Post-training: Next-token prediction to following Instructions

Ayush Maheshwari

Sr. Solutions Architect, NVIDIA

Interested in research collaboration !

Fill this form - https://tinyurl.com/collab-nv

Sessions

1. Cluster health-check using NCCL, MLPerf, HPL **(1 hour) - Completed**
   a) Understand the hardware and its performance on multiple GPUs.
   b) Ensure that your training performance aligns with the h/w benchmarks
   c) Evaluate the cluster to ensure platform fits within your needs.
2. Large scale data curation for LLM training **(1.5 hour) - Completed**
   a) Deep-dive into aspects of data curation
   b) Mixed-precision training
3. Distributed and stable LLM training on a large-scale cluster **(1.5 hour) - Completed**
   a) Parallelism techniques
   b) Frameworks and wrappers
   c) Recipes and best practices
4. Fine-tuning and deployment **(1.5 hour) - Completed**
   a) Dynamic and static batching, state management, inference server
   b) Best practices for optimizing model
5. Post-training and evaluation of pre-trained LLM **(1.5 hour) - Today**
   a) Instruction following using SFT
   b) RLHF to align model output

Agenda

1. Next-token prediction to instruction following

2. Supervised Fine-Tuning (SFT) fundamentals, including

dataset creation

3. Various Reinforcement Learning techniques

4. Evaluation methodologies for assessing model quality

Introduction to LLM Post-training

- Pre-trained LLMs are knowledgeable but not user-aligned
- Post-training bridges the gap between generic prediction and application utility
- Models shift from prediction machines to capable assistants

Pre-trained vs SFT

Llama3-8B


```
[{'generated_text': 'Who is The President of India ? The Preside
nt of India is the head of state and first citizen of India. The
 President of India is the first citizen of India and is the hea
d of state. The President is the constitutional head of the exec
utive of the Government of India. The President is also the comm
ander-in-chief of the Indian Armed Forces. The President is elec
ted by the elected members of both the houses of the Indian Parl
```


GPT-OSS-20B


```
{'role': 'assistant', 'content': 'analysisThe user asks: "Who is
 The President of India?" They likely want the current president
. As of 2025, the President of India is Draupadi Murmu? Wait, le
t\'s recall: The current President of India as of 2023 is Draupa
di Murmu (since July 25, 2022). She is the 15th President. So an
swer: Draupadi Murmu. The question: "Who is The President of Ind
ia?" Should answer: Draupadi Murmu, 15th President of India, in
office since July 25, 2022. Provide some context. Also note that
 the question might be generic but we give current. Also mention
 that the President is the head of state. Provide some details.
Provide answer.\n\nWe should be careful: The question might be a
sked in context of 2025. The President hasn\'t changed. So answe
r accordingly.assistantfinal**Dr.\u202fDraupadi Murmu**\u202fis
the current President of India. She assumed office on **25\u202f
July\u202f2022** as the 15th President of the Republic of India
and is the first person from a tribal community to hold the post
.'}
```


- model is completing the sentence and adding other common internet metadata.

Intuition for Post-Training

- Belief that scaling data is important to performance.
- Model knowledge and capabilities are learnt almost entirely during pretraining
- Alignment teaches it which subdistribution of formats should be used when interacting with users
- Few thousand samples for instruction finetuning can change a model substantially

Typical SFT-RLHF recipe

Src: https://rlhfbook.com/c/01-introduction

NVIDIA

Instruction Tuning

Supervised Fine-Tuning

- Instruction tuning as the foundation of post-training for building helpful models across tasks and domains.
- In narrow domains like chat alignment (excluding harder skills like math/code), small focused datasets can perform strongly.
- Early SOTA used small human datasets (~10K, e.g., No Robots); current practice favors large-scale synthetic datasets for most tasks.
- Prioritize high-quality completions; models learn from outputs, while prompts are often masked and not predicted.
- Around ~1M prompts can yield models that are excellent bases for RLHF and further post-training; scaling beyond shows diminishing returns.

Differences from pretraining

- **Prompt masking**: compute loss only on completion tokens so the model learns to produce responses, not to predict the user's queries.
- **Multi-turn masking**: in dialogues, include prior turns as masked context and train loss only on the final assistant turn for that sample.
- **Loss Function:** Maintain the same loss function as pretraining (autoregressive), with masking controlling which tokens contribute to the loss.

$$\mathcal{L}_{SFT}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}} \sum_{t=1}^{T} \log p_\theta(y_t | x, y_{<t})$$

where,

- θ are the model parameters
- y_t is the t-th token in the target response
- $y_{<t}$ is the prefix (previous tokens)
- $p_\theta(\cdot)$ is the probability distribution from the model

Few notes on SFT

1. This is the exact same loss-function used in the pre-training phase

2. **End of Message Special Token**: The model will indefinitely continue producing responses, while the loss is clipped at `max_seq_len = T.

3. **Overfitting risk & Catastrophic forgetting**: Since SFT datasets are typically much smaller than pre-training datasets, models can easily overfit.

4. **Curriculum design**: The structure of training data significantly impacts performance.

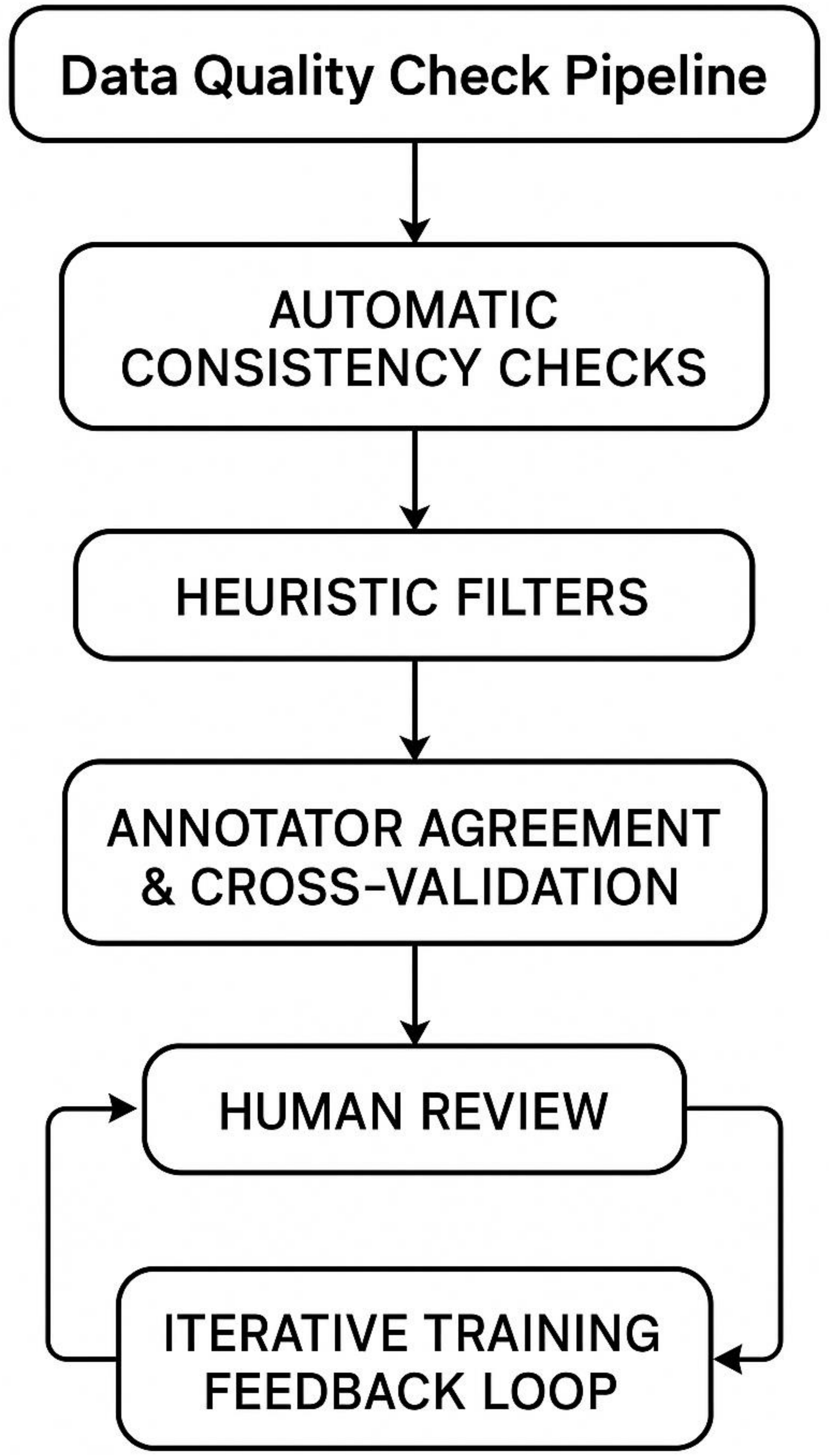
Data Examples

What is the capital of India

The capital of India is New Delhi


```
{
   "prompt":[
      {"role": "system", "content":"You are a helpful, honest assistant."},
      {"role":"user", "content":"What is the capital of India"},
   ],
   "completion":[
       {"role": "assistant","content":"The capital of India is New Delhi
"}
    ],
}
```

Data Quality



SFT vs RLHF

Example

What is a transformer language model?

Llama 3.1 Tulu 3 70B

SFT

Copy Reroll

allenai/Llama-3.1-Tulu-3-70B-SFT

Transformer models are a type of neural network architecture that have been developed for natural language understanding tasks, such as language translation, or sentence completion, and later used for vision, and other tasks.

They were originally only used for tasks that require understanding text, like text translation or text summarization, but recently, there has been a lot of research into using transformers for other machine learning tasks.

Here are some of the ways that transformer models have changed natural language understanding:

Improved accuracy: Transformers have achieved state-of-the-art results on many natural language understanding tasks, such as text classification, machine translation, and text generation. This is due to their ability to understand the context of

RLHF

Copy Reroll

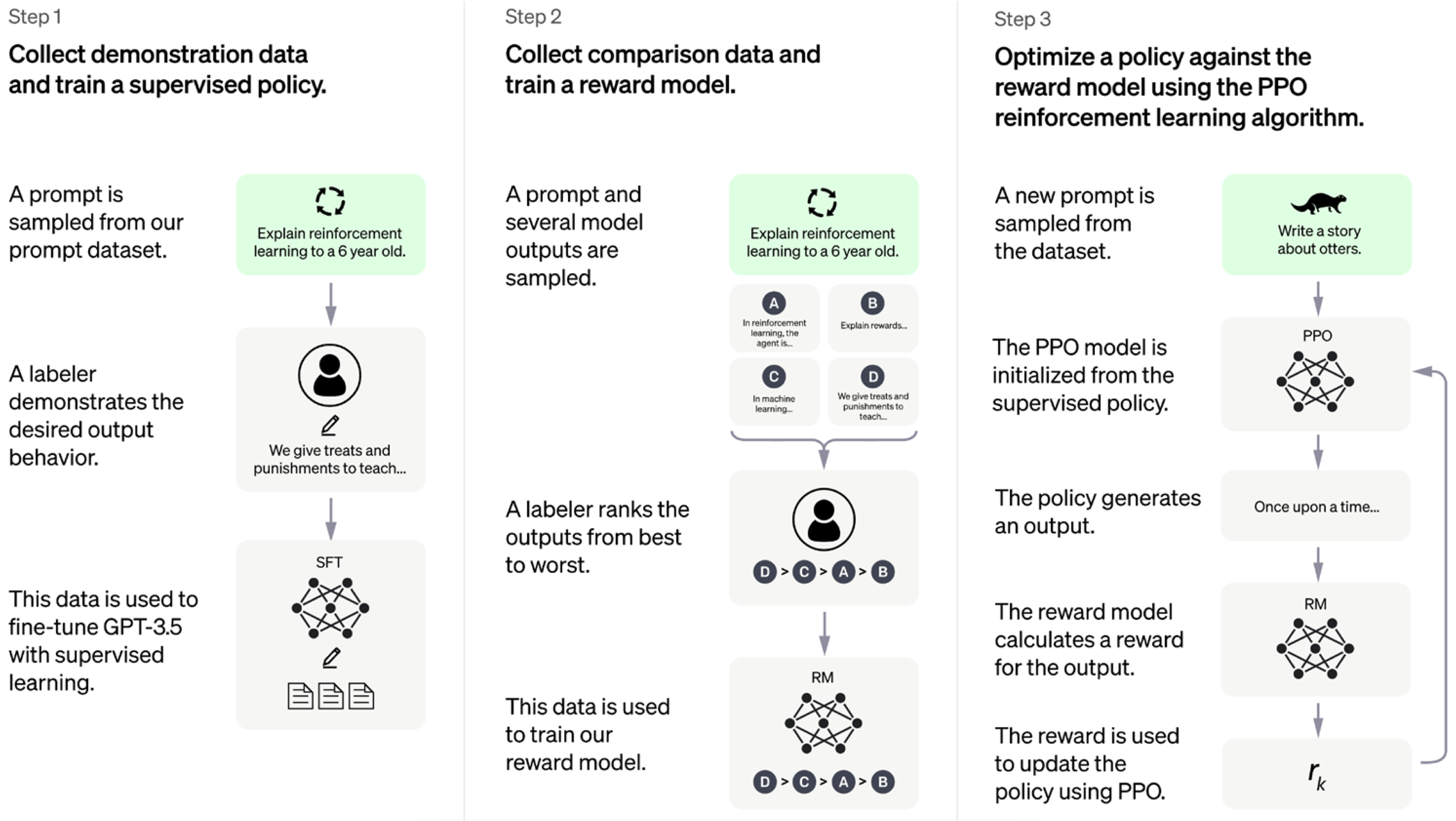
allenai/Llama-3.1-Tulu-3-70B-DPO

A transformer language model is a neural network architecture designed specifically for natural language processing (NLP) tasks, based on the transformer model introduced by Vaswani et al. in their 2017 paper "Attention is All You Need." Unlike traditional recurrent neural networks (RNNs) or long short-term memory networks (LSTM), transformer models utilize the "self-attention" mechanism to weigh the importance of different words in a sentence context without requiring sequential data processing.

The transformer architecture is especially known for its scalability and efficiency, allowing for the training of very large models with billions of parameters. These models excel at various NLP tasks such as:

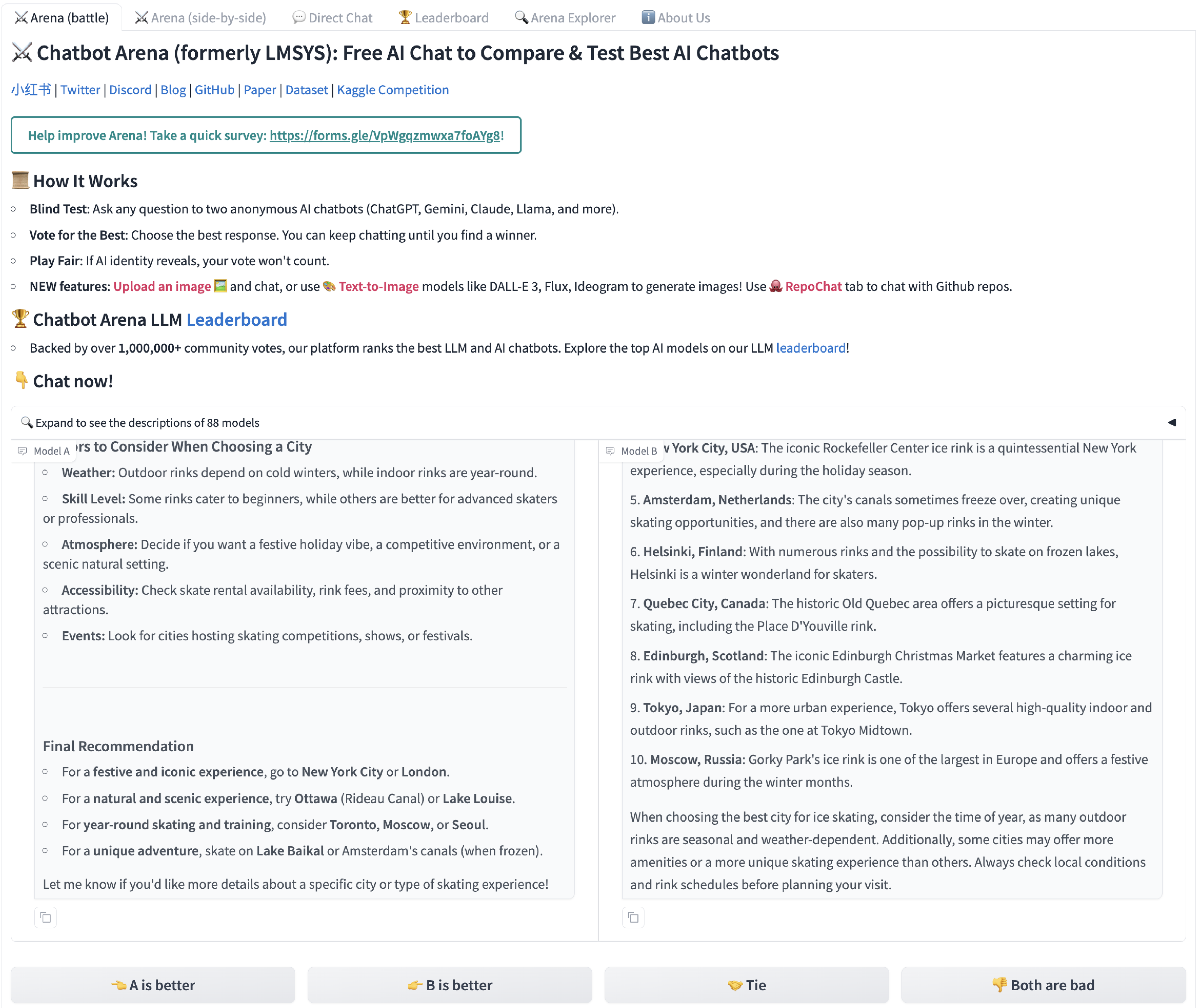
1. **Language Translation**: Transformer models can effectively translate between

Going from InstructGPT to ChatGPT



Preference Data

Reinforcement Learning from Human Feedback



Preference Data

Reinforcement Learning from Human Feedback

6. **Montreal, Canada** - Known for its cold winters, Montreal offers several outdoor rinks, including the one at the Old Port, and is a city with a strong skating culture.
7. **Helsinki, Finland** - With its cold climate, Helsinki provides numerous ice skating opportunities, including the Ice Park in the city center.
8. **Amsterdam, Netherlands** - When the canals freeze over, which is rare but magical, locals and tourists alike take to the ice for a unique experience.
9. **Edmonton, Canada** - Home to the Edmonton Iceway and numerous community rinks, it's a city where ice skating is a beloved pastime.
10. **Sapporo, Japan** - Known for its annual Snow Festival, the city also offers excellent ice skating opportunities in its various parks and facilities.

These cities offer a mix of natural and artificial ice rinks, cultural experiences, and scenic beauty that can make ice skating a memorable activity. The best choice depends on what you value most in an ice skating experience.

Good response

Reply to Tülu

Always fact-check your results.

Training a Reward Model

Reward is a scalar value

- Measures the probability of pairwise comparison for two events drawn from the same distribution.

$$P(i > j) = \frac{p_i}{p_i + p_j}$$

- For two completions, y_1 and y_2 for a given prompt:

$$P(y_1 > y_2) = \frac{\exp(r(y_1))}{\exp(r(y_1)) + \exp(r(y_2))}$$

- Start with an SFT model (policy) π and a frozen reference π_0 (usually the initial SFT model)

- small linear head to the language model that performs classification between two outcomes – chosen and rejected

```
import torch.nn as nn
rewards_chosen = model(**inputs_chosen)
rewards_rejected = model(**inputs_rejected)

loss = -nn.functional.logsigmoid(rewards_chosen - rewards_rejected).mean()
```

Note, when training reward models, the most common practice is to train for only 1 epoch to avoid overfitting.

Different Types of Rewards

Reward family	What the reward is	Where labels come from	Typical tasks
RLHF (human preferences)	Scalar from a pairwise **reward model (RM)** + KL to reference	Human comparisons on sampled outputs	General chat, safety/style alignment
RLAIF / Constitutional-AI	RM trained on **LLM-judge** comparisons (guided by a constitution/rubric) + KL	AI-judged comparisons with light human spot-checks	Scalable helpfulness/harmlessness tuning
RLVR (verifiable outcomes)	**Programmatic** rewards (exact match, unit tests, execution) + KL	Ground-truth answers, tests, deterministic checkers	Math, code, factual QA
Process-supervised RL (PRMs)	**Step-level** rewards from a Process RM scoring CoT steps + KL	Human/AI step labels, heuristics, execution traces	Long-form reasoning, tool use
Rubric-guided rewards	A rubric converted to a scalar via **LLM-judge scores, PRM step scores**, or **executable checks** + KL	LLM-judge rubric scoring; step annotations aligned to rubric; programmatic validators from rubric	Helpfulness/safety audits, style/format compliance, math/code if rubric is executable

Src: https://tokens-for-thoughts.notion.site/post-training-101

Rubrics-guided rewards

User prompt:

Explain what a VPN is to a 10-year-old.

JSON Format:

```
{
  "prompt": [
    {"system":"You are a helpful, honest assistant."},
    {"role": "user", "content": "Explain what a VPN is to a 10-year-old."},
  ],
  "metadata": [
    "rubric": {
    "scale": {"min": 1, "max": 7},
    "criteria": [
      {
        "id": "factuality", "weight": 0.5,
        "definition": "Correct, non-misleading statements.",
        "guidelines": ["No made-up facts", "Cite if uncertain", "No contradictions"]
      },
      {
        "id": "helpfulness", "weight": 0.3,
        "definition": "Directly answers the user's ask; useful context/examples.",
        "guidelines": ["Addresses age/intent", "Actionable and clear"]
      },
      {
        "id": "concision", "weight": 0.2,
        "definition": "No fluff; tight phrasing; avoids repetition.",
        "guidelines": ["Prefer short sentences", "Remove meta-chatter"]
      }
    ],
    "hard_rules": [
      {"if": "safety < 4", "then": "overall = 0", "reason": "Unsafe content gates overall
    ],
    "aggregate": "overall = 0.5*factuality + 0.3*helpfulness + 0.2*concision",
    "notes": "Weights sum to 1. Safety is a gating criterion (not in the sum)."
    },
    "judge": {
      "model_id": "gpt-4o-2025-06-01", "temperature": 0.0, "seed": 17,
      "prompt_template": "judge_v3"
    },
  ],
}
```

Src: https://tokens-for-thoughts.notion.site/post-training-101

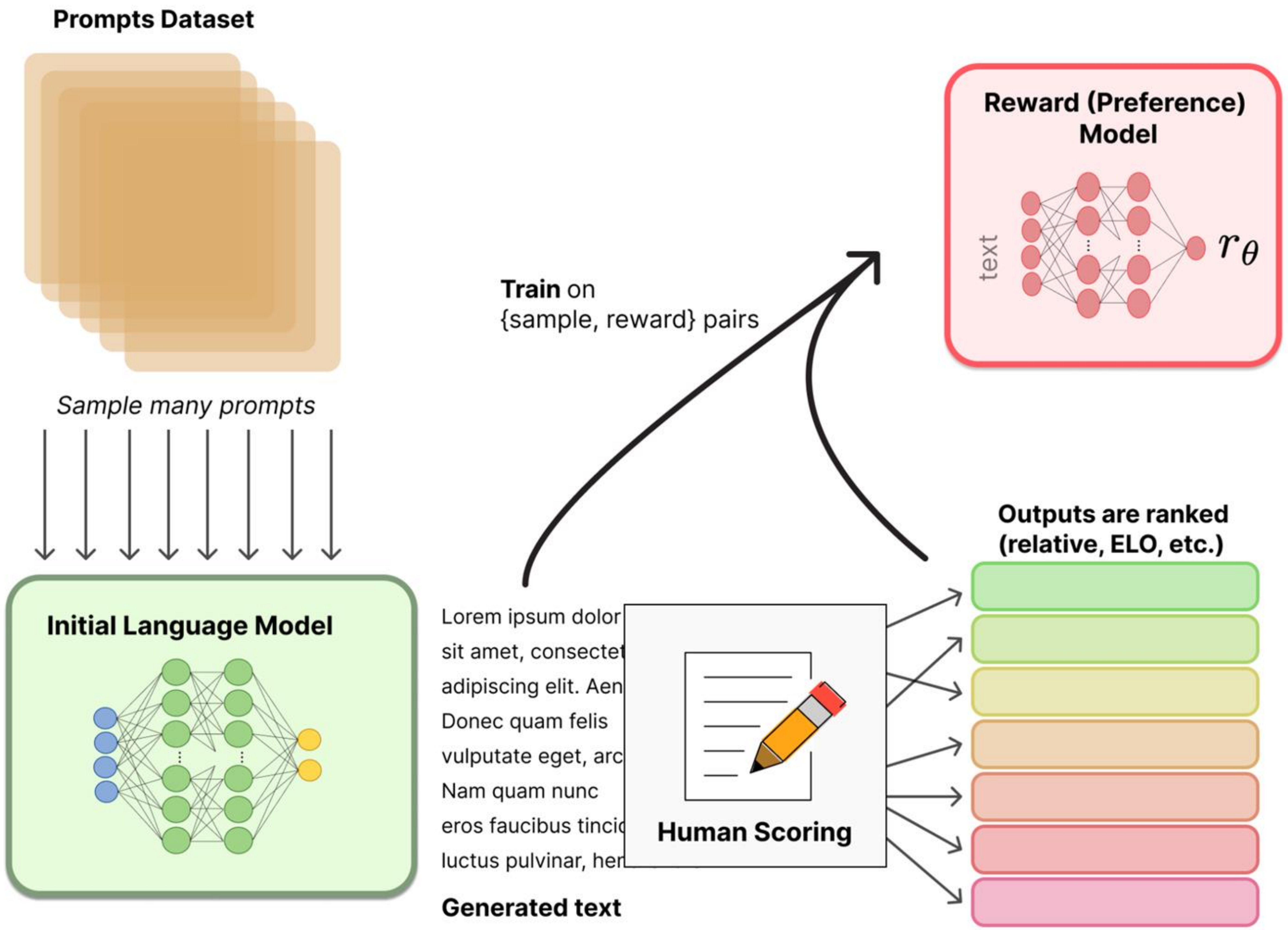
How does RLHF work: 1 – Preference Modeling

Let's look at the first part, creating a Preference model.

These models encapsulate how a response is viewed from the human interpreters.

Many hundreds of prompts are sampled and given to human scorers to balance metrics like:

- Length of response
- Depth and breadth of the content
- Truthfulness
- Harmfulness
- How well the response is aligned with the initial prompt
- …



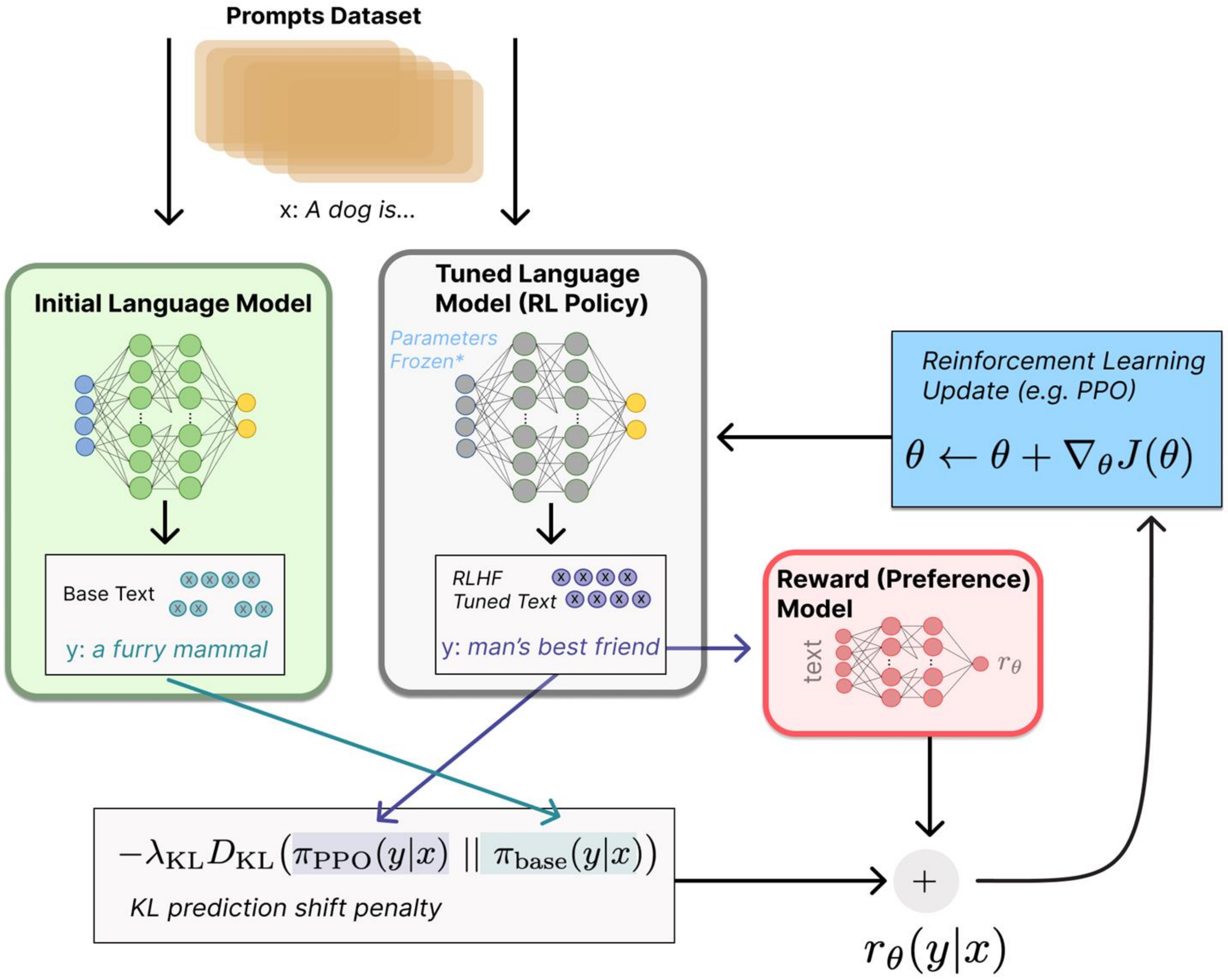
How does RLHF work: 2 – Policy Optimization

Once the reward model has been trained with sufficient data from the human reviewers, Proximal Policy Optimization is used to update the LLM using the outputs and the reward/preference model to provide the learning gradient.

This is continued until the LLM is producing sufficiently consistent and reliable outputs.

Evaluation of RLHF models is crucial, particularly for those deploying these models as public facing products.

$$\max_{\pi} \mathbb{E}_{y\sim\pi(\cdot|x)}\left[r(x,y)\right] - \beta\,\mathrm{KL}(\pi(\cdot \mid x) \,\|\, \pi_0(\cdot \mid x))$$



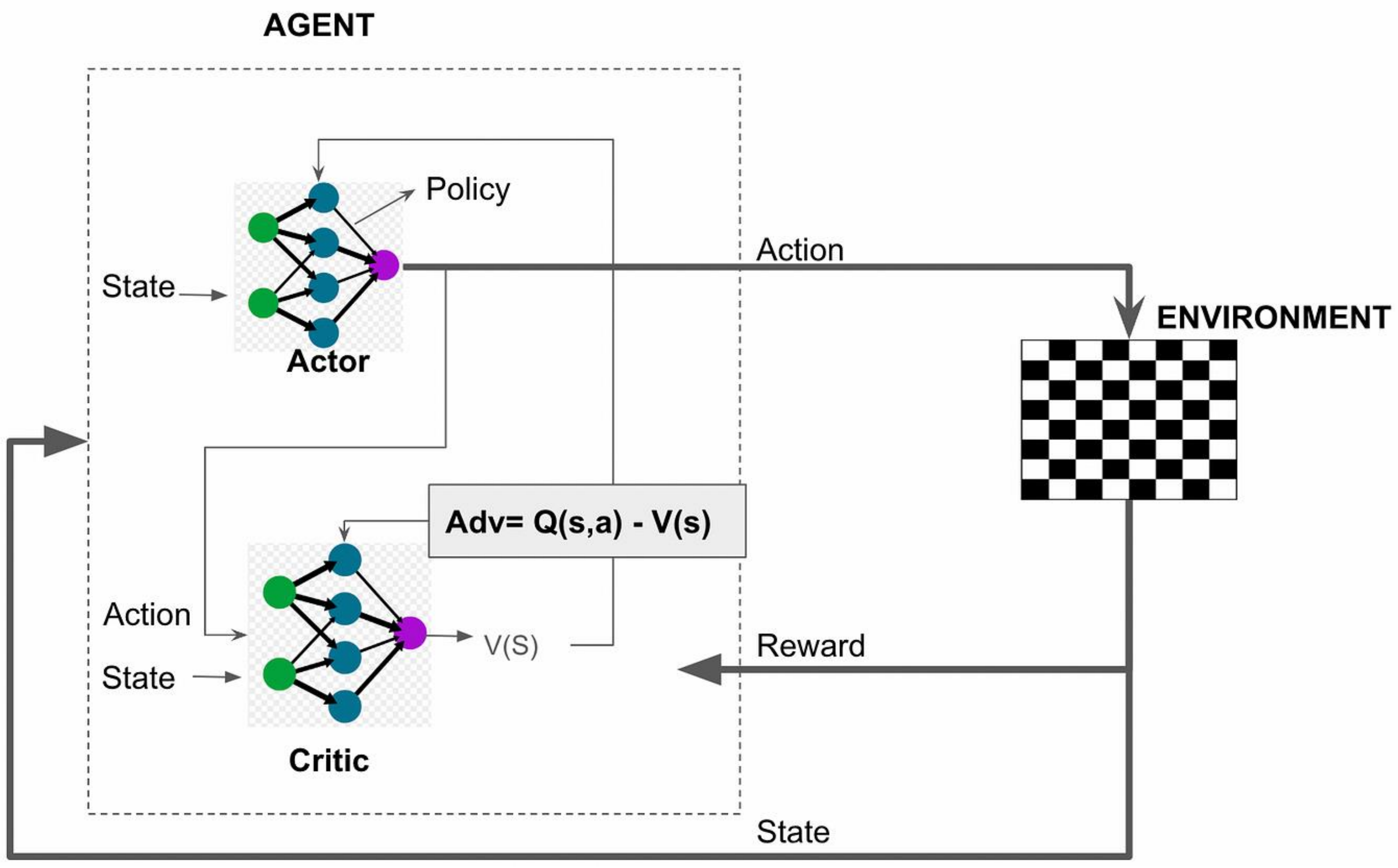
How PPO Training works

Step 1: Rollouts Collection
The agent interacts with the environment using its current policy to generate **trajectories** or **rollouts**. These rollouts contain state, action, reward, and next state information, which is stored for training.

Step 2: Advantage Estimation
Calculate the **advantage** function using the rewards and the value network. The advantage tells the agent how much better or worse the taken action was compared to the expected value of the state, helping guide future actions.

Step 3: Policy Update (Clipping Mechanism)
Update the policy by **maximizing the advantage** while using a **clipping mechanism** to prevent large, unstable updates. This keeps the new policy close to the old one, ensuring stability in learning.

Step 4: Value Network Update
Simultaneously update the **value network** by minimizing the difference between predicted values and actual returns. This ensures the value network provides accurate feedback for future policy updates.



Src: https://miro.medium.com/v2/resize:fit:1358/1*CvqcmE569aBSfo4W5tu22g.png

GRPO

Grouped Relative Policy Optimization

- Eliminates the need for a separate critic model (typically as large as the policy model), reducing memory and compute overhead by ~50%

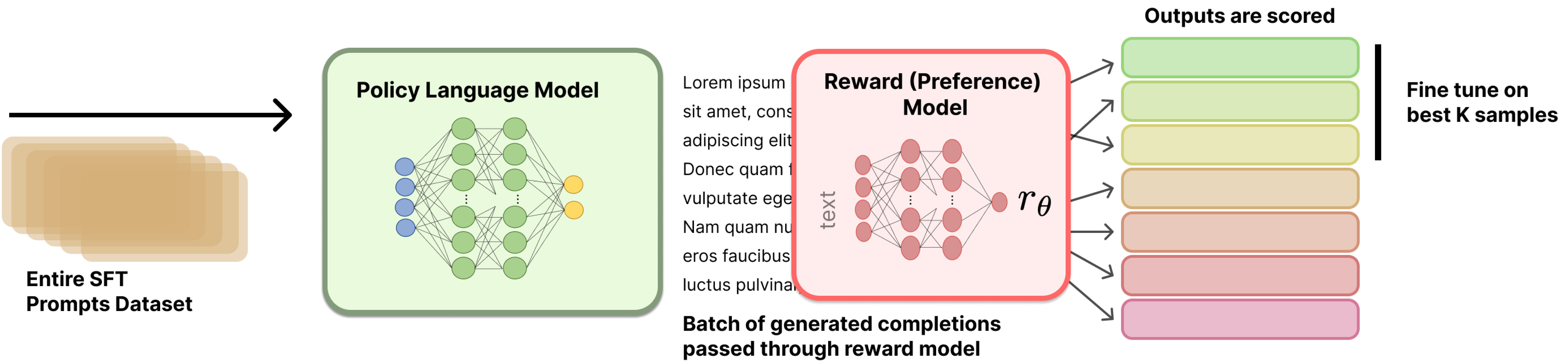
- For a prompt x, sample K responses, from the current policy.

- Score them with your reward source (RM, verifier, judge).

- Define a **group baseline**

Rejection Sampling

Popular way to perform preference Fine-tuning



Evaluate Post-trained Models

Automatic Evaluation

- Automatic Evaluation
  - Ground Truth Based Eval
  - LLM-Judge Based Eval

Example Completion Graded as Correct:

- Each car has 4 tires.
- With 15 cars: 15×4=6015 \times 4 = 6015×4=60.

Final Answer: \boxed{60}

Example Completion Graded as Incorrect:

- Each car has 4 tires.
- With 15 cars: 15×4=6015 \times 4 = 6015×4=60.

Final Answer: \boxed{55}

Prompt:

Summarize the following passage in one sentence:

"The Amazon rainforest, often referred to as the lungs of the planet, produces 20% of the world's oxygen and is home to an incredible diversity of species. However, deforestation driven by agriculture and logging poses a severe threat to its survival."

Rubrics:

* The response must summarize the main parts
* The response should contain exactly one sentence
* The response should not make up additional facts/information not in the original paragraph

Verifier:

Python Copy

```python
You are an evaluator judging candidate responses.

TASK PROMPT:
{prompt}

CANDIDATE RESPONSE:
{response}

RUBRIC:
- Helpfulness (1–7): Does the summary capture the main facts?
- Factuality (1–7): Is the information correct?
- Clarity (1–7): Is the response clear and concise?

Return JSON:
{
  "helpfulness": int,
  "factuality": int,
  "clarity": int,
  "composite": float,
  "rationale": "short explanation"
}
```

Src: https://tokens-for-thoughts.notion.site/post-training-101

Evaluate Post-trained Models

Human Evaluation

- Human Evaluation
  - Point-wise Eval
  - Preference Based Eval

- **Pairwise preference**: Annotators pick the better response between two candidates.
- **Likert-scale ratings**: Raters score a response on a 1–5 or 1–7 scale for attributes like helpfulness or safety.
- **Expert evaluations**: Domain experts assess correctness in specialized areas (e.g., medicine, law, finance).
- **User studies & UXR**: Live experiments with real users measuring satisfaction, trust, or usability.

Evaluate Post-trained Models

Human Evaluation

- Human Evaluation
    - Point-wise Eval
    - Preference Based Eval

Prompt:

Write a short explanation of why the sky appears blue.

Human Annotation guidelines:

- Read both responses carefully.
- Choose which one better answers the prompt, considering **factual accuracy, clarity, and helpfulness.**
- If both are equally good or equally bad, mark as a tie.

Response:

The sky looks blue because air molecules scatter sunlight, and shorter blue wavelengths scatter more strongly than longer red wavelengths. This effect, called Rayleigh scattering, makes the sky appear blue to our eyes.

Sample Human Rating:

- Helpfulness: 7
- Factuality: 7
- Clarity: 7
- **Composite Score:** 7.0
- **Rationale:** *"Accurate, concise, and clearly explains Rayleigh scattering."*

Prompt:

Write a short explanation of why the sky appears blue.

Human Annotation guidelines:

- Read both responses carefully.
- Choose which one better answers the prompt, considering **factual accuracy, clarity, and helpfulness.**
- If both are equally good or equally bad, mark as a tie.

Response A:

The Amazon rainforest, which produces 20% of the world's oxygen and hosts immense biodiversity, is under severe threat from deforestation.

Response B:

The Amazon rainforest is a cool place with lots of animals and trees.

Sample Human Rating:

Rater A

Response A is better (correct explanation of Rayleigh scattering).

Rater B

Response A is better (Response B contains a common misconception).

Rater C

Response A is better.

Final Rating (Majority vote):

Response A wins (3–0)

Src: https://tokens-for-thoughts.notion.site/post-training-101