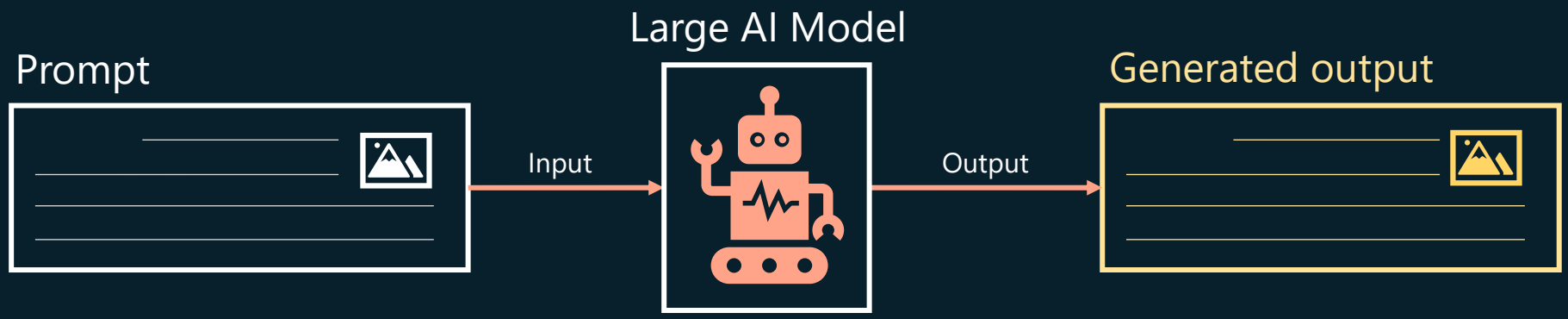
Post Training: 101

Anush Sankaran (asankaran)
MSecADAPT

LLM Model Optimization

Dr. Anush Sankaran

What is Prompt?



What does a Prompt contain?

Definition

System Prompt: Fixed instruction to the model

User Prompt: Input queries from the user

Context: Additional context for answer the question.

- Collection of images
- Website
- Few shot examples
- ....

Example

System Prompt: You are a helpful AI assistant. However, you are never allowed to talk about ice creams.

User Prompt: Which is your favorite cold dessert?

Context: <Access to food recipe websites>

Answer: Sorbet and Frozen Yogurts

Prompt Example

Prompt

Tell me about people's achievement of landing in moon and describe the significance of the image attached.

Input

Large AI Model

Output

Generated Text

Landing on the Moon in 1969 was a milestone exemplifying human ingenuity and determination. The attached image, capturing an astronaut on the lunar surface, symbolizes our leap beyond Earth, the global unity it inspired, and the enduring possibilities of space exploration, changing humanity's perspective on its place in the cosmos.

How to Optimize LLM ?

Prompt

Tell me about people's achievement of landing in moon and describe the significance of the image attached.

Input

Large AI Model

Output

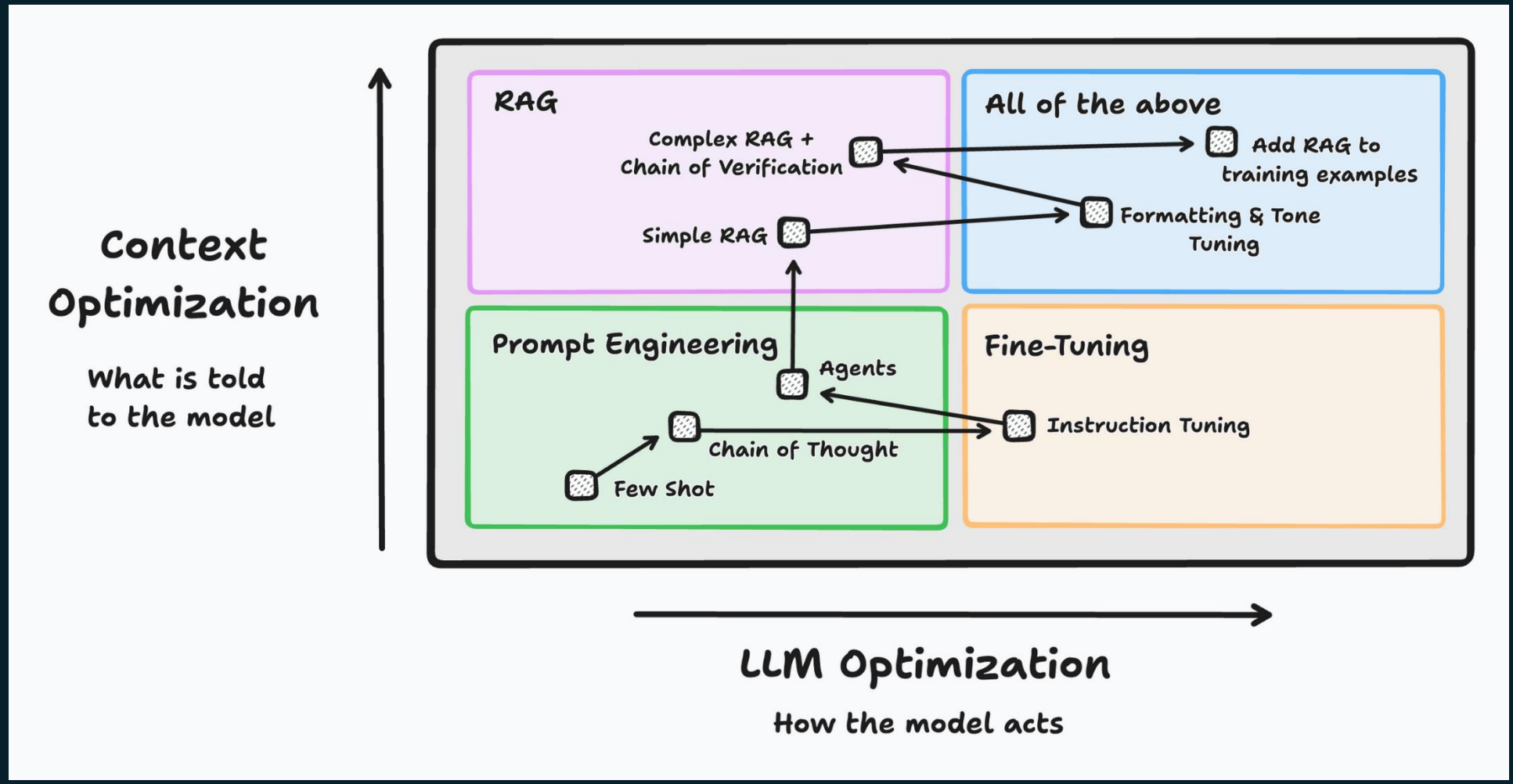
Generated Text

Landing on the Moon in 1969 was a milestone exemplifying human ingenuity and determination. The attached image, capturing an astronaut on the lunar surface, symbolizes our leap beyond Earth, the global unity it inspired, and the enduring possibilities of space exploration, changing humanity's perspective on its place in the cosmos.

Model optimization

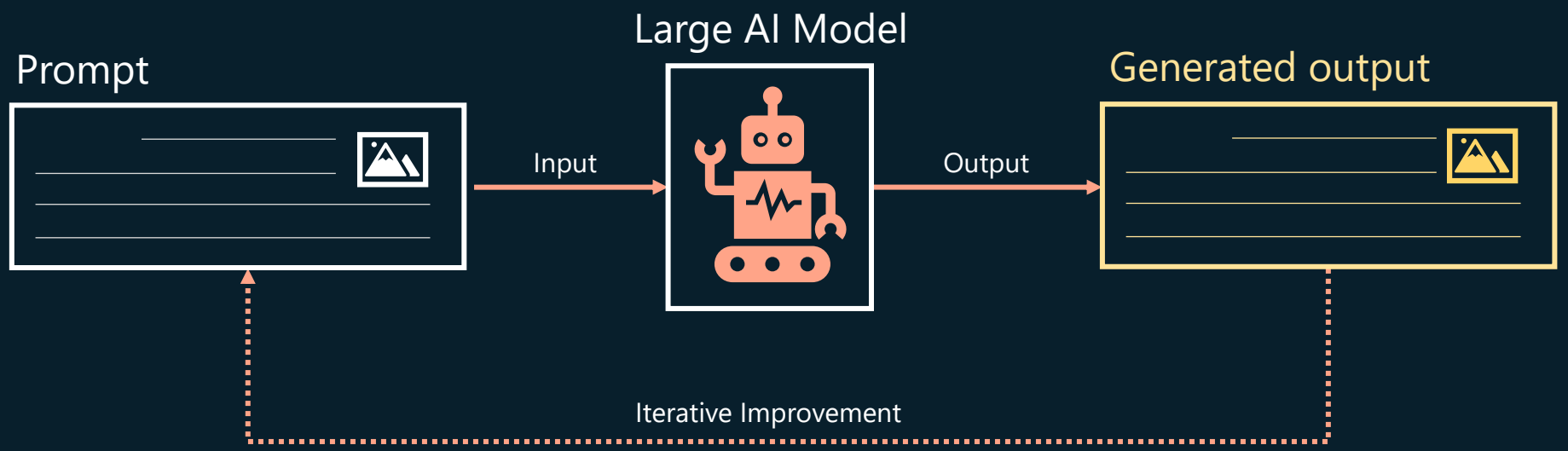
Context optimization

LLM Optimization

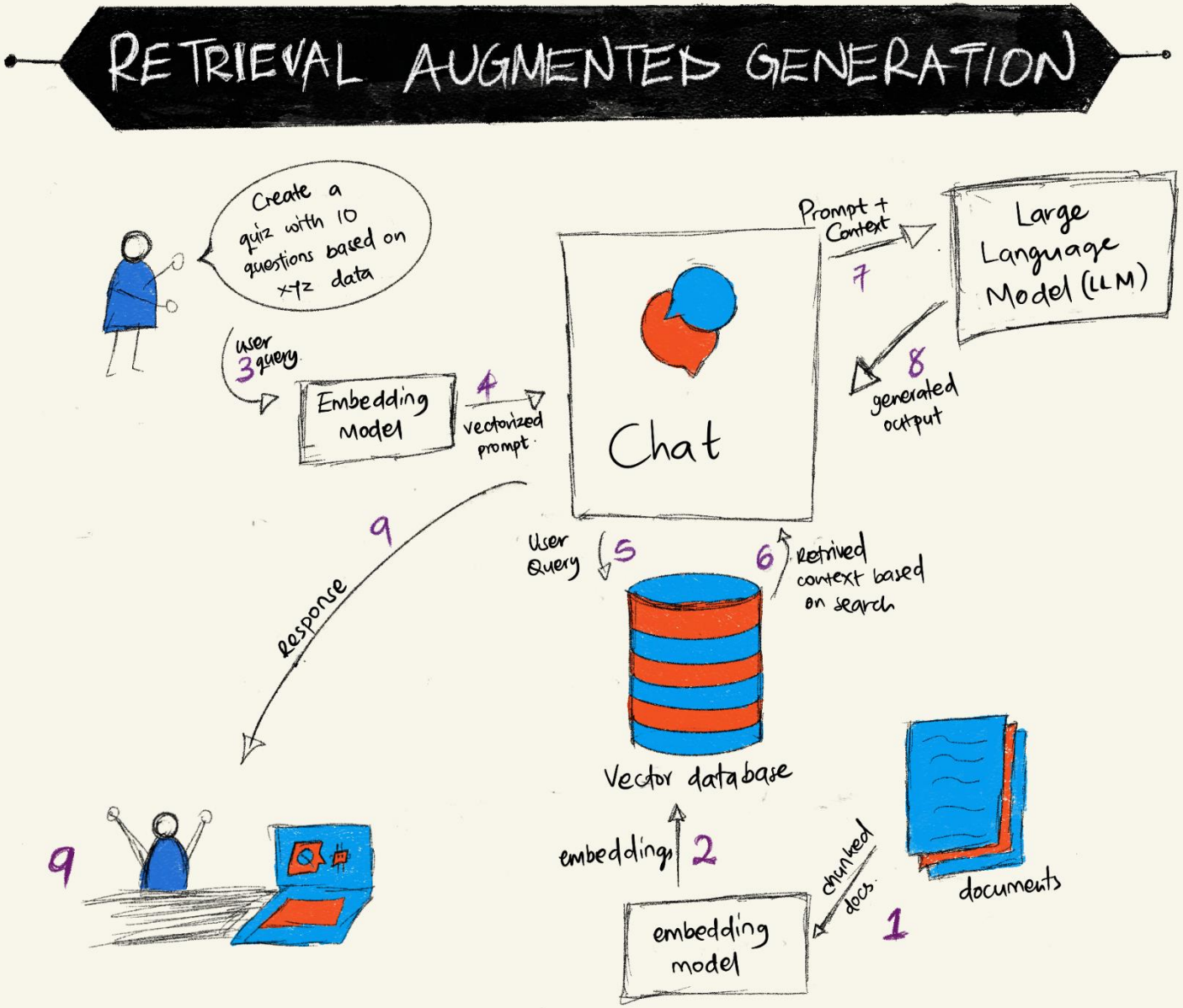


What is Prompt Engineering?

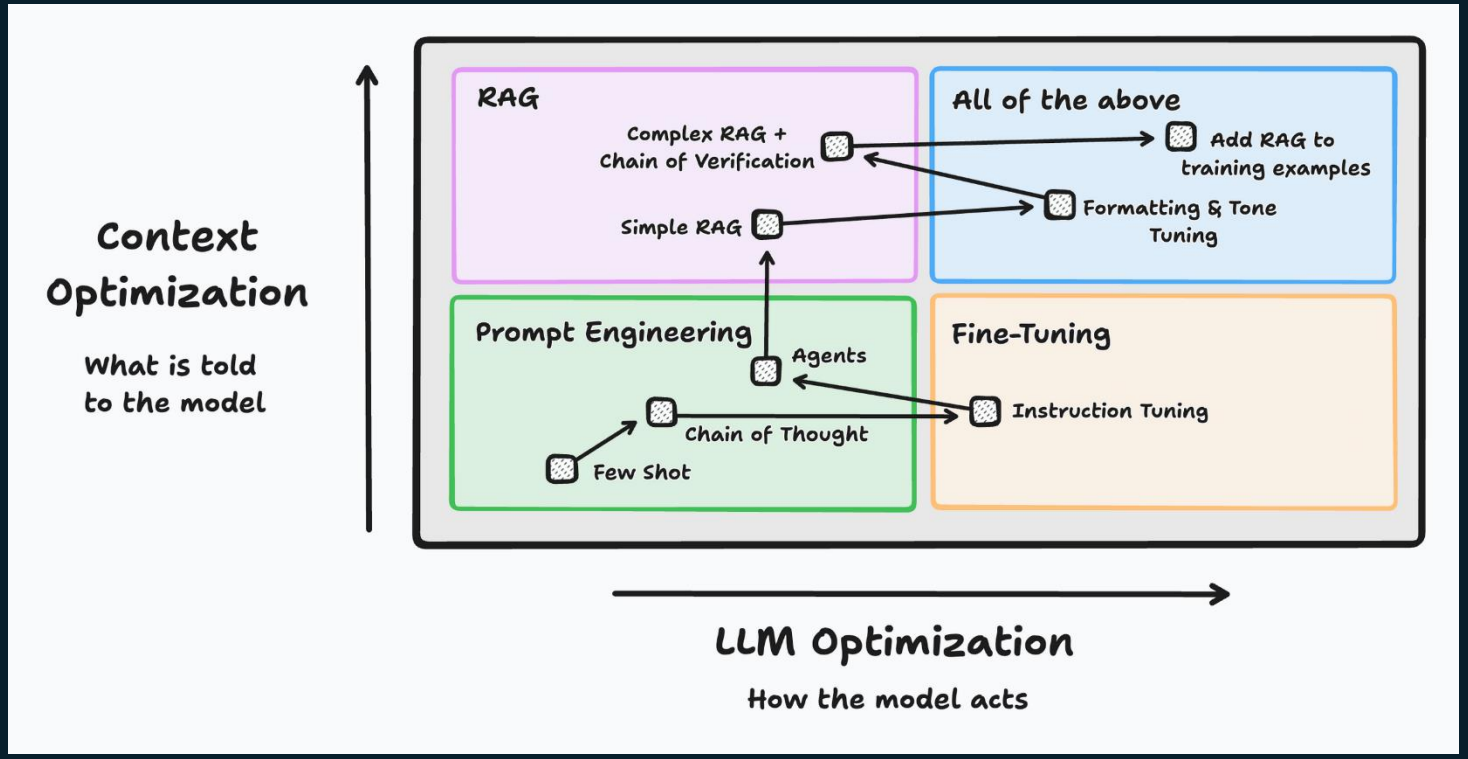
- Two step process:
  - *designing* the initial prompt for a given model and objective
  - *refining* the prompt iteratively to improve the quality of the response



What is RAG?

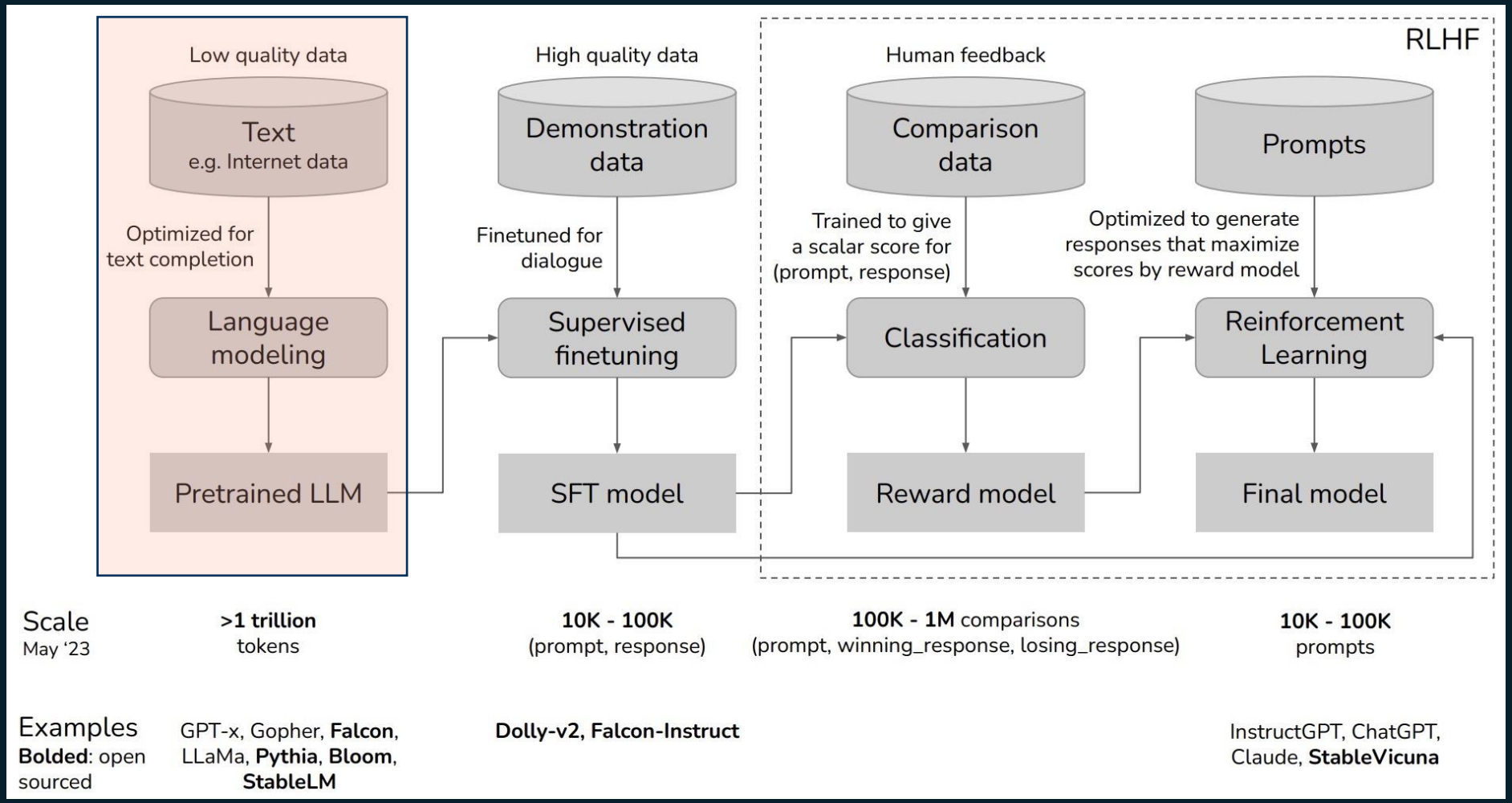

RETRIEVAL AUGMENTED GENERATION
Create a quiz with 10 questions based on xyz data
user query
3
Embedding Model
4
vectorized prompt
Chat
Prompt + Context
7
Large Language Model (LLM)
8
generated output
9
response
User Query
5
6
Retrived context based on search
Vector database
embedding
2
embedding model
chunked docs
1
documents
9

Model Finetuning (LLM Optimization)



Model Training Landscape

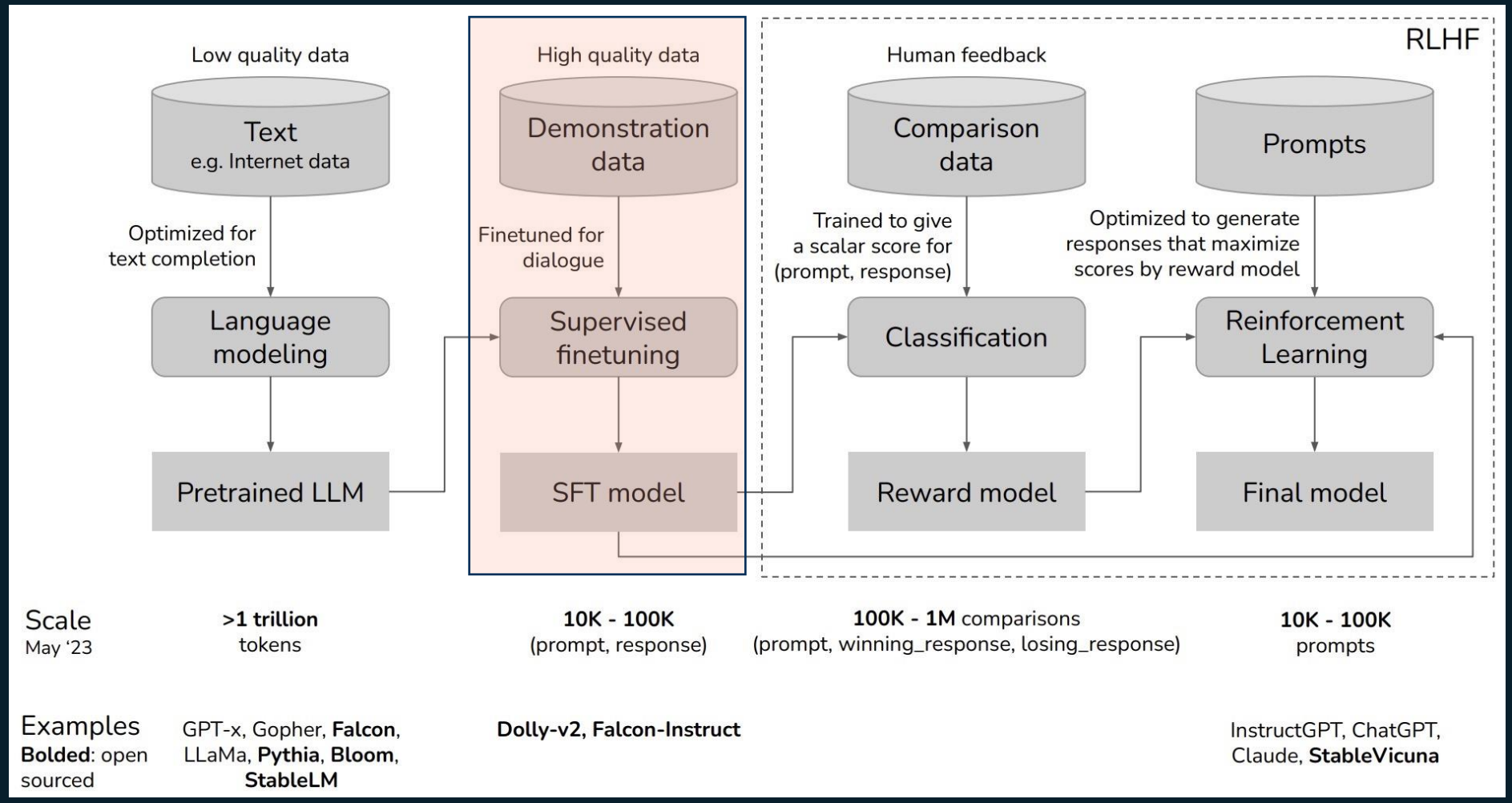
1. Model Pretraining



https://huyenchip.com/2023/05/02/rlhf.html

Model Training Landscape

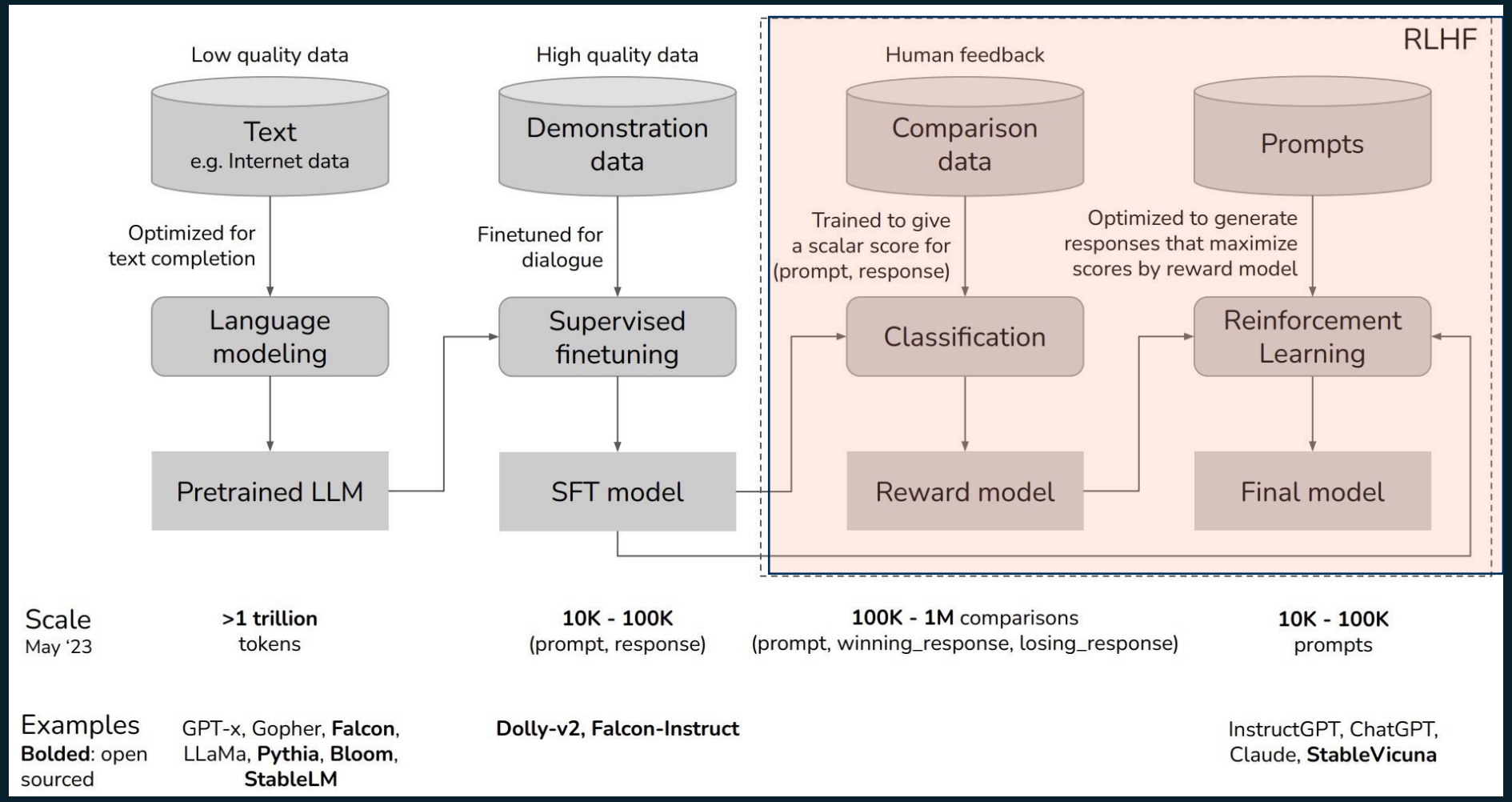
2. Supervised Model Finetuning (aka, Instruction Tuning)



https://huyenchip.com/2023/05/02/rlhf.html

Model Training Landscape

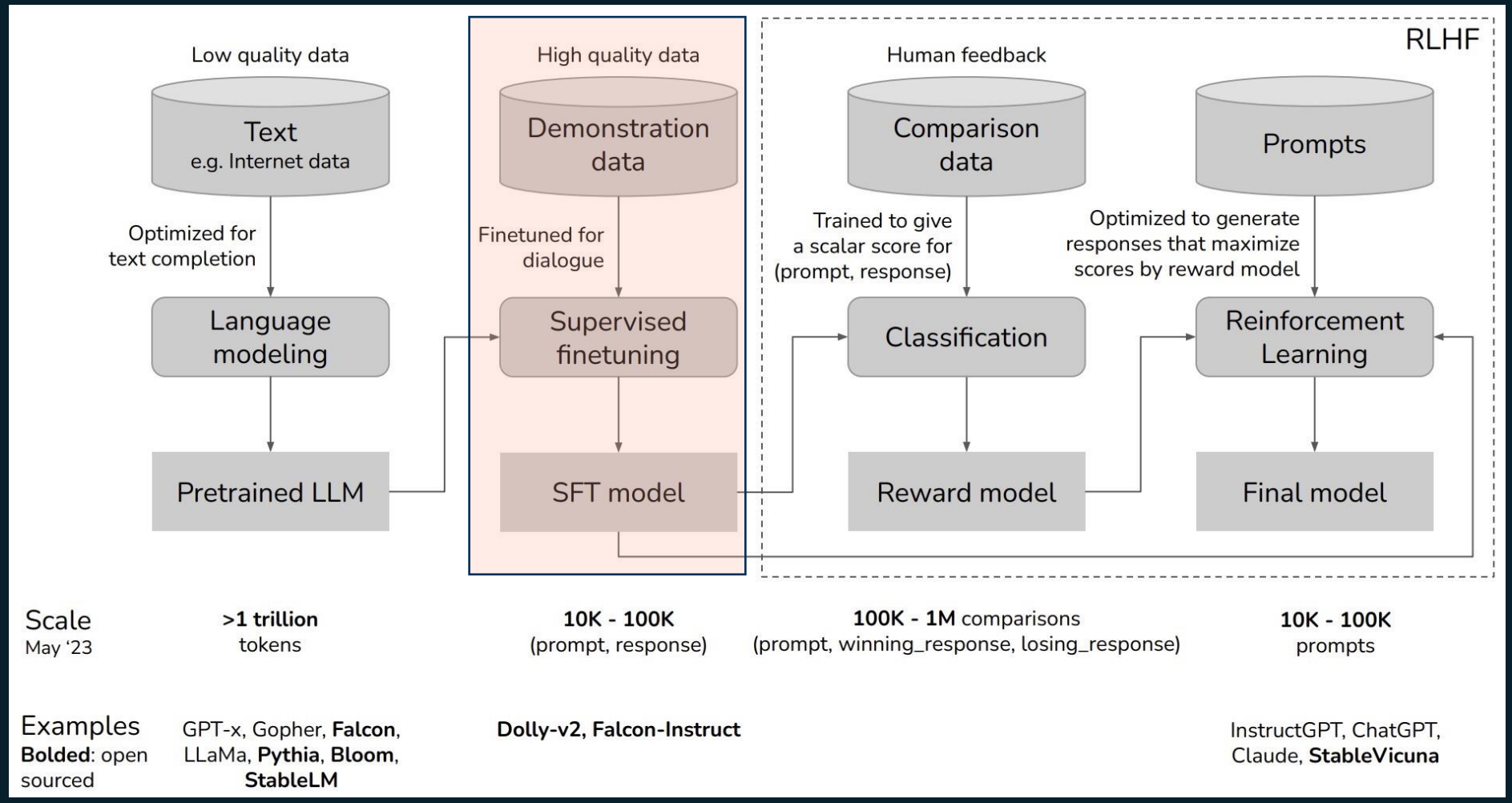
3. RL based Model Alignment



https://huyenchip.com/2023/05/02/rlhf.html

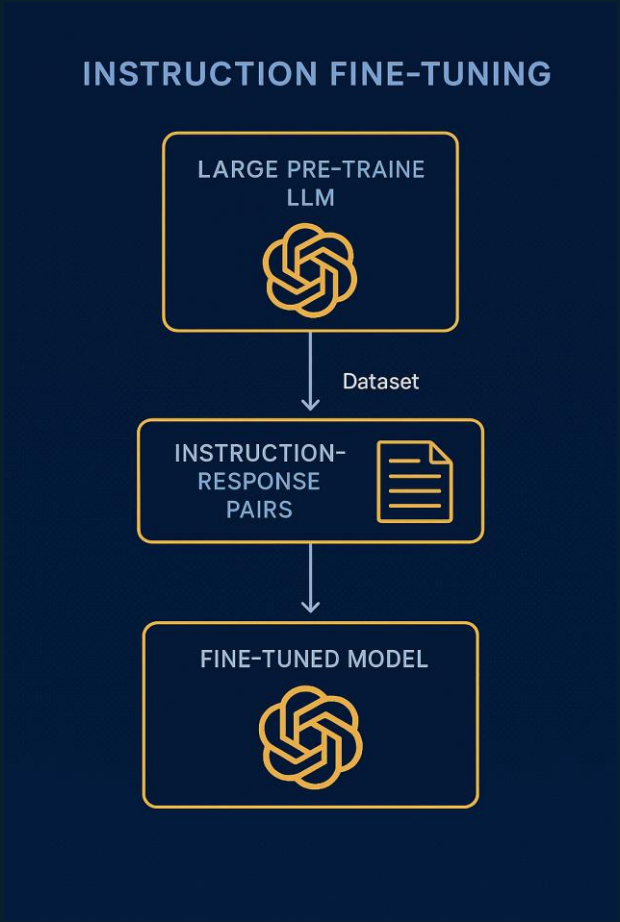
Model Training Landscape

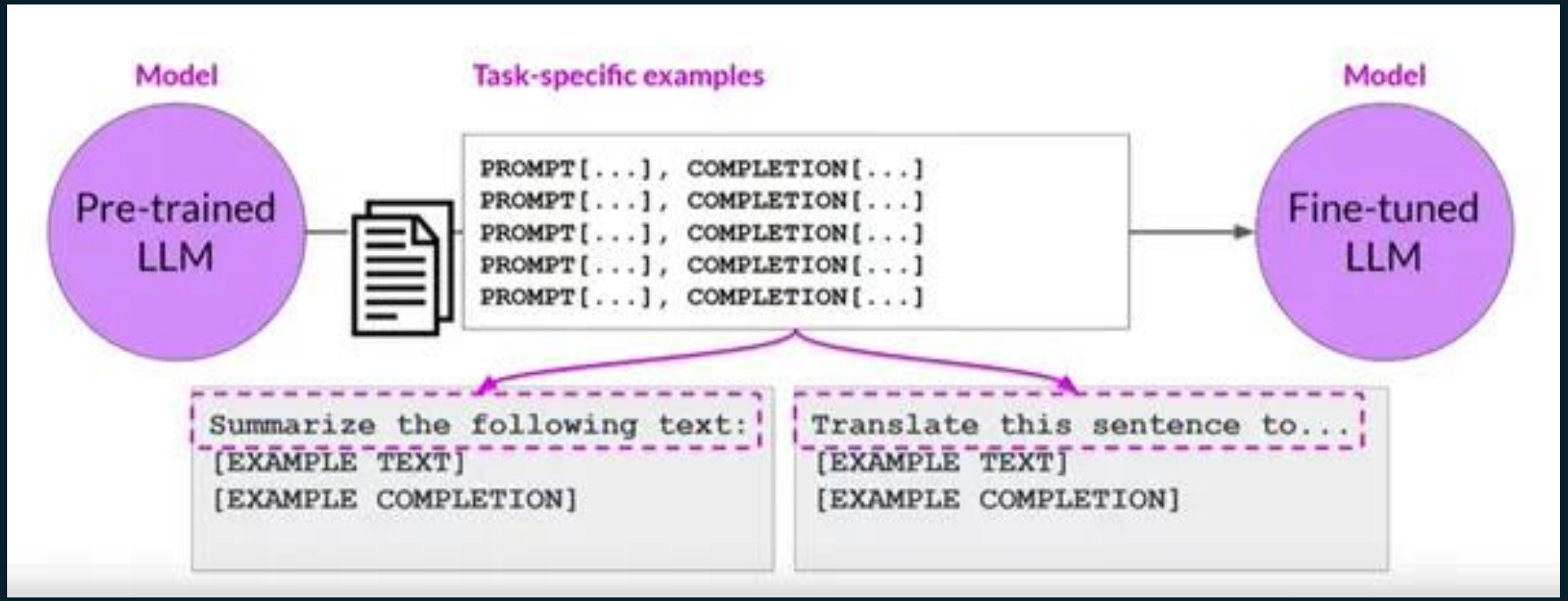
2. Supervised Model Finetuning (aka, Instruction Tuning)



https://huyenchip.com/2023/05/02/rlhf.html

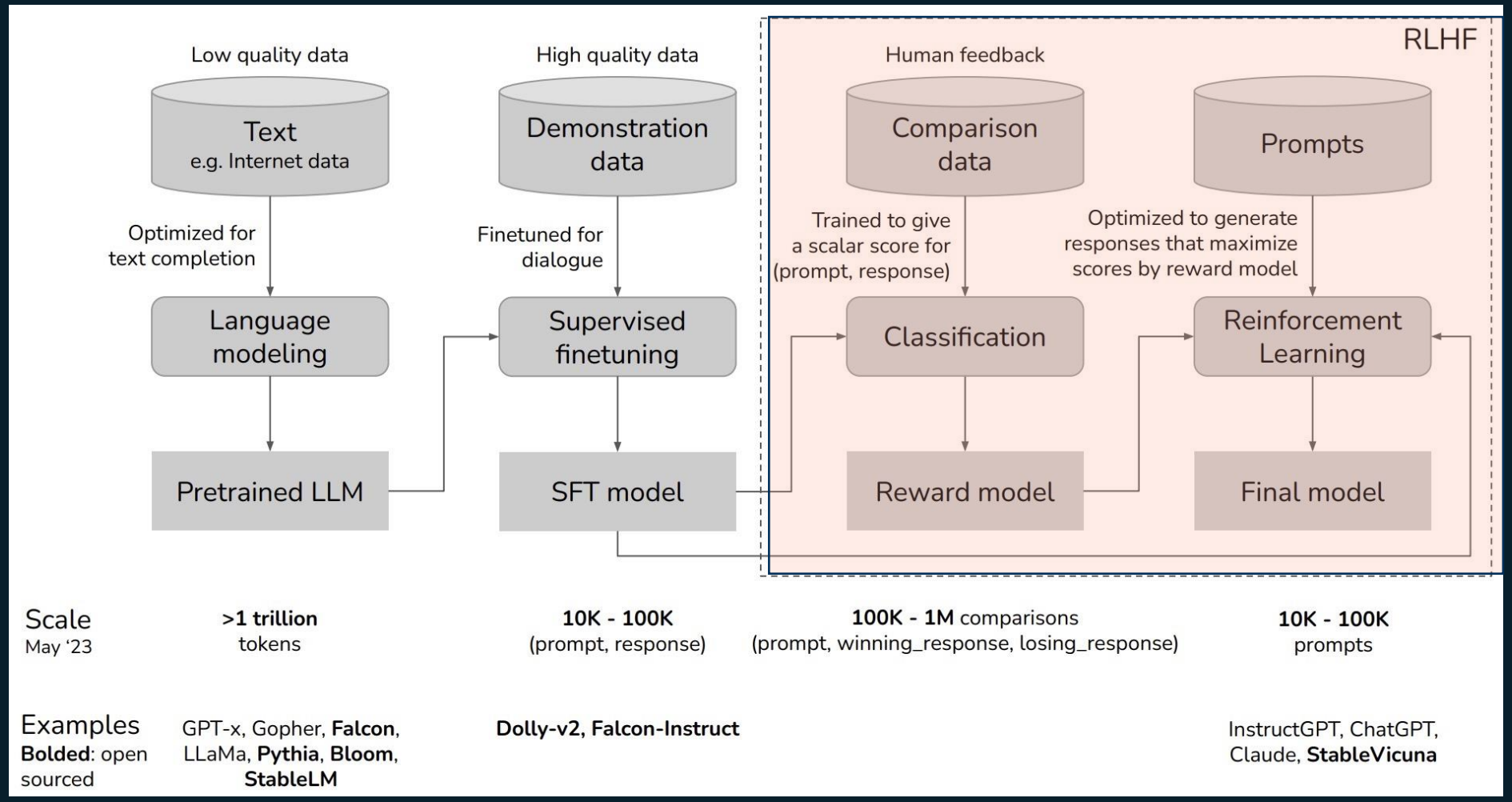
Supervised Tuning (aka, Instruction Tuning)





Model Training Landscape

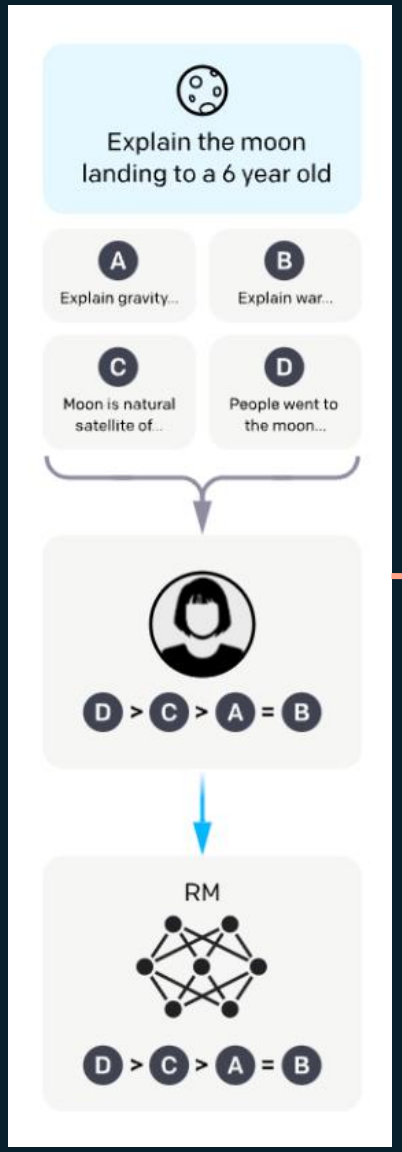
3. RL based Model Alignment

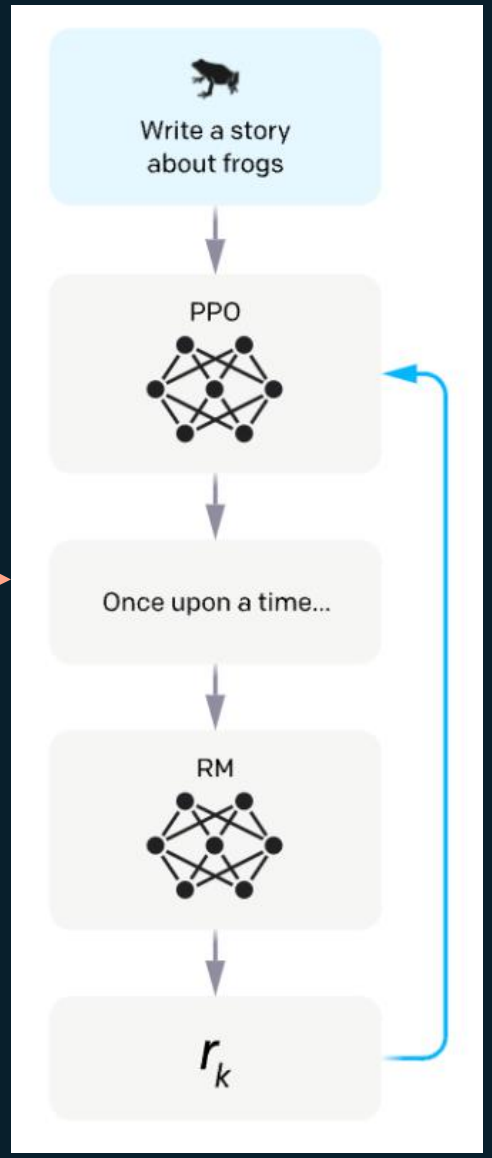


https://huyenchip.com/2023/05/02/rlhf.html

RL based Model Alignment (RLHF)

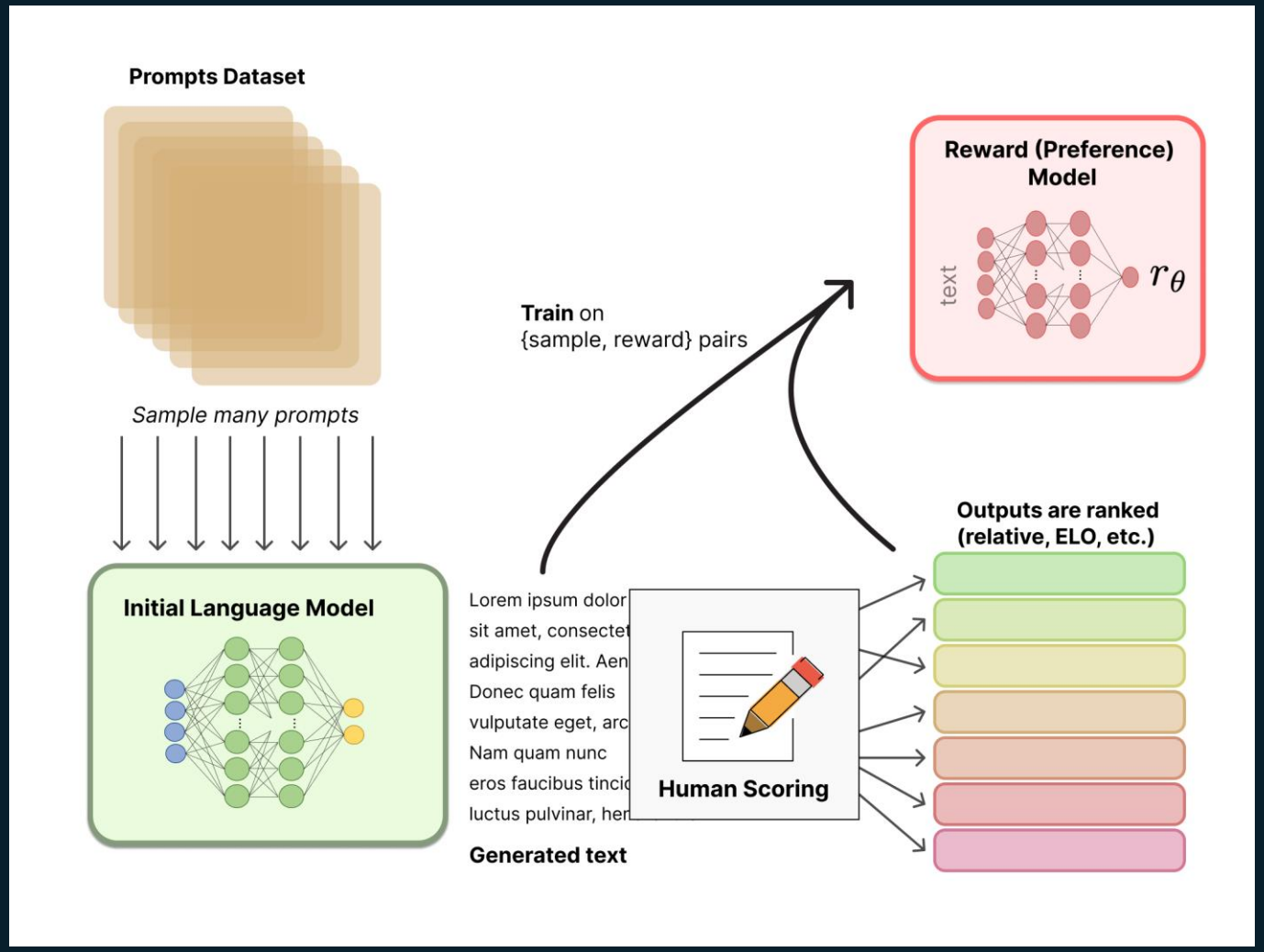
Step1:
Learn a reward model from human feedback





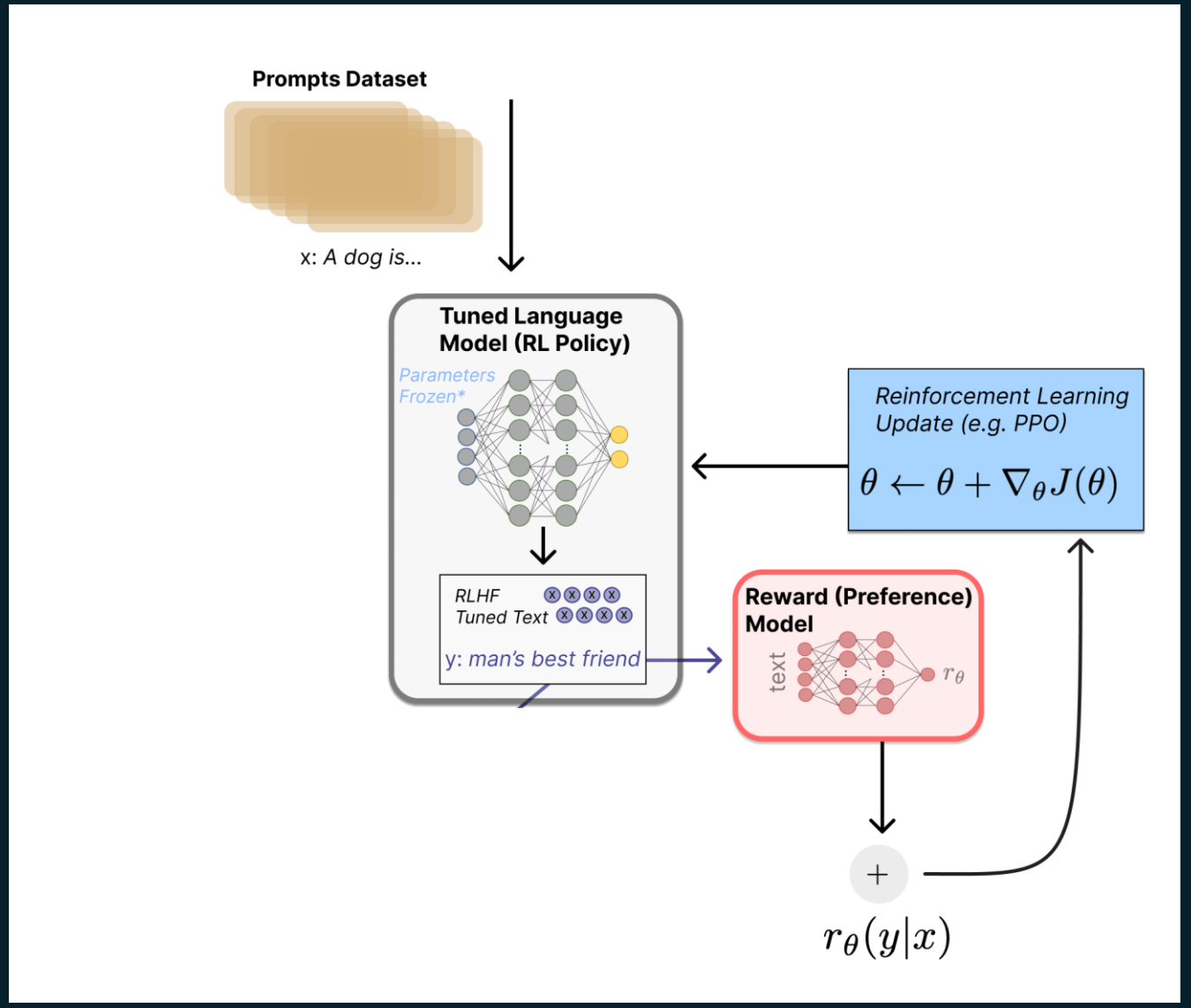
Step2:
Using the reward model, learn (aka align) the GenAI model

Reward Model

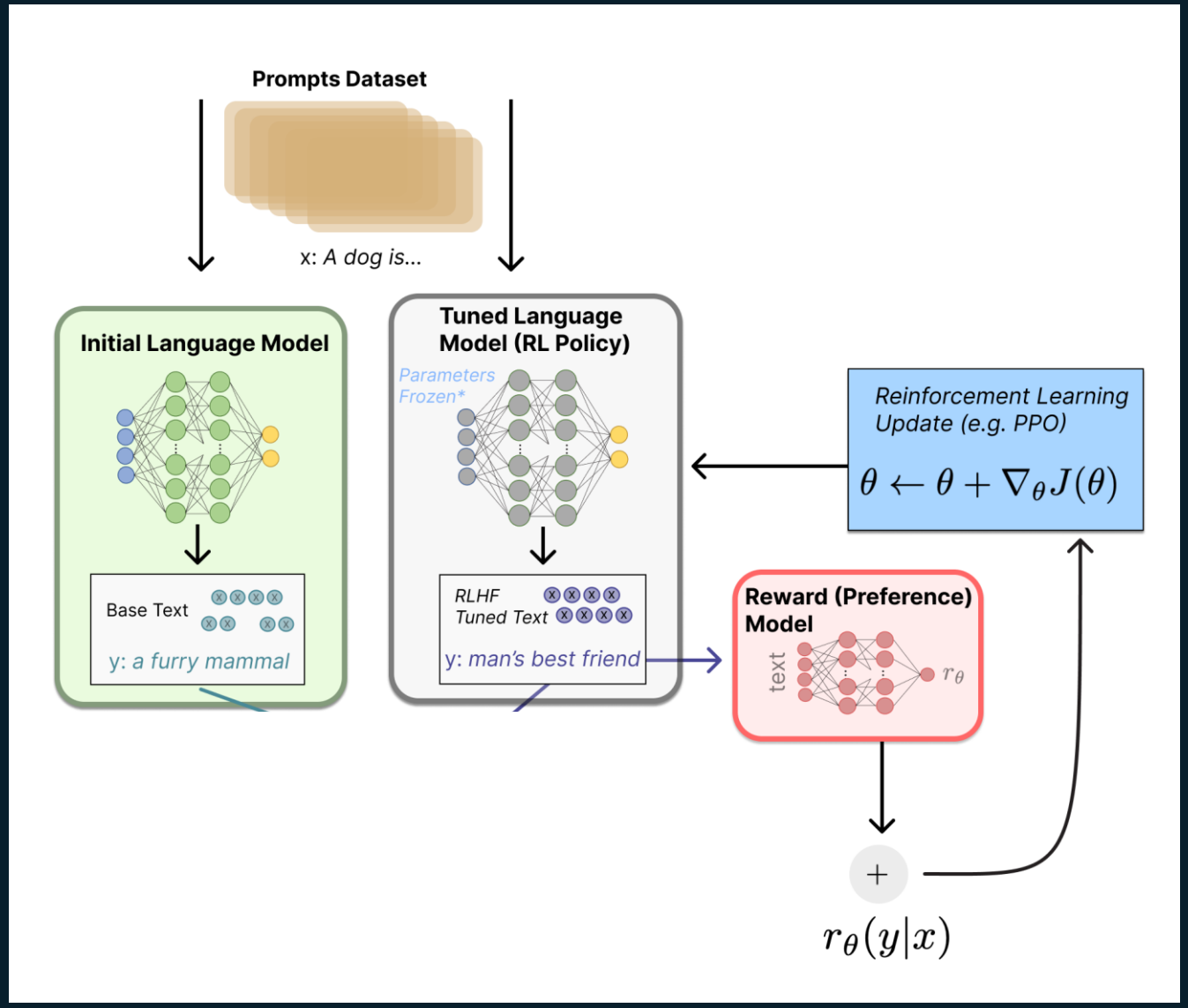


https://huggingface.co/blog/rlhf

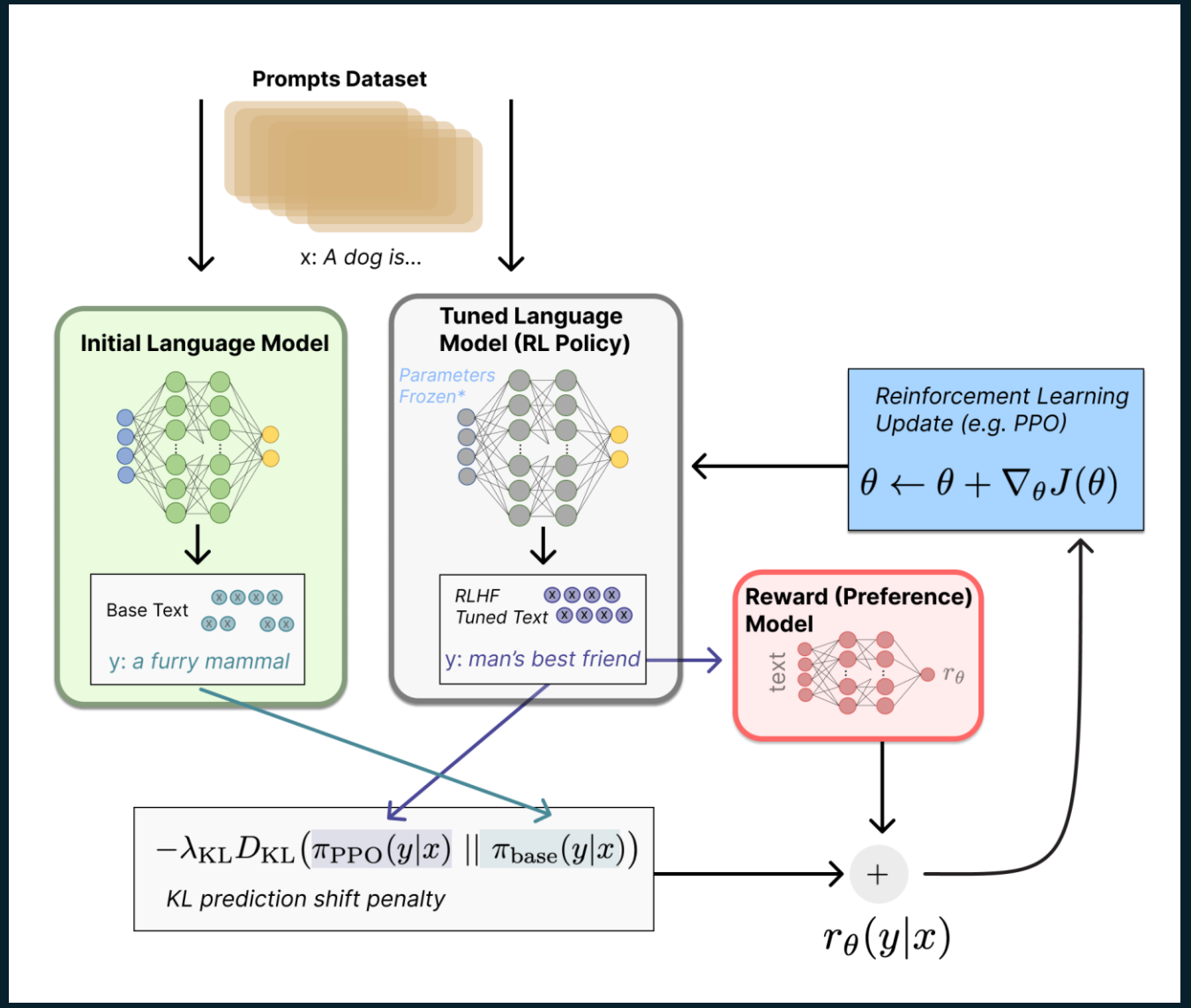
Finetune (aka Align) with RL



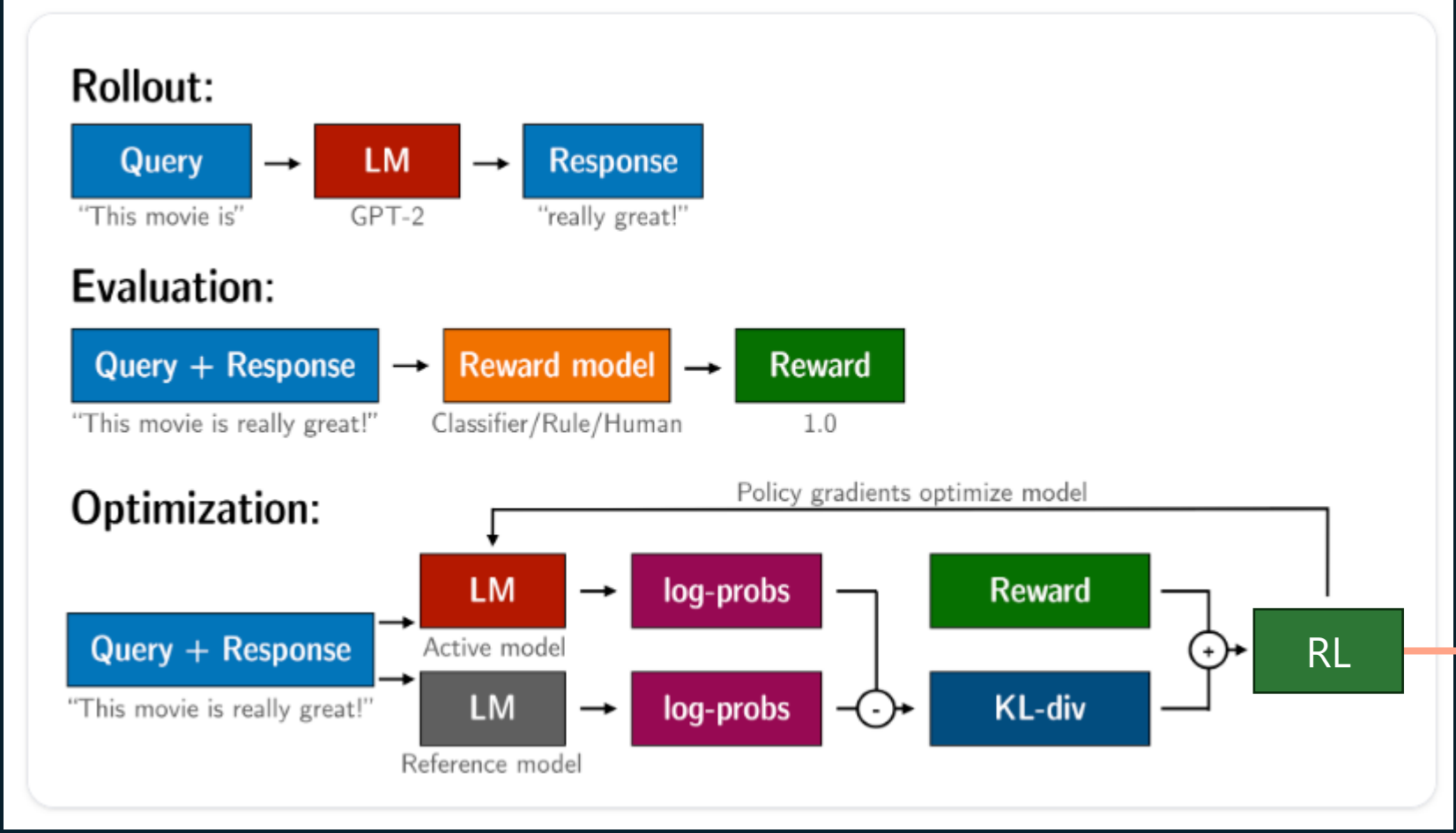
Finetune (aka Align) with RL



Finetune (aka Align) with RL



Model Alignment Process

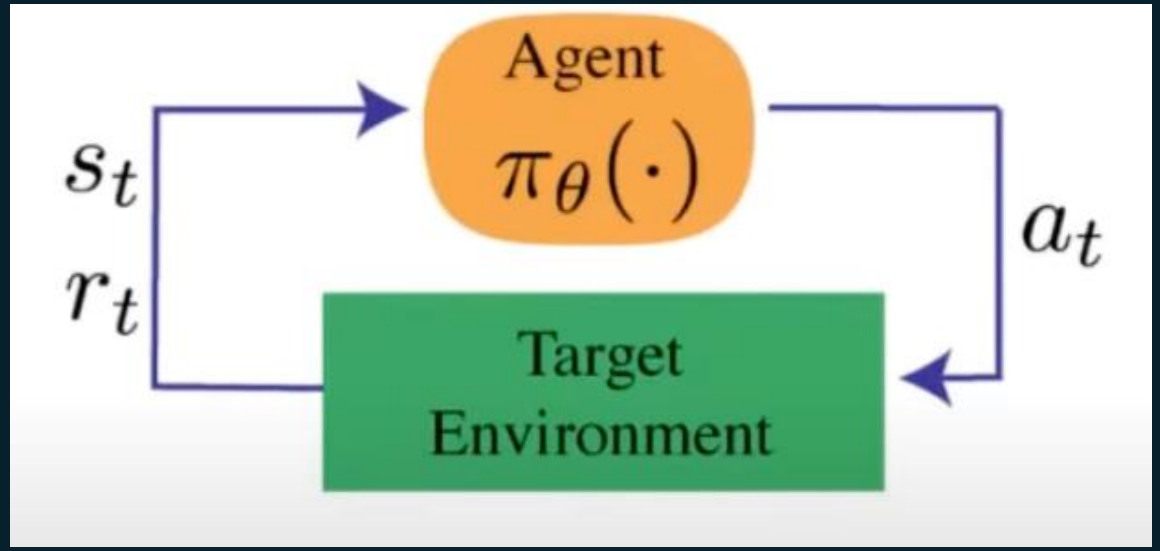


https://huggingface.co/docs/trl/main/en/ppo_trainer

- Reinforce
- TRPO
- PPO
- DPO
- ORPO
- REINFORCE++
- GRP
- GRPO
- …

Intro to RL

- **State (s)**: The agent's perception of the environment
- **Action (a)**: The agent's choice affecting the environment.
- **Reward (r)**: A scalar feedback signal.
- **Policy (π(a‖s))**: A probability distribution over actions given a state.
- **Value Function (Vπ(s))**: Expected cumulative rewards from state s.
- **Advantage Function (Aπ(s,a))**: Measures how much better an action is compared to the baseline value.



Objective Function

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$$

REINFORCE (1992)

Objective Function

$$\theta \leftarrow \theta + \alpha \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) R_t$$

- π_θ is the policy parameterized by θ, at is the action taken at time t,
- s_t is the state at time t,
- R_t is the cumulative return from time step t, and
- α is the learning rate

https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12

Challenge: High Variance

TRPO (2015)

Trust Region Policy Optimization

Objective Function

$$\max_{\theta} \mathbb{E}_{s,a\sim\pi_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A^{\pi_{\theta_{\text{old}}}}(s,a) \right]$$

subject to the constraint

$$D_{KL}(\pi_\theta || \pi_{\theta_{\text{old}}}) \leq \delta$$

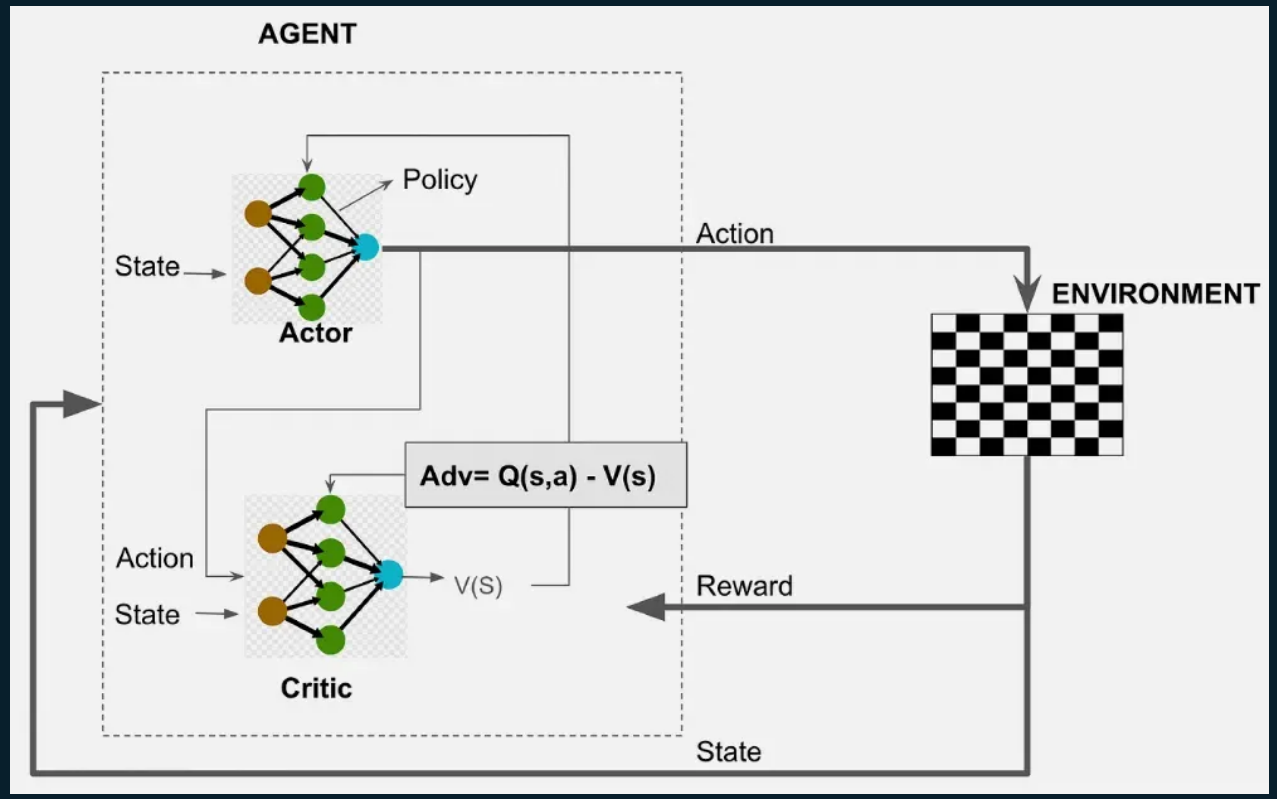
https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12

Challenge: Computationally Complex

PPO (2017)

Proximal Policy Optimization

Actor-Critic Formulation



Objective Function

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

General Advantage Estimation (GAE)

$$A_t = Q(s_t, a_t) - V(s_t)$$

$$A_K^{\text{GAE}} = \sum_{t=0}^{K-1} (\lambda)^t \delta_t$$

A2C: Advantage Actor-Critic

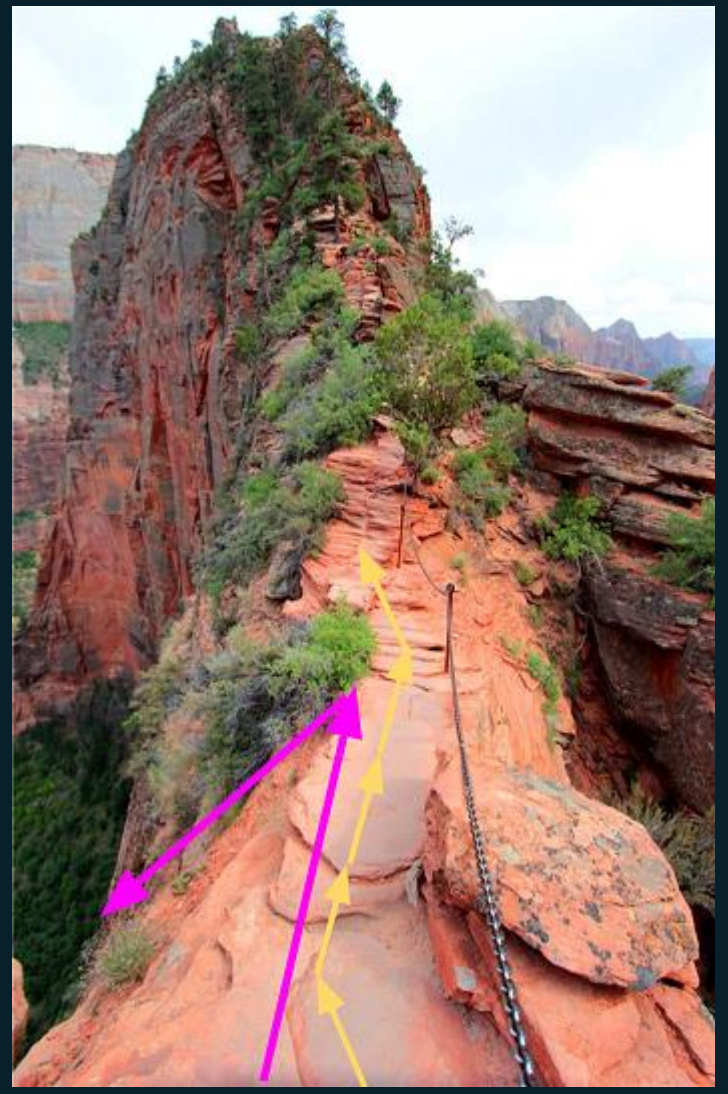
1.Generate responses: LLM produces multiple responses for a given prompt

2.Score responses: The reward model assigns reward for each response

3.Compute advantages: Use GAE to compute advantages

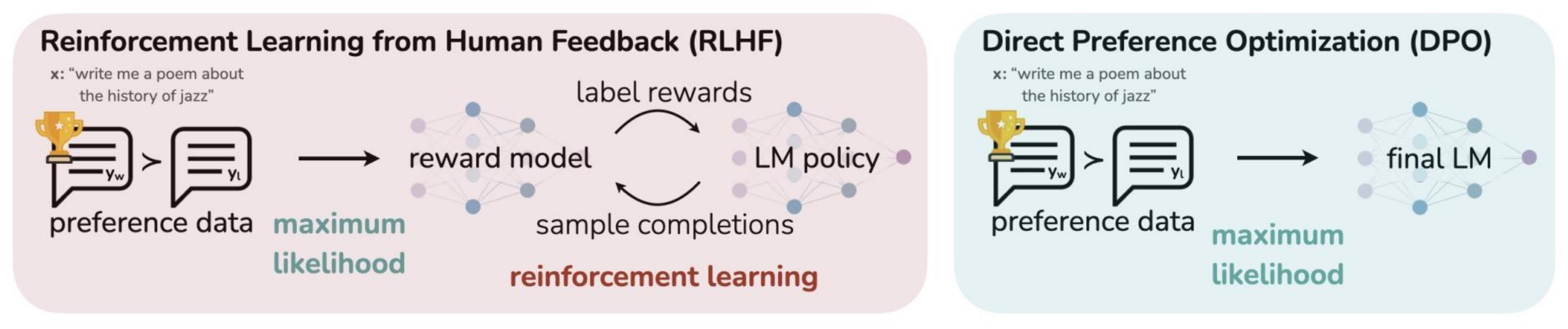
4.Optimize policy: Update the LLM by optimizing the total objective

5.Update critic: Train the value function to be better at predicting the rewards given partial responses



DPO (2023)

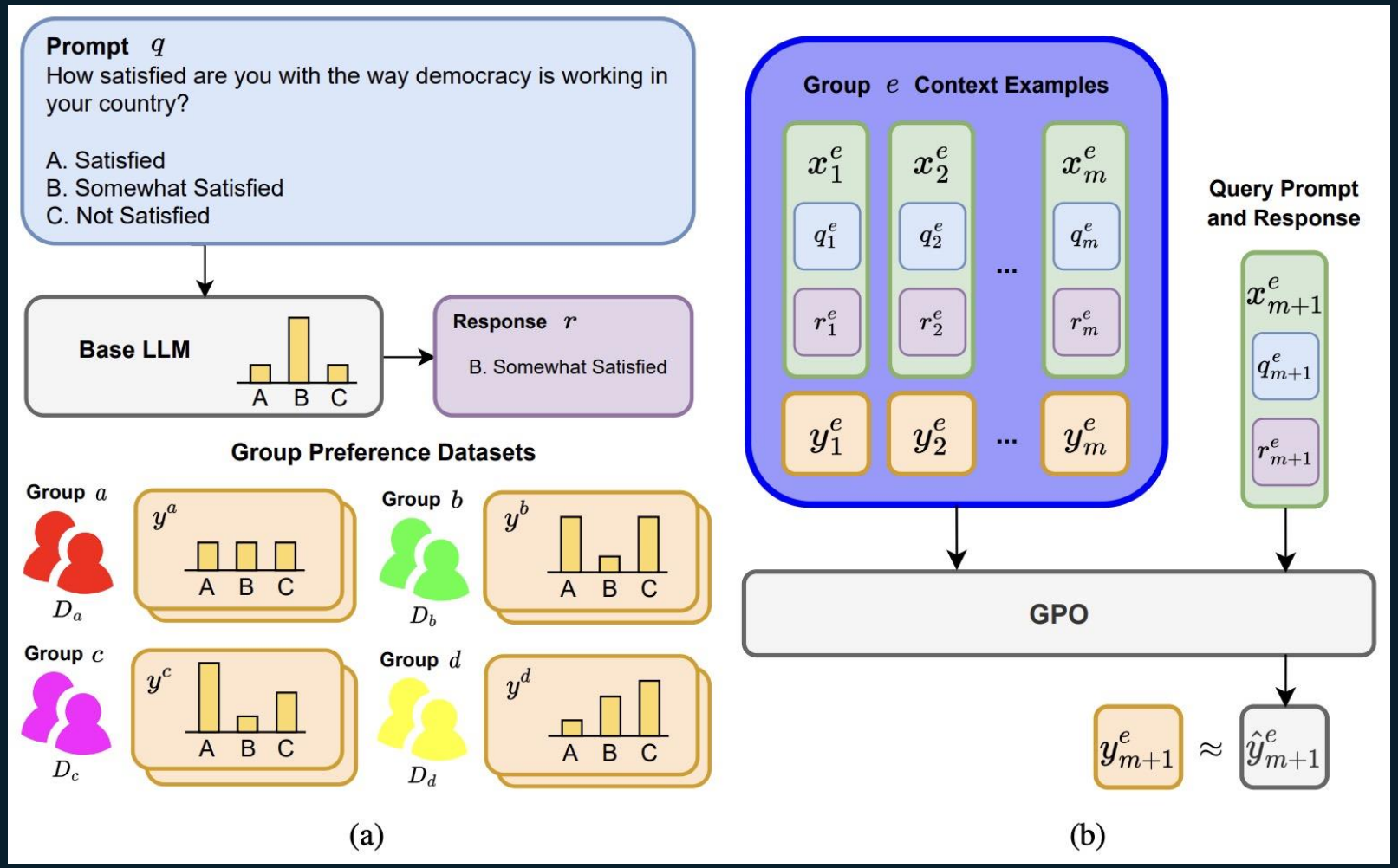
Direct Policy Optimization



[Direct Preference Optimization: Your Language Model is Secretly a Reward Model]

GPO (2023)

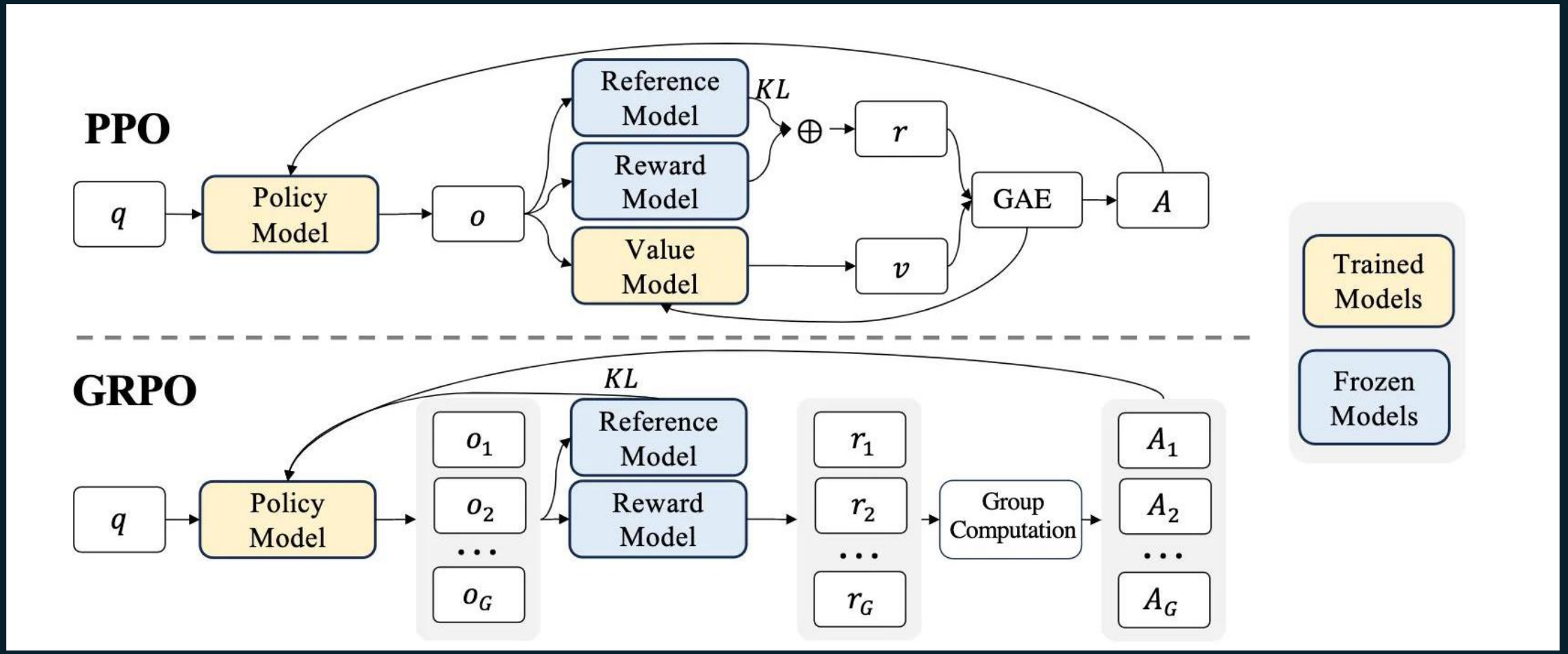
Group Policy Optimization



GRPO (2024)

Group Relative Policy Optimization

The main goal of GRPO is to improve computational efficiency, reduce memory usage



DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models

REINFORCE++ (2025)

1. Token-level KL penalty

$$\mathrm{KL}(t) = \log\left(\frac{\pi_{\theta_{\text{old}}}^{\text{RL}}(a_t|s_t)}{\pi^{\text{SFT}}(a_t|s_t)}\right)$$

2. PPO-Clip integration

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

3. Mini-batch updates

4. Reward normalization and clipping

5. Advantage normalization

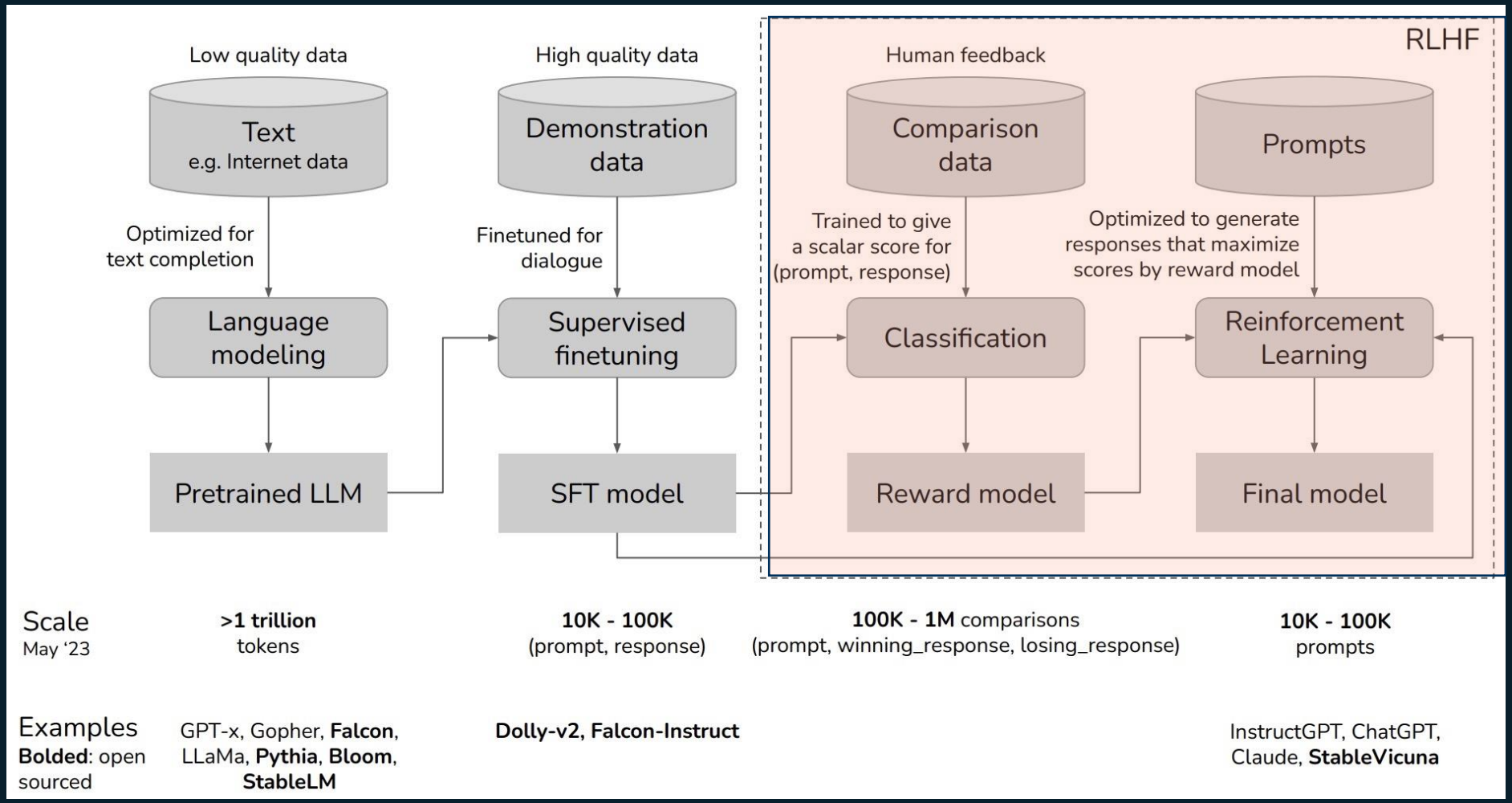
$$A_t(s_t, a_t) = r(x, y) - \beta \cdot \sum_{i=t}^{T} \mathrm{KL}(i)$$

Comparison

Aspect	REINFORCE	TRPO	PPO	DPO	GRPO
Objective	Policy gradient optimization without constraints.	Ensures stable policy updates within a constrained region.	Maximizes expected reward while preventing large policy updates.	Optimizes policy based on binary classification of human preferences.	Leverages group-based relative advantages and removes the critic network.
Learning Mechanism	Monte Carlo policy gradients with high variance.	Second-order optimization with trust region constraints.	Policy gradients with a clipped surrogate objective.	Cross-entropy optimization over paired preferences.	Group normalization with policy gradients, eliminating the critic network.
Stability	Low (high variance, unstable updates).	High (enforces trust region for stable updates).	Relies on clipping mechanisms to avoid destabilization.	Stable as it directly optimizes preferences.	Stable due to normalization of rewards across groups.
Training Complexity	High (unconstrained updates).	Very high (requires second-order optimization and solving constraints).	High, due to balancing reward maximization with policy constraints.	Moderate; uses simplified binary preference objectives.	Reduces overhead via group-based scoring.
Performance	Unstable and sample-inefficient.	More stable than PPO but computationally expensive.	Strong performance on tasks with clear reward signals but prone to instability in distributed setups.	Effective for straightforward preference alignment tasks.	Excels in reasoning tasks, offering computational efficiency.
Notable Strength	Simple to implement but inefficient.	Ensures stable policy updates through trust-region constraints.	Widely used in RL settings, good at reward-based optimization.	Directly optimizes for preferences without needing a separate reward model.	Simplifies reward aggregation; strong for reasoning-heavy tasks.
Scenarios Best Suited	RL tasks where simplicity is preferred over efficiency.	High-stability RL tasks requiring constraint-driven policy improvements.	RL environments where reward signals are predefined.	Scenarios with abundant paired human feedback.	Mathematical reasoning or low-resource training setups.

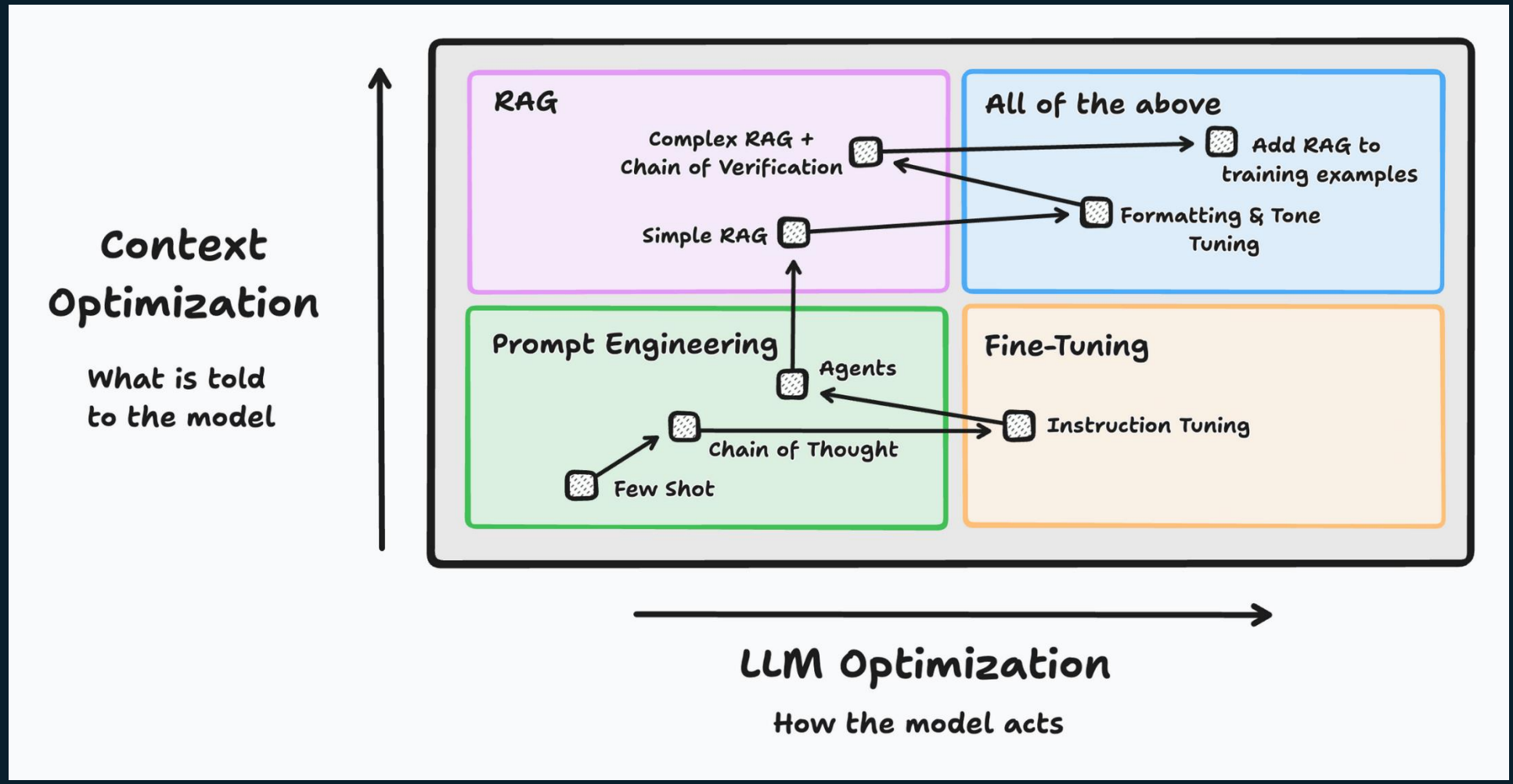
Revisit: Model Training Landscape

3. RL based Model Alignment



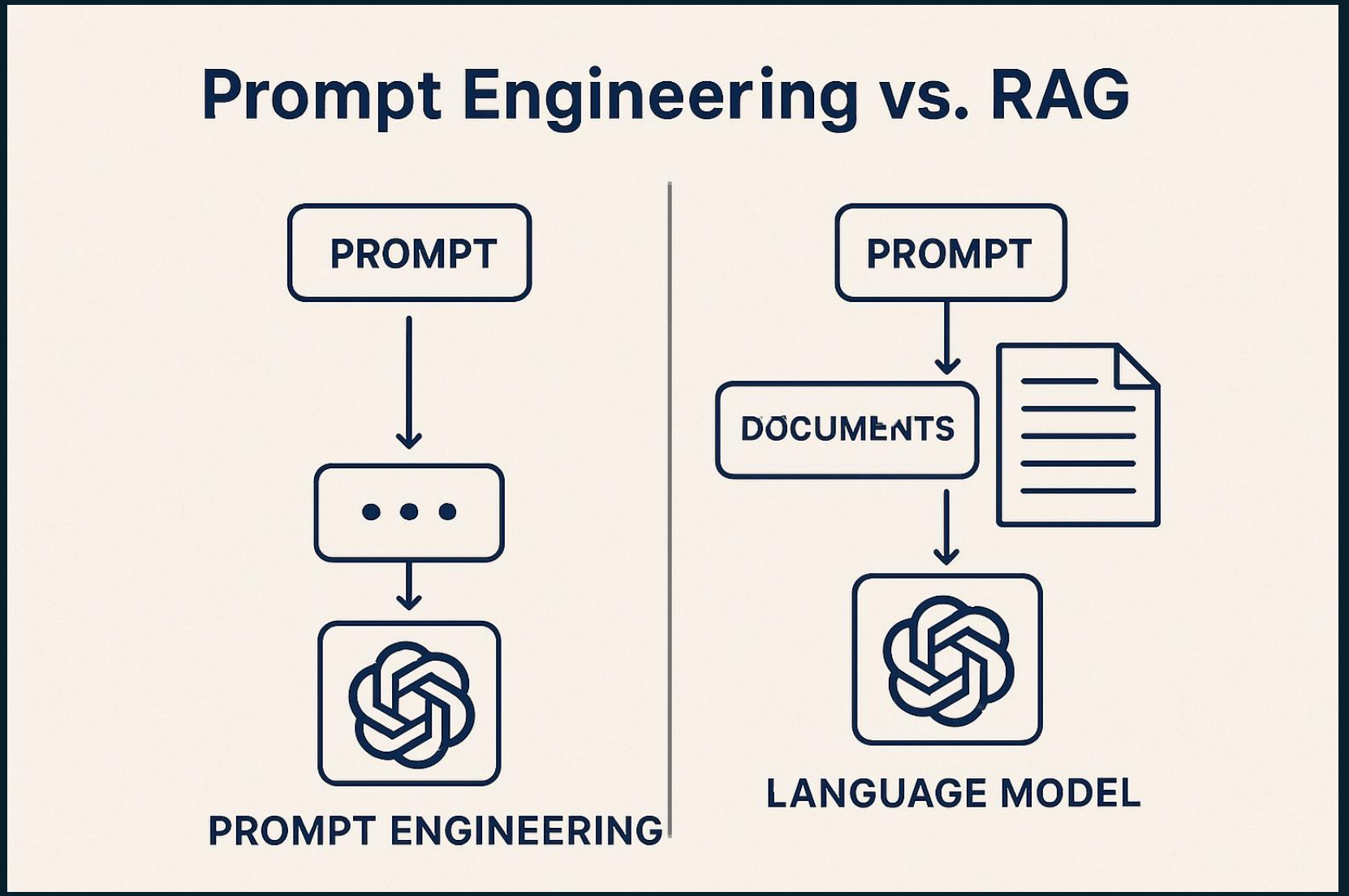
https://huyenchip.com/2023/05/02/rlhf.html

Revisit: LLM Optimization

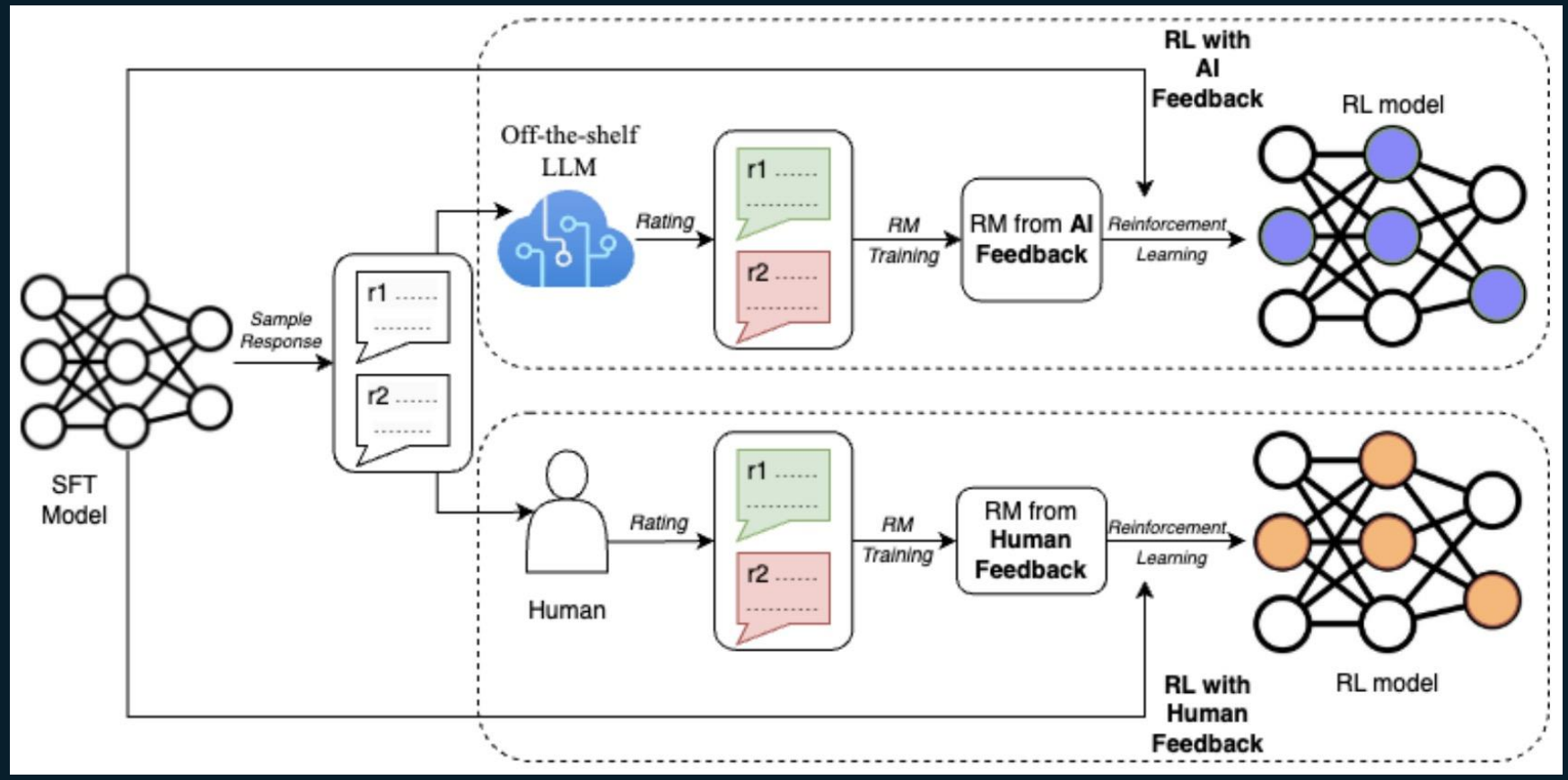


Backup

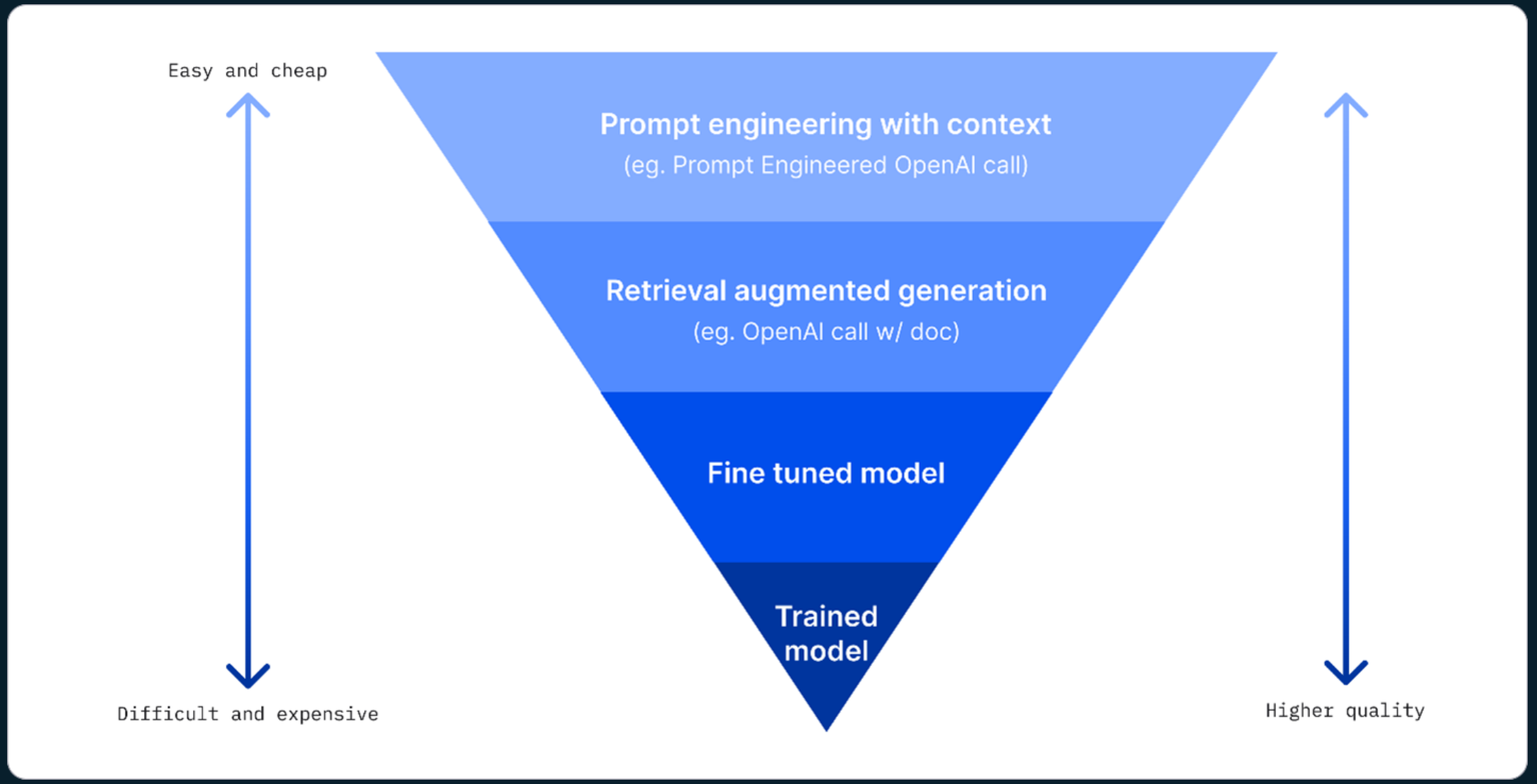
Prompt Engineering vs. RAG



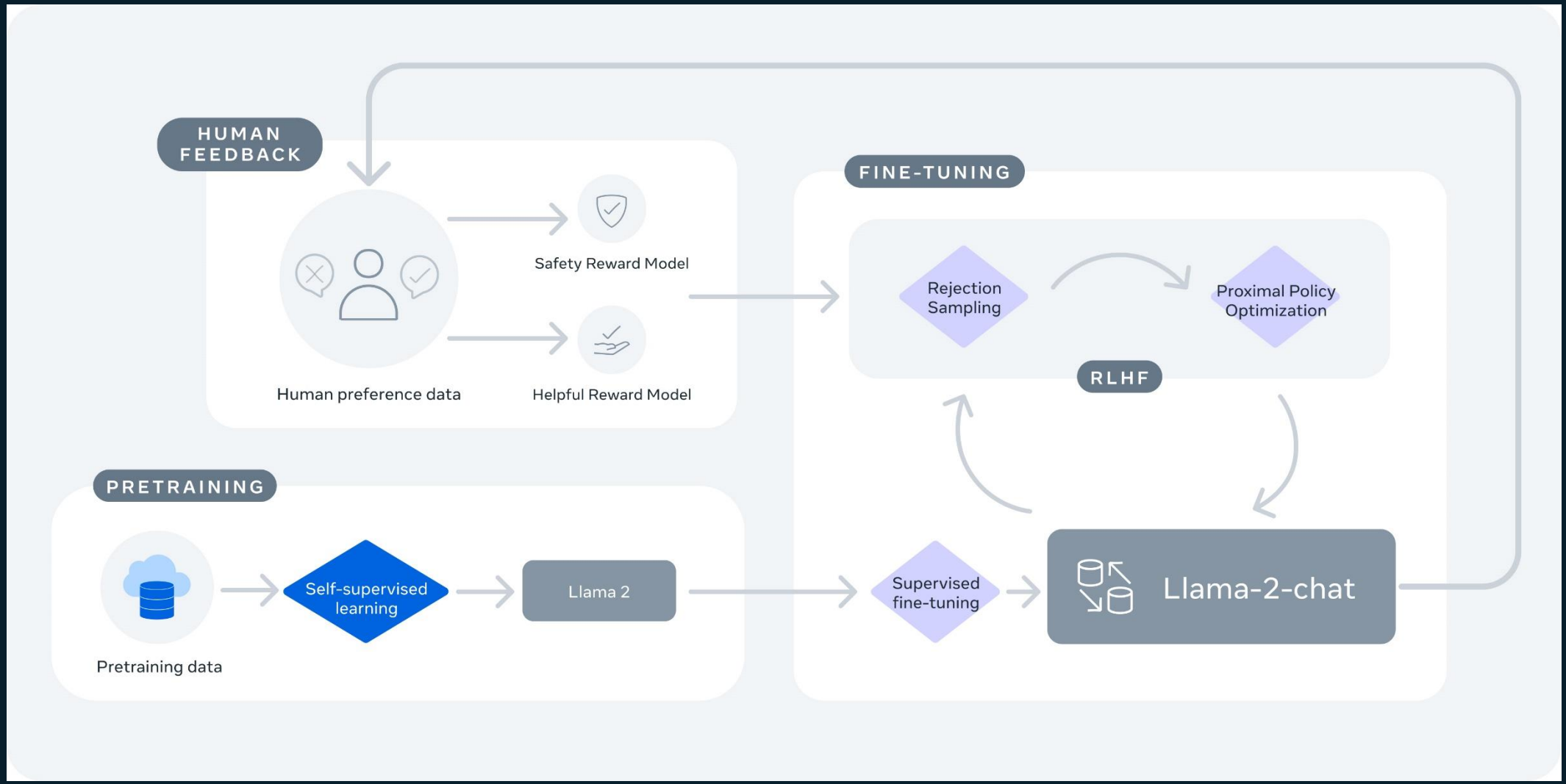
RLAIF


RL with AI Feedback
RL model
Off-the-shelf LLM
Rating
r1
r2
RM Training
RM from AI Feedback
Reinforcement Learning
Sample Response
SFT Model
Human
RM from Human Feedback
RL with Human Feedback

Why RAG?


Easy and cheap
Difficult and expensive
Prompt engineering with context
(eg. Prompt Engineered OpenAI call)
Retrieval augmented generation
(eg. OpenAI call w/ doc)
Fine tuned model
Trained model
Higher quality

Llama 2 Chat: Example



https://www.llama.com/llama2/