropy loss (cont'd)

The predicted probabilities

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_V \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_V \end{bmatrix}$$

The ground truth label

$$y_i = \begin{cases} 1, & \text{if } i = c \text{ (correct class index)} \\ 0, & \text{otherwise} \end{cases}$$

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{V} e^{z_j}}, \quad \text{for } i = 1, 2, \ldots, V$$

Cross-entropy loss (cont'd)

$$L_{CE}(\hat{y}, y) = -\sum_{i=1}^{V} y_i \log \hat{y}_i$$

$$L_{CE}(\hat{y}, y) = -\left(y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + \cdots + y_V \log \hat{y}_V\right)$$

Since the true label y is one-hot encoded, only one term in the sum is nonzero, corresponding to the correct class c, where $y_c = 1$ and $y_i = 0$ for all $i \neq c$. This simplifies the sum to:

$$L_{CE}(\hat{y}, y) = -y_c \log \hat{y}_c$$

Since $y_c = 1$, this further reduces to:

$$L_{CE}(\hat{y}, y) = -\log \hat{y}_c$$

The loss models the distance between the system output and the gold output —lower is better

Let $p(x)$ denote the true distribution and $q(x)$ denote a model distribution.

Cross entropy

$$H(p, q) = -\sum_x p(x) \log q(x)$$

Entropy

$$H(p) = -\sum_x p(x) \log p(x)$$

KL divergence

$$D_{KL}(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

These quantities satisfy the identity:

$$H(p, q) = H(p) + D_{KL}(p\|q)$$

This decomposition shows why KL divergence is relevant.

- The entropy $H(p)$ depends only on the true distribution.
- When optimizing a model, $H(p)$ is constant.
- Therefore, minimizing cross entropy with respect to q is exactly equivalent to minimizing KL divergence $D_{KL}(p\|q)$.

In supervised learning, the empirical distribution plays the role of p. Cross entropy loss, which is used for classification, therefore minimizes the KL divergence between the empirical data distribution and the model distribution.

The entropy measures how uncertain, unpredictable, or spread out a distribution is.

Intuition

Entropy is the expected amount of information in a draw from the distribution.

- If one outcome has probability 1, then there is no uncertainty. The entropy is 0.
- If all outcomes are equally likely, uncertainty is maximal. The entropy is large.
- If probability mass is concentrated on a few outcomes, entropy is lower.

In other words, entropy quantifies how surprised you are on average when you observe a sample.

Cross entropy differs from KL divergence only by a constant that does not depend on the model. That is why minimizing cross entropy is the same objective as minimizing KL divergence.

Step 3: RL fine-tuning (cont'd)

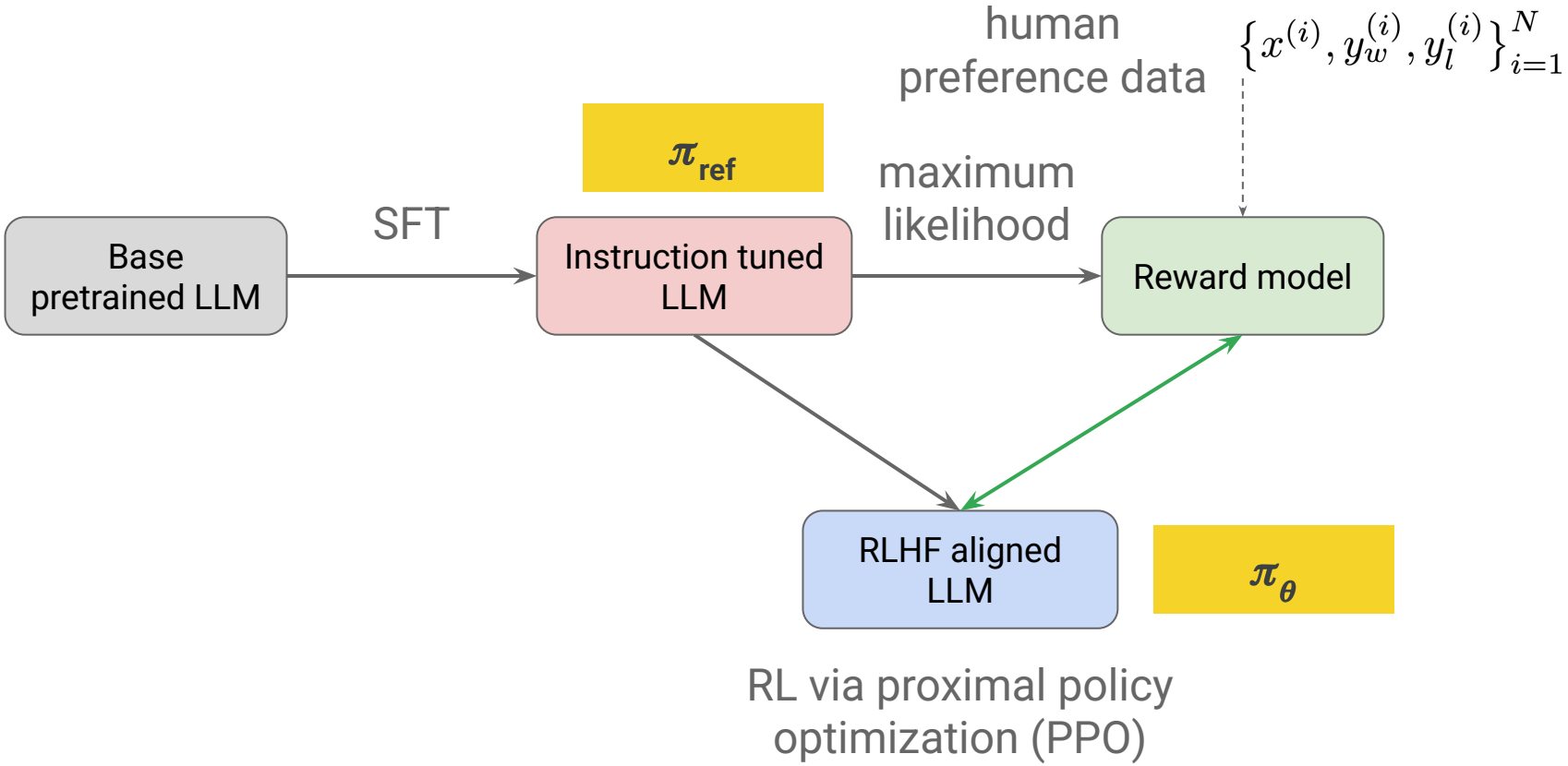
The sequence reward is distributed across tokens. PPO (Proximal Policy Optimization) updates happen at the token level.

y = π_θ(x)

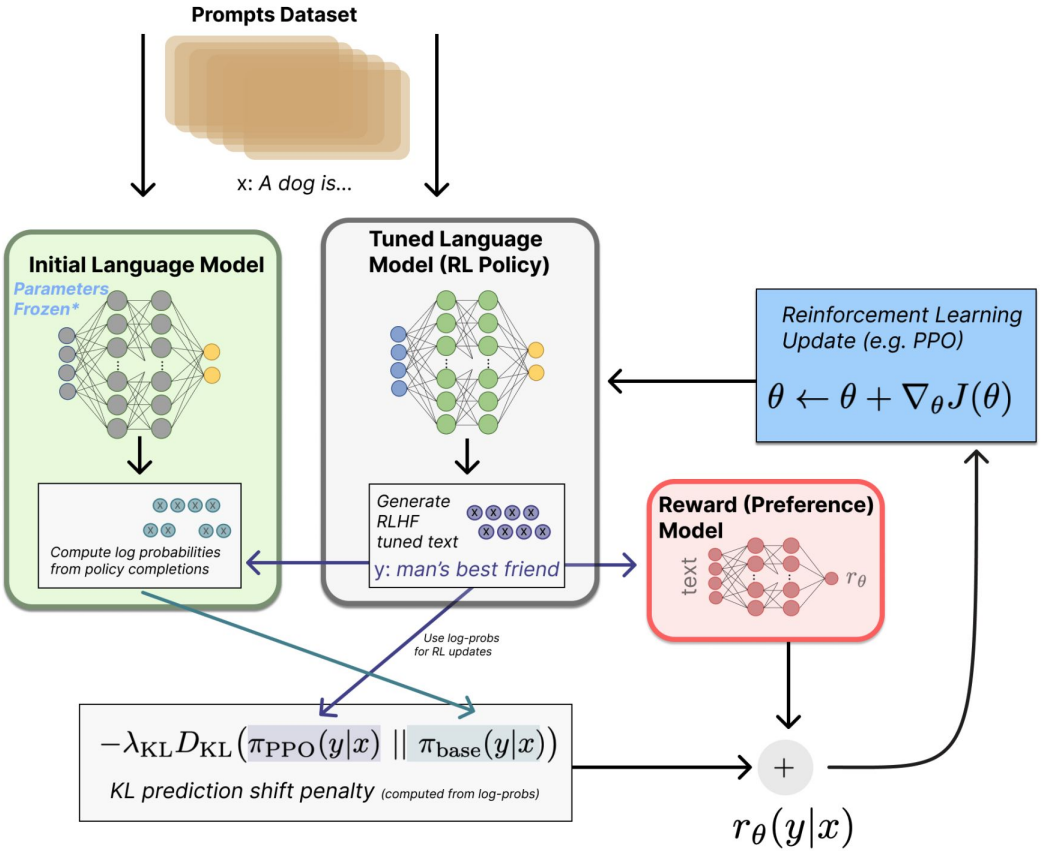
$$\max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)} \left[r_\phi(x,y)\right] - \beta D_{KL}\left[\pi_\theta(y|x) \,||\, \pi_{\text{ref}}(y|x)\right]$$

where β is a parameter controlling the deviation from the base reference policy π_{ref}, namely the initial SFT model π^{SFT}. In practice, the language model policy π_θ is also initialized to π^{SFT}.

RLHF pipeline: putting it all together



RLHF pipeline: putting it all together



The effects of RLHF on LLM generalization & diversity

SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training

Tianzhe Chu * Yuexiang Zhai * Jihan Yang Shengbang Tong
Saining Xie Dale Schuurmans Quoc V. Le Sergey Levine Yi Ma

Abstract

Supervised fine-tuning (SFT) and reinforcement learning (RL) are widely used post-training techniques for foundation models. However, their respective role in enhancing model generalization in rule-based reasoning tasks remains unclear. This paper studies the comparative effect of SFT and RL on generalization and memorization, focusing on text-based and visual reason-

1. Introduction

Although SFT and RL are both widely used for foundation model training (OpenAI, 2023b; Google, 2023; Jaech et al., 2024; DeepSeekAI et al., 2025), their distinct effects on *generalization* (Bousquet & Elisseeff, 2000; Zhang et al., 2021) remain unclear, making it challenging to build reliable and robust AI systems. A key challenge in analyzing the generalizability of foundation models (Bommasani et al., 2021; Brown et al., 2020) is to separate data mem-

UNDERSTANDING THE EFFECTS OF RLHF ON LLM GENERALISATION AND DIVERSITY

Robert Kirk* [α] **Ishita Mediratta** [β] **Christoforos Nalmpantis** [β] **Jelena Luketina** [γ]
Eric Hambro [β] **Edward Grefenstette** [α] **Roberta Raileanu** [β]

[α] University College London, [β] Meta, [γ] University of Oxford

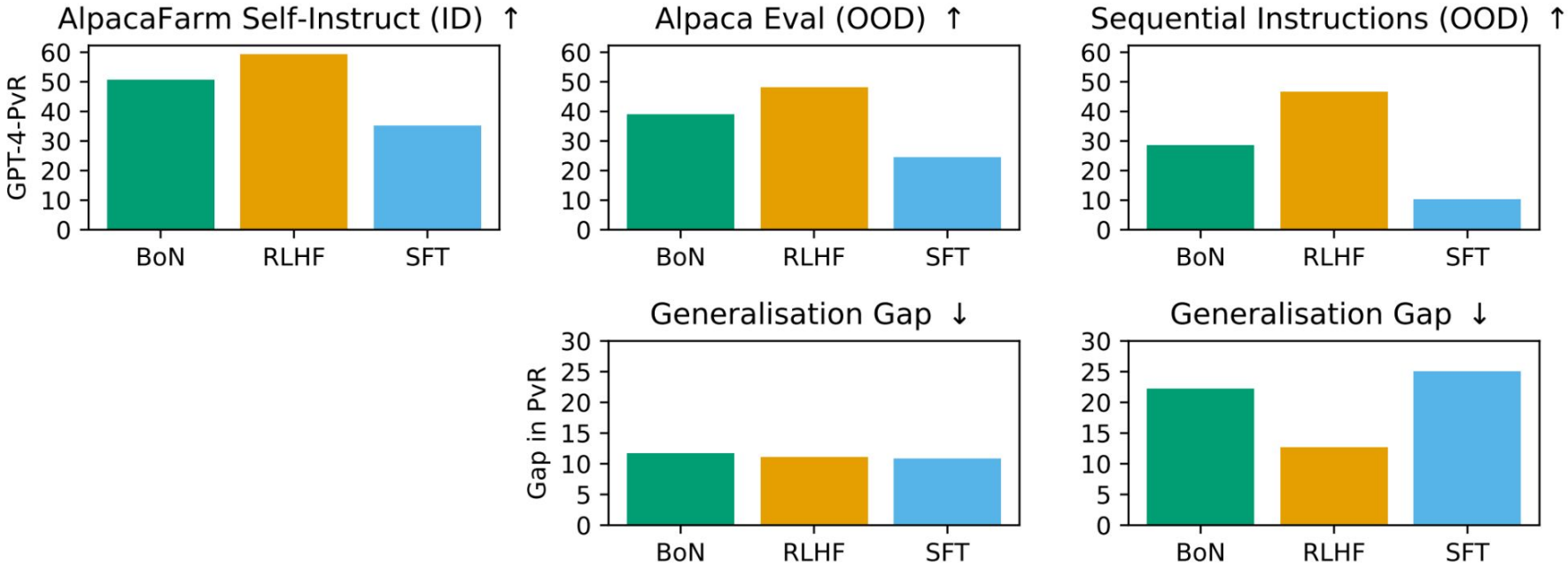


Figure 3: **Instruction Following Generalisation Results.** GPT-4 PvR for SFT, BoN and RL policies, based on LLaMa 7B, trained on the AlpacaFarm Self-Instruct instruction following task. ID is on AlpacaFarm Self-Instruct, OOD is on the AlpacaEval and Sequential Instructions datasets respectively, and generalisation gap is ID – OOD performance.

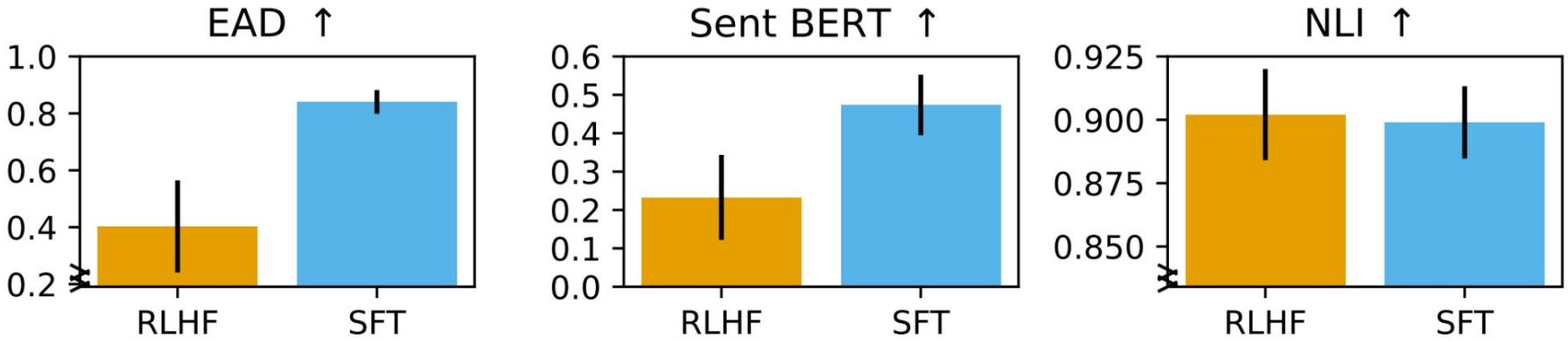


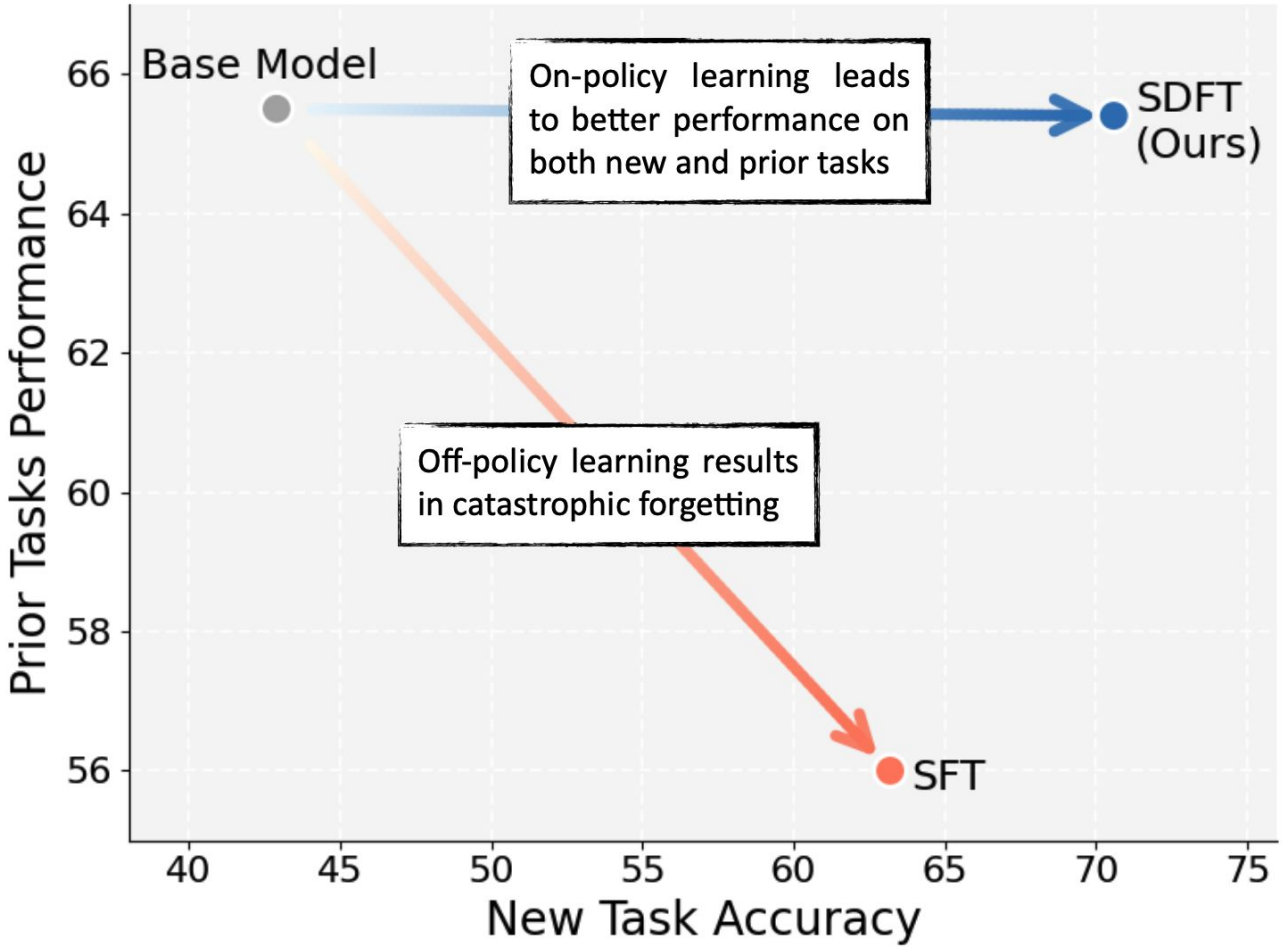
Figure 5: **Per-input diversity metrics for RLHF and SFT models**. For these scores the outputs used to calculate the diversity are a sample of outputs from the model for single input. These per-input scores are then averaged, as in Eq. (2). Error bars are standard deviation of the per-input diversity score across different inputs. Note that some plots have broken y-axis for better visualisation.

SELF-DISTILLATION ENABLES CONTINUAL LEARNING

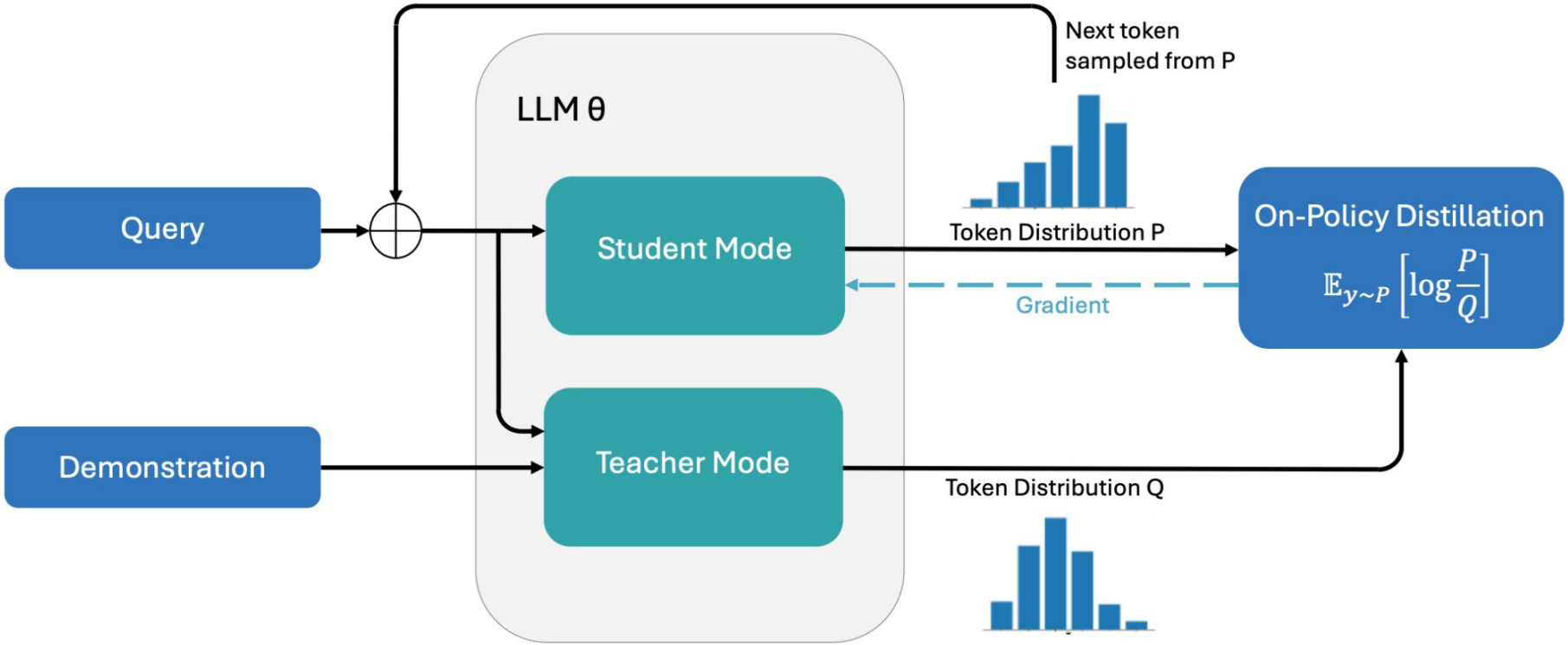
Idan Shenfeld[1 2*] **Mehul Damani**[1] **Jonas Hübotter**[3] **Pulkit Agrawal**[1 2]
[1]MIT [2]Improbable AI Lab [3]ETH Zurich

ABSTRACT

Continual learning, enabling models to acquire new skills and knowledge without degrading existing capabilities, remains a fundamental challenge for foundation models. While on-policy reinforcement learning can reduce forgetting, it requires explicit reward functions that are often unavailable. Learning from expert demonstrations, the primary alternative, is dominated by supervised fine-tuning (SFT), which is inherently off-policy. We introduce **Self-Distillation Fine-Tuning (SDFT)**, a simple method that enables on-policy learning directly from demonstrations. SDFT leverages in-context learning by using a demonstration-conditioned model as its own teacher, generating on-policy training signals that preserve prior capabilities while acquiring new skills. Across skill learning and knowledge acquisition tasks, SDFT consistently outperforms SFT, achieving higher new-task accuracy while substantially reducing catastrophic forgetting. In sequential learning experiments, SDFT enables a single model to accumulate multiple skills over time without performance regression, establishing on-policy distillation as a practical path to continual learning from demonstrations. Code and Datasets are available at `http://idanshenfeld.com/SDFT`.


Base Model
On-policy learning leads to better performance on both new and prior tasks
SDFT (Ours)
Off-policy learning results in catastrophic forgetting
SFT
Prior Tasks Performance
66
64
62
60
58
56
40
45
50
55
60
65
70
75
New Task Accuracy

Self-Distillation Fine-Tuning



Let us call:

- $p(x)$ = teacher distribution
- $q(x)$ = student distribution

There are two common ways to match them:

- **Forward KL**: $\mathrm{KL}(p\|q)$
- **Reverse KL**: $\mathrm{KL}(q\|p)$

They behave very differently.

1. Forward KL: mode covering

$$\mathrm{KL}(p\|q) = \mathbb{E}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right]$$

You sample from the **teacher**, then ask:

Can you imitate the teacher everywhere?

2. Reverse KL: mode seeking

$$\mathrm{KL}(q\|p) = \mathbb{E}_{x\sim q}\left[\log \frac{q(x)}{p(x)}\right]$$

You sample from the **student**, then ask:

Where you already speak, can you adjust slightly?

The student sticks to its own peaks instead of spreading probability across all the teacher's peaks

Thank you!