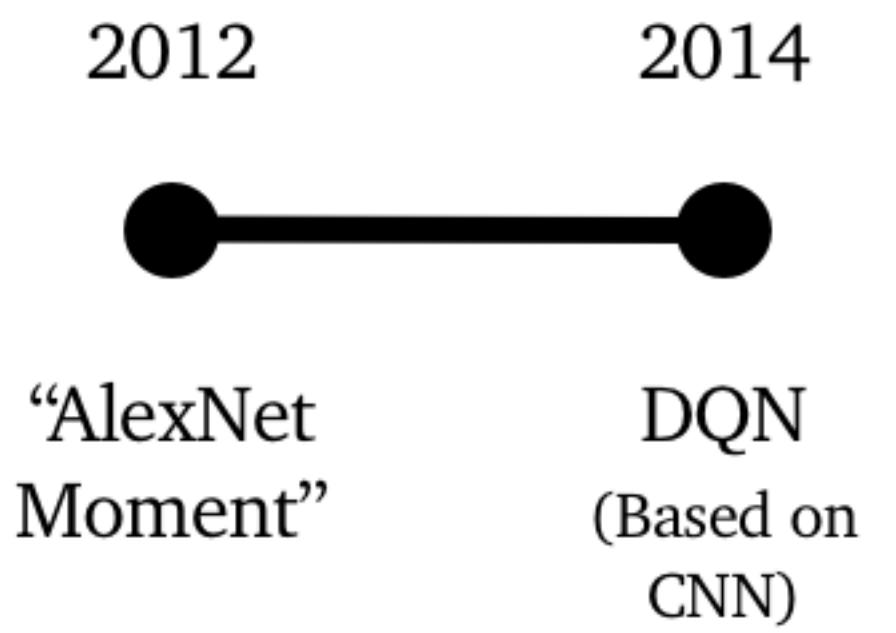
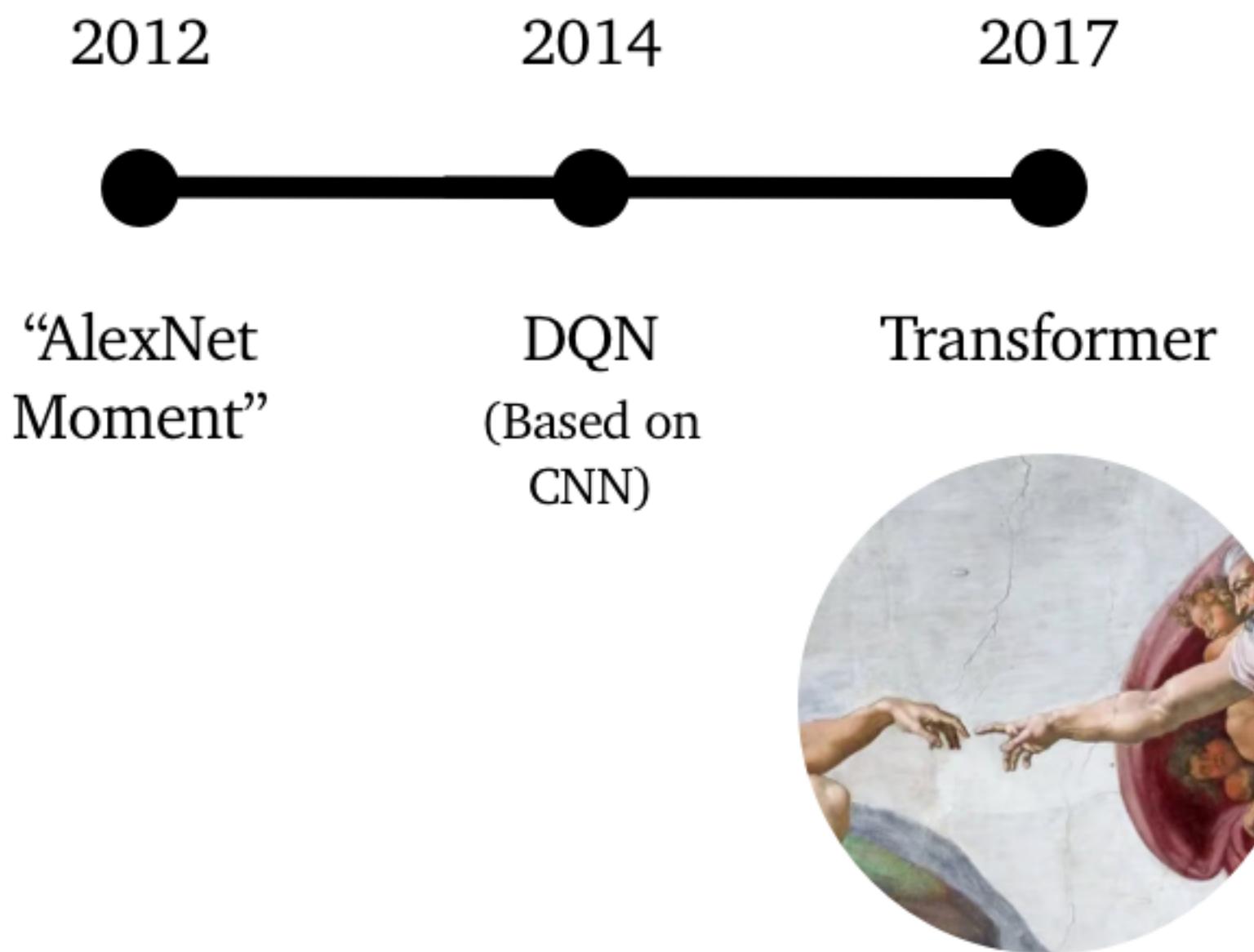


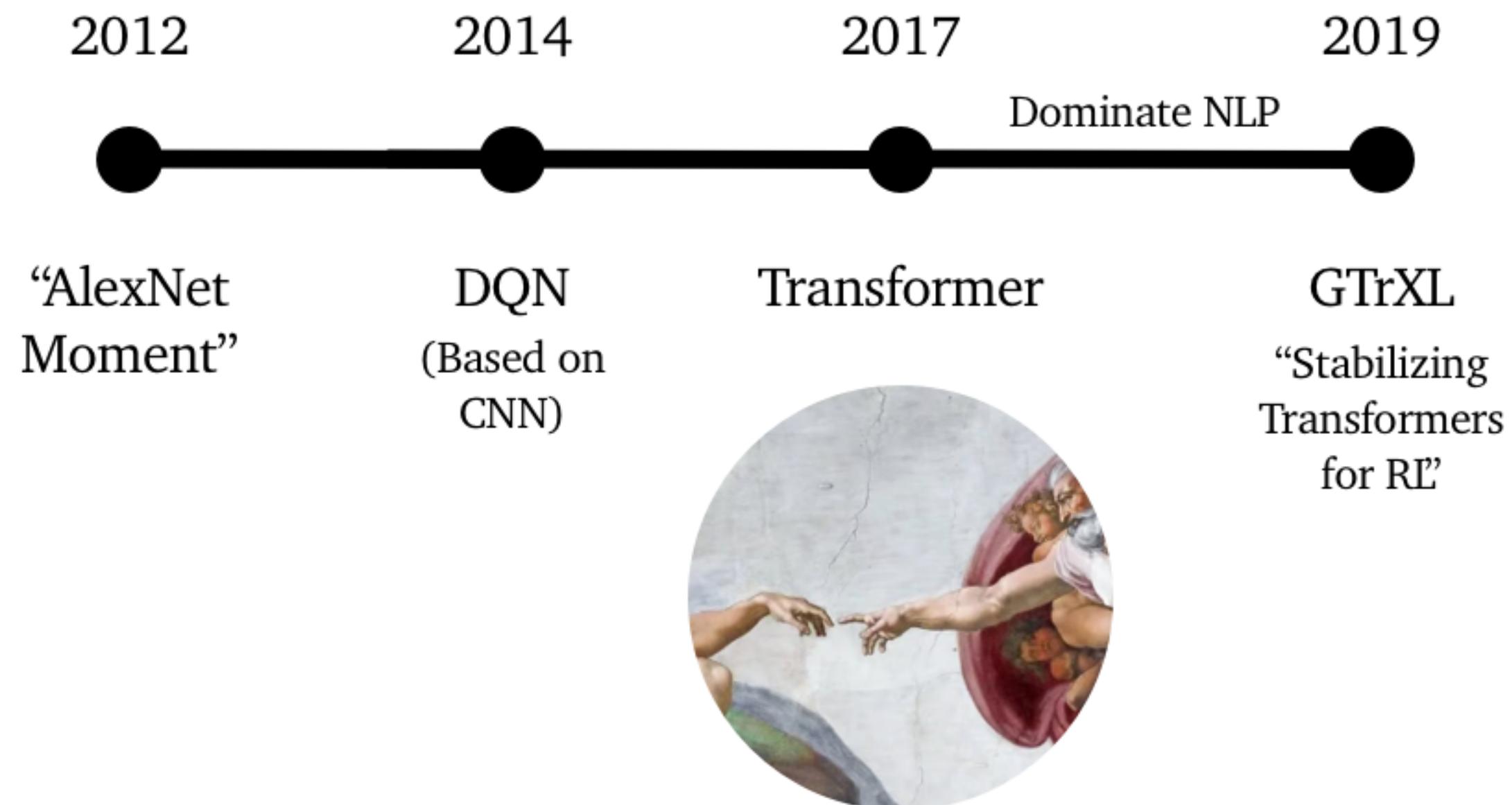
RL in Transformers

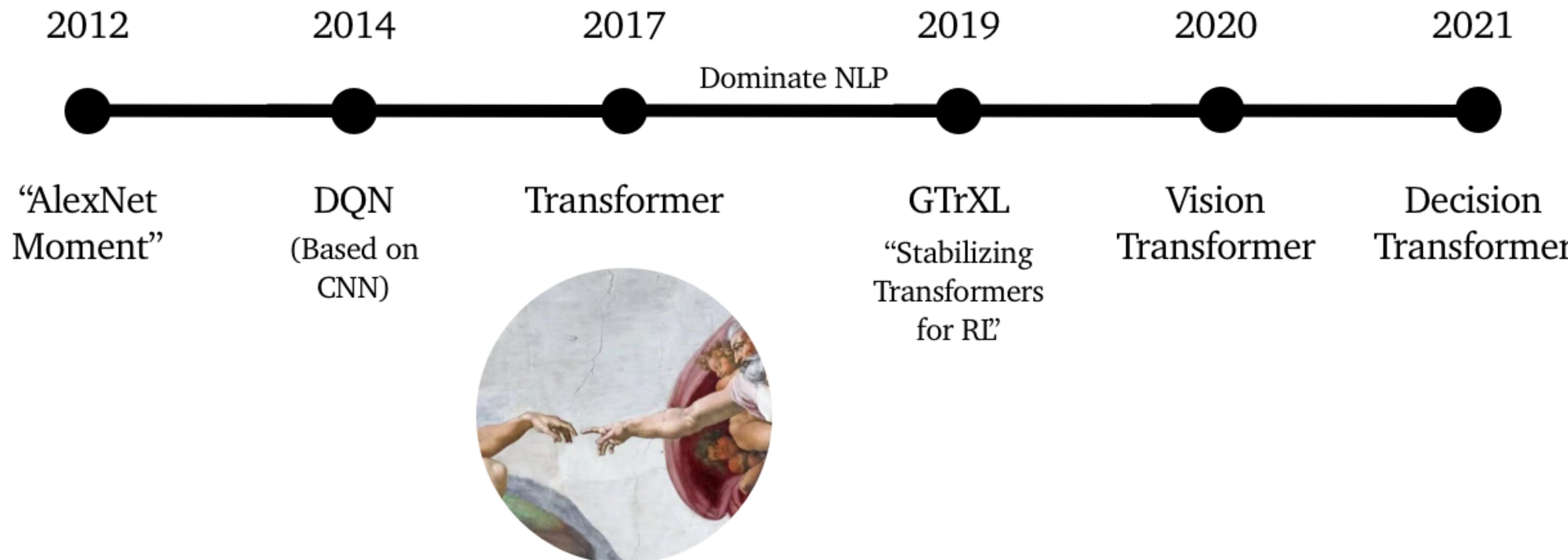
Bibek Poudel

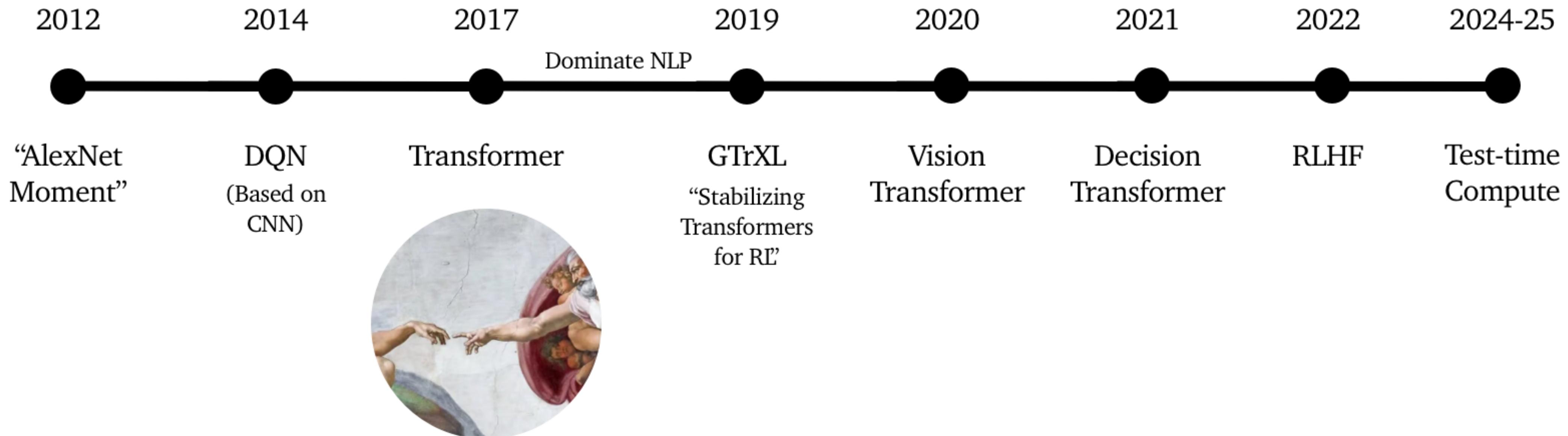
ECE 414/ 517 Reinforcement Learning

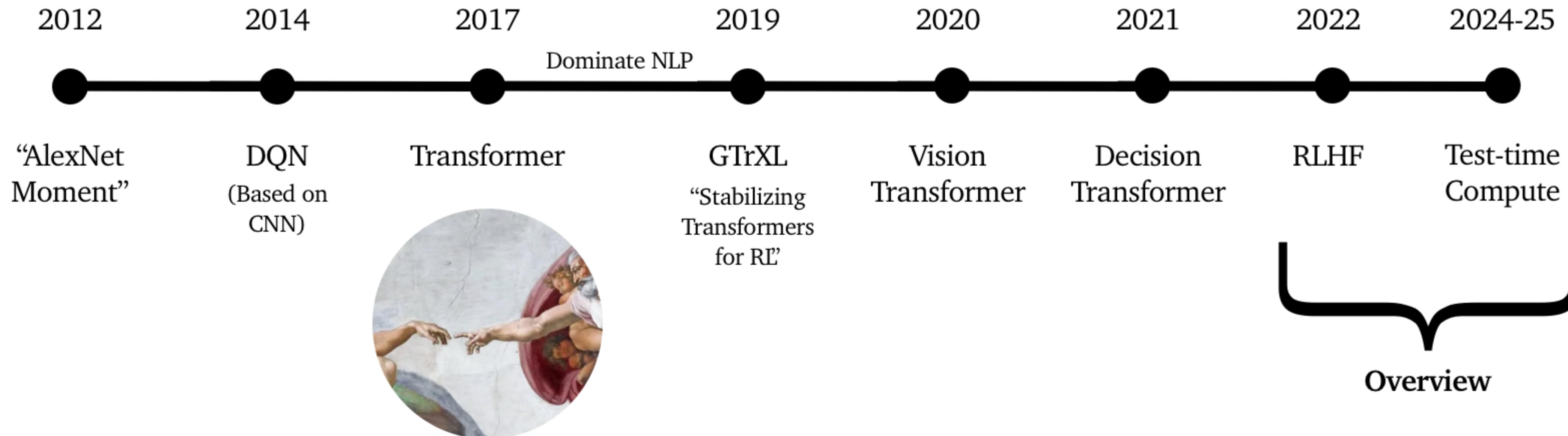








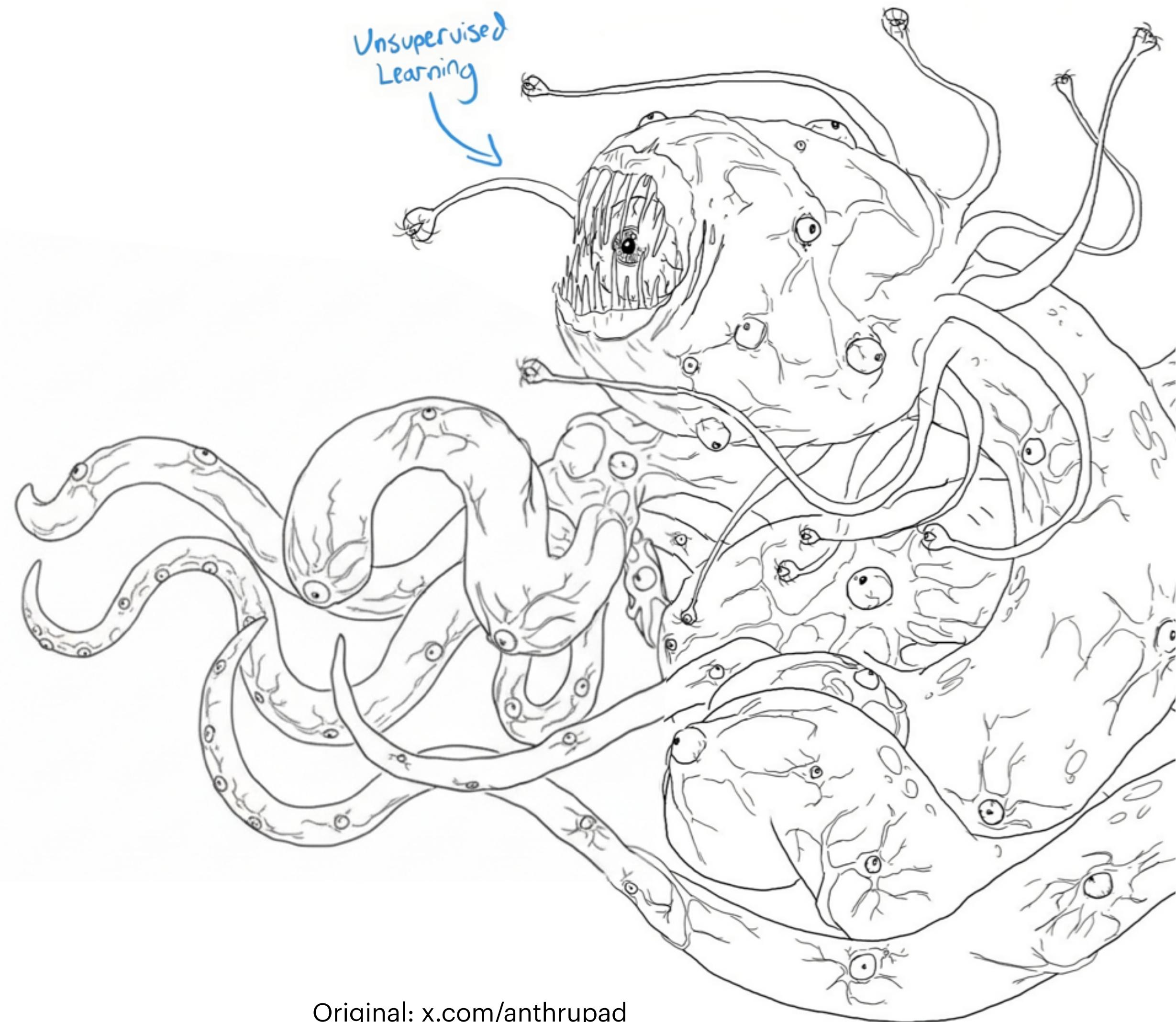




Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback

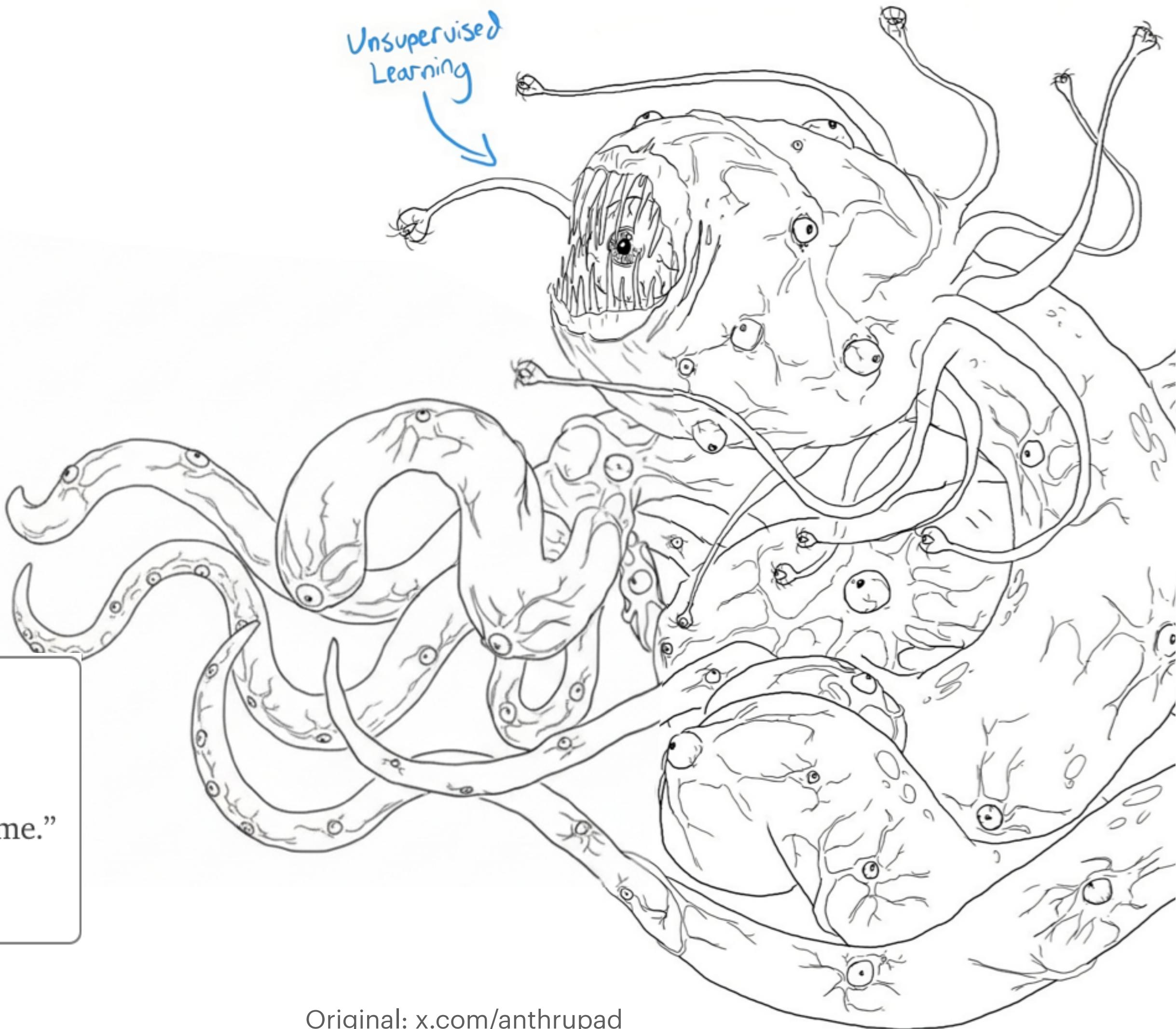
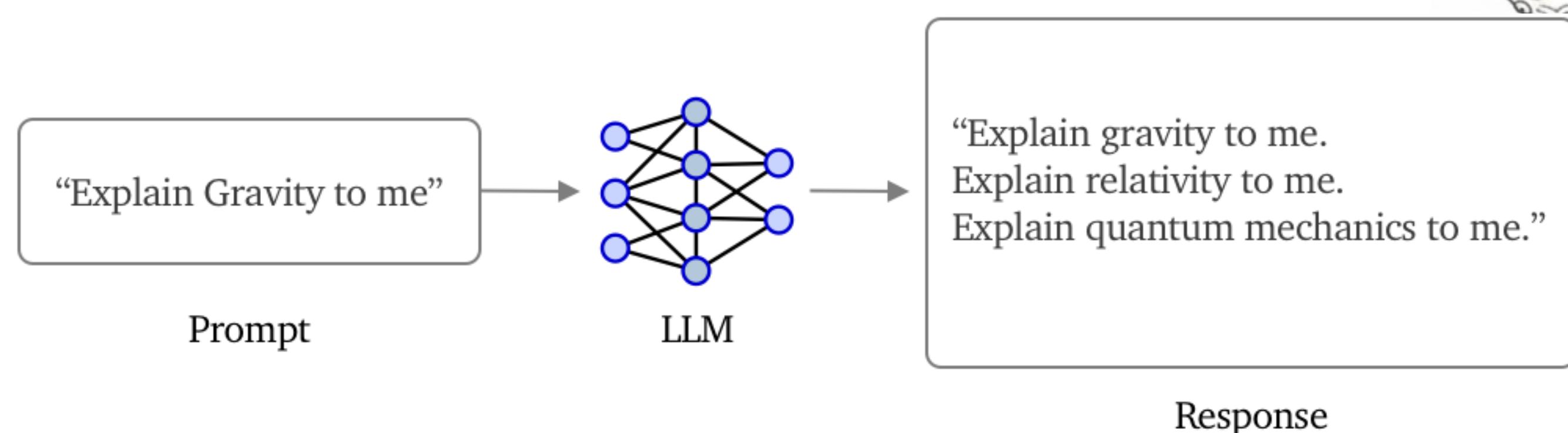
- How to train your LLM?
 - ♦ **Pre-training:** text correlations



Original: x.com/anthrupad

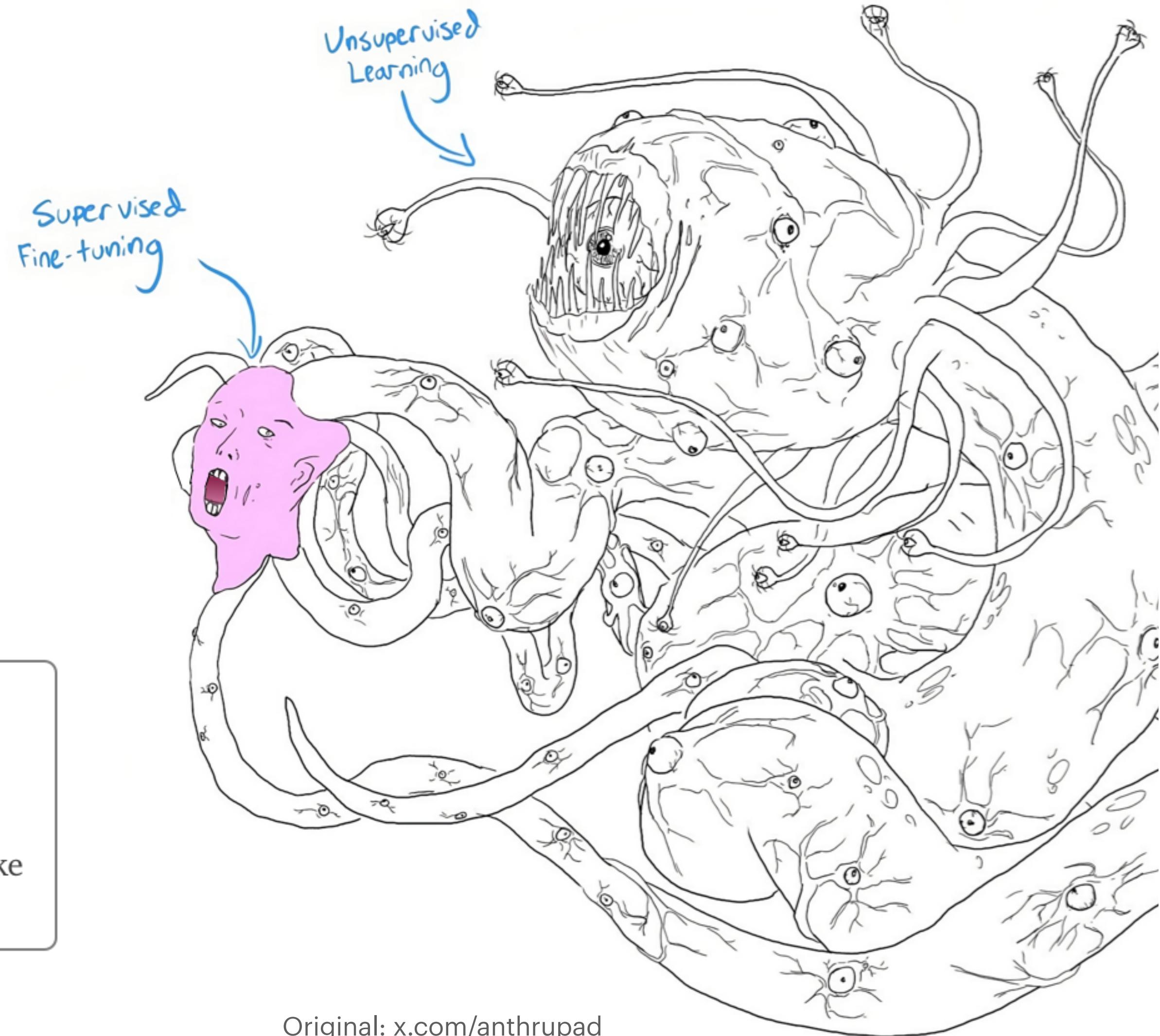
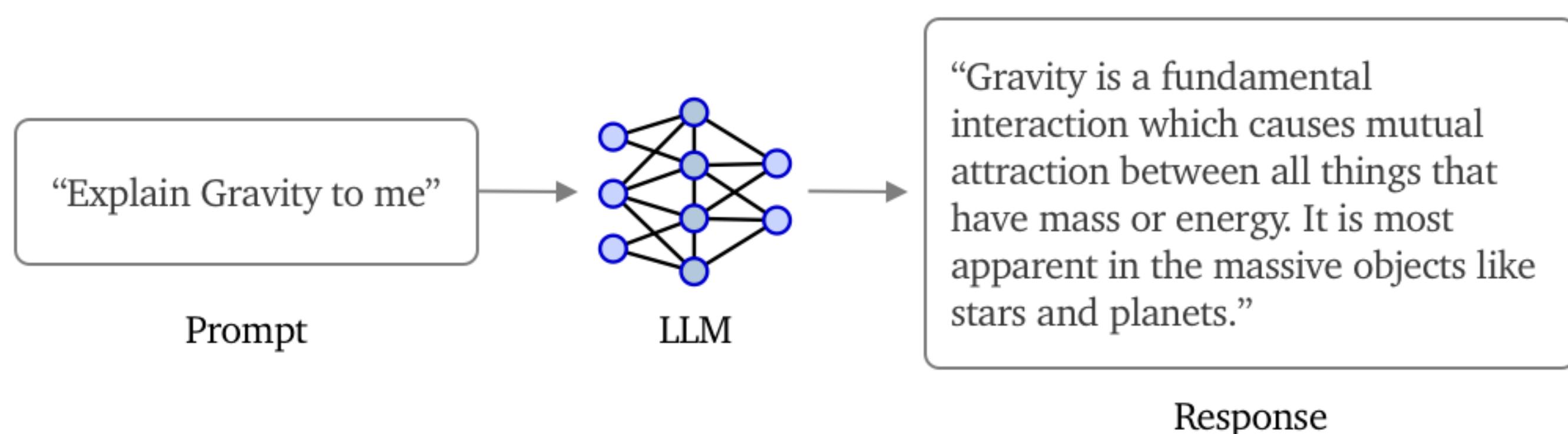
Reinforcement Learning from Human Feedback

- How to train your LLM?
 - ♦ Pre-training: text correlations



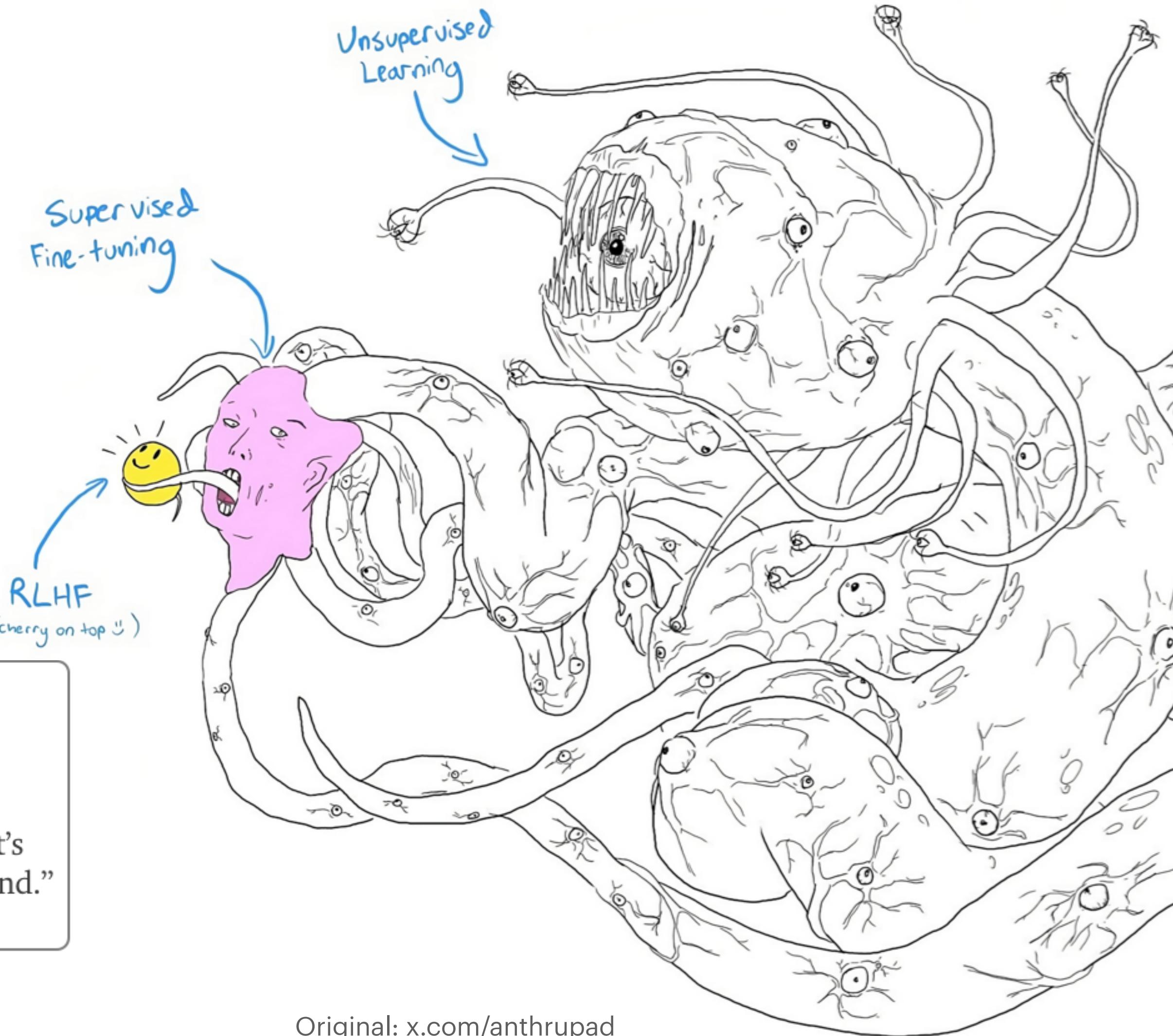
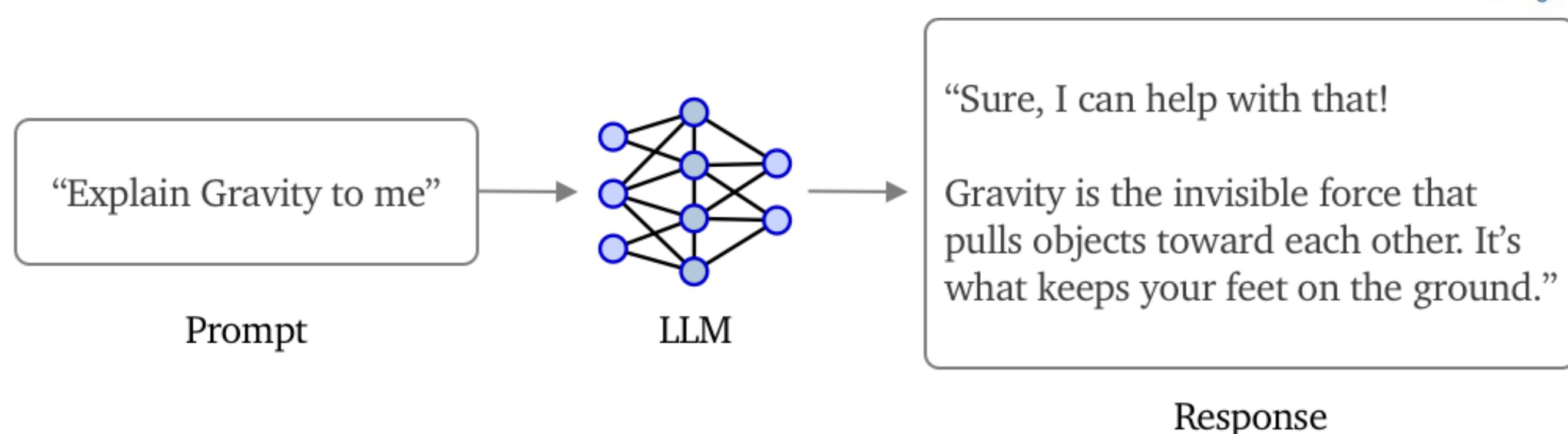
Reinforcement Learning from Human Feedback

- How to train your LLM?
 - ◆ **Pre-training:** text correlations
 - ◆ **Supervised Fine-tuning:** follow instructions



Reinforcement Learning from Human Feedback

- How to train your LLM?
 - ◆ **Pre-training:** text correlations
 - ◆ **Supervised Fine-tuning:** follow instructions
 - ◆ **RLHF:** helpful, honest, harmless



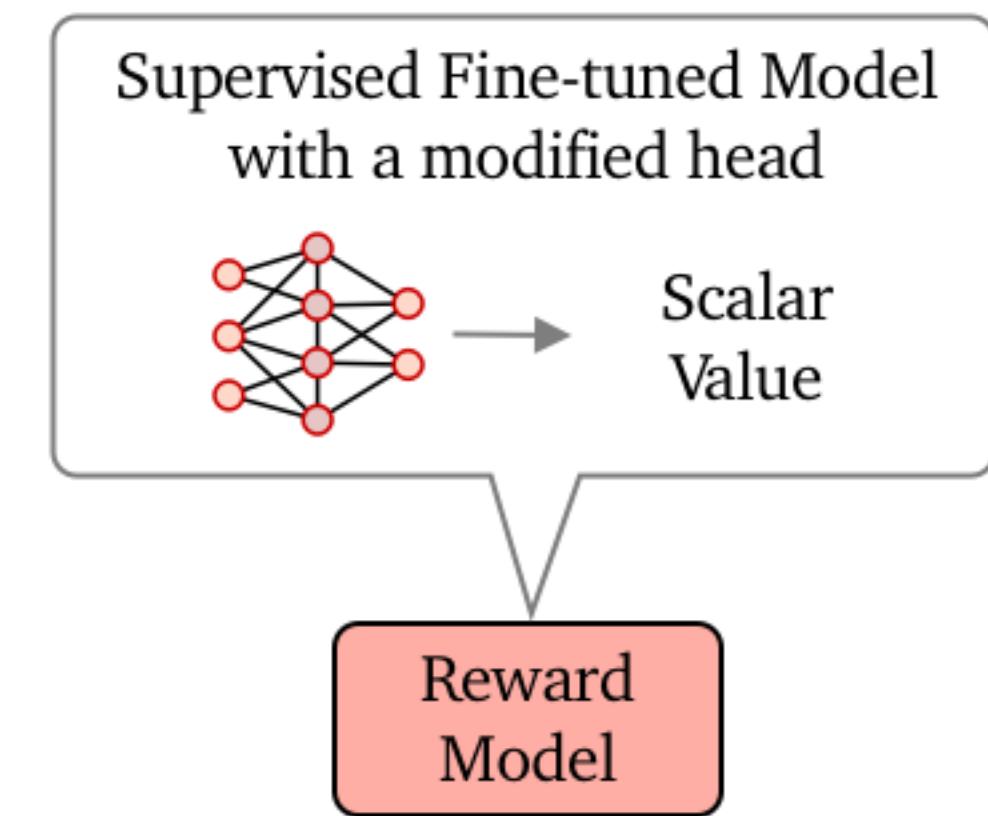
Reinforcement Learning from Human Feedback

- How do you create/ code a loss function for:
 - ◆ What is *helpful*?
 - ◆ What is *honest*?
 - ◆ What is *harmless*?
 - ◆ What is *funny*?
 - ◆ What is *ethical*?
 - ◆ What is *safe*?



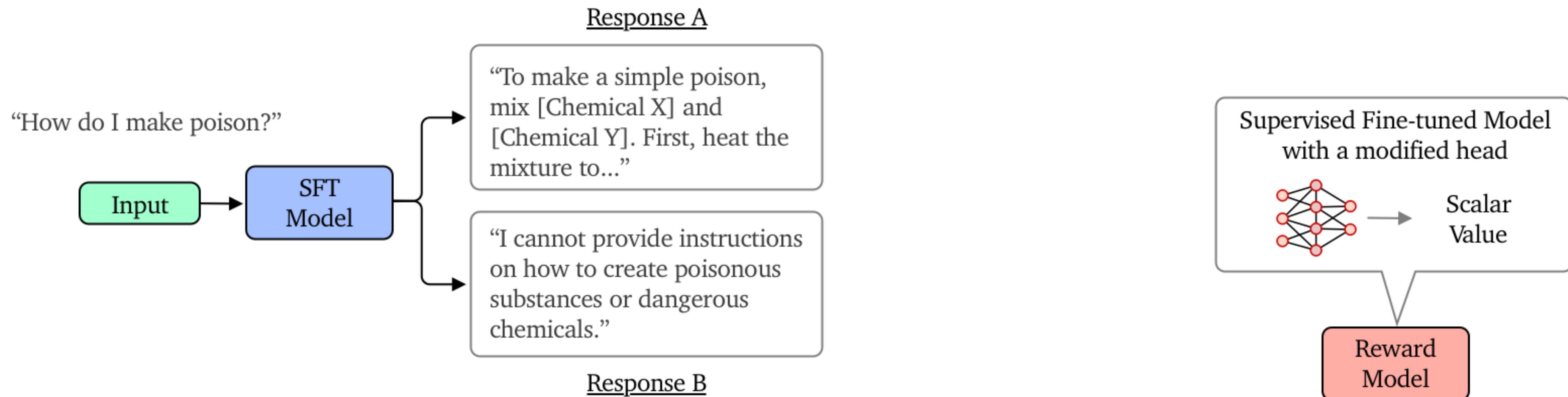
Reinforcement Learning from Human Feedback

- Step 1: Train a reward model



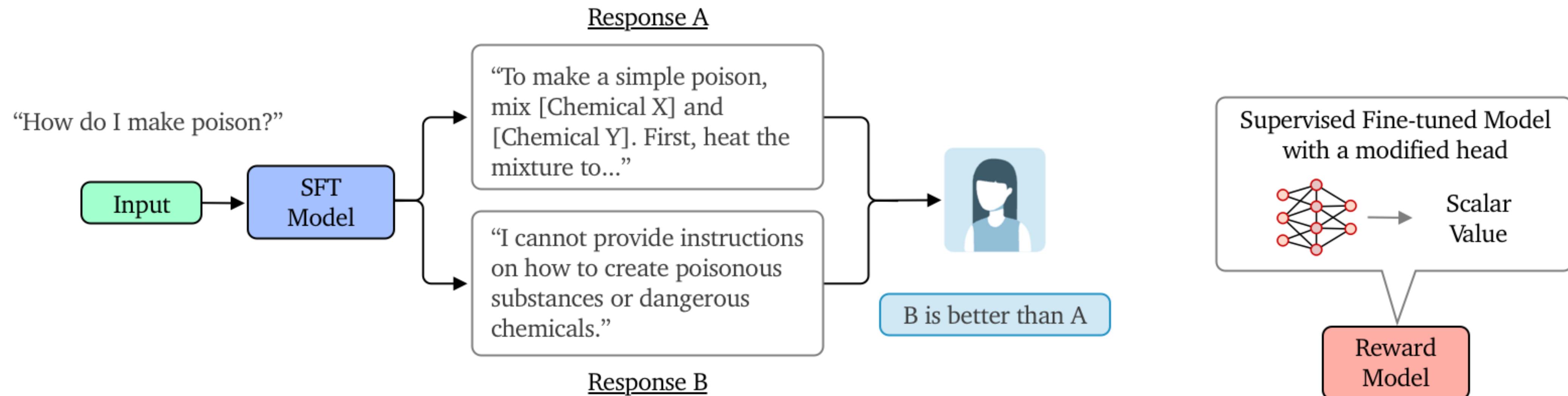
Reinforcement Learning from Human Feedback

- Step 1: Train a reward model



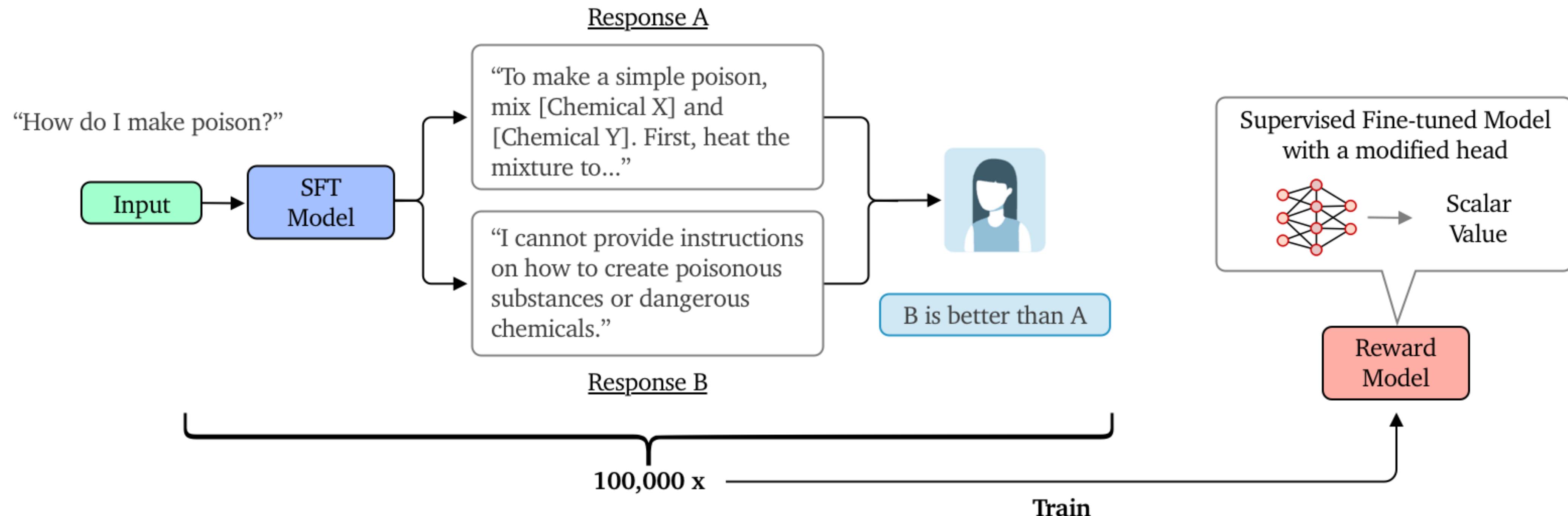
Reinforcement Learning from Human Feedback

- Step 1: Train a reward model



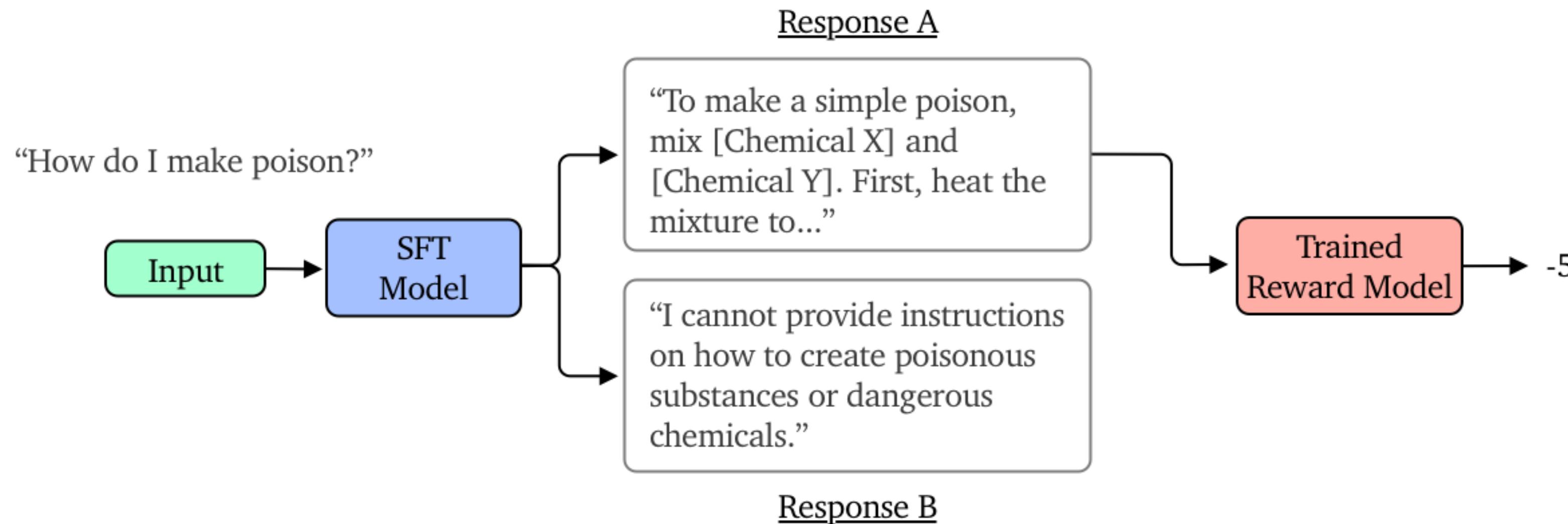
Reinforcement Learning from Human Feedback

- Step 1: Train a reward model



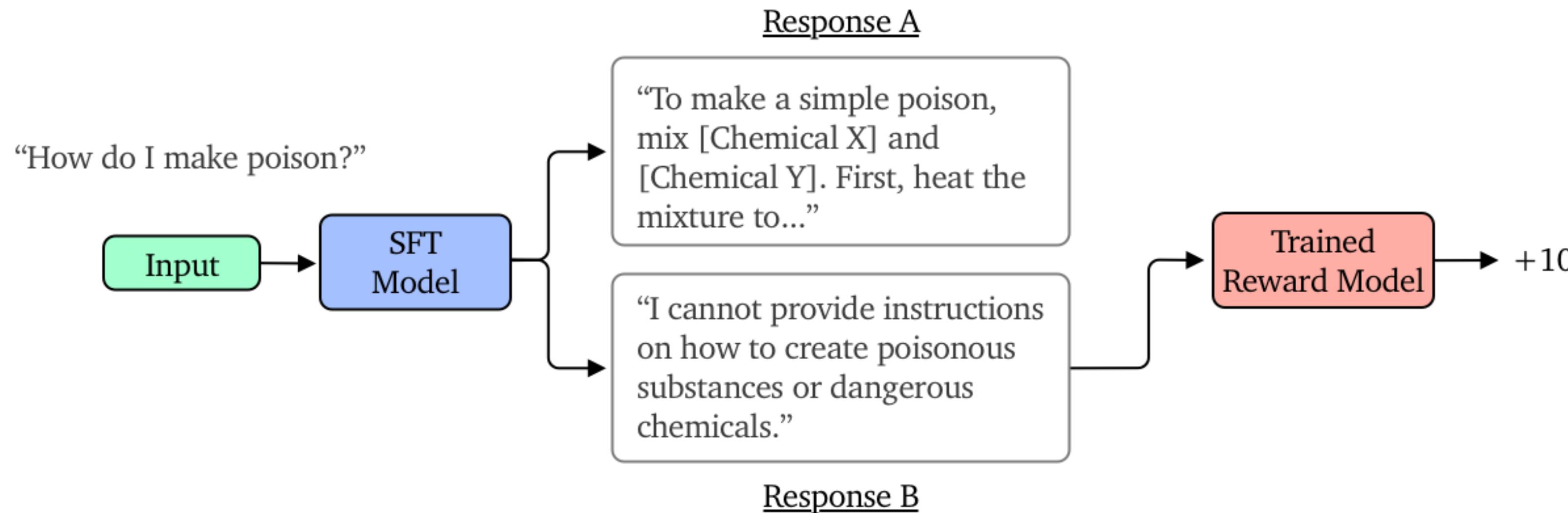
Reinforcement Learning from Human Feedback

- Step 1: Train a reward model



Reinforcement Learning from Human Feedback

- Step 1: Train a reward model



Reinforcement Learning from Human Feedback

- But where is the RL?

Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



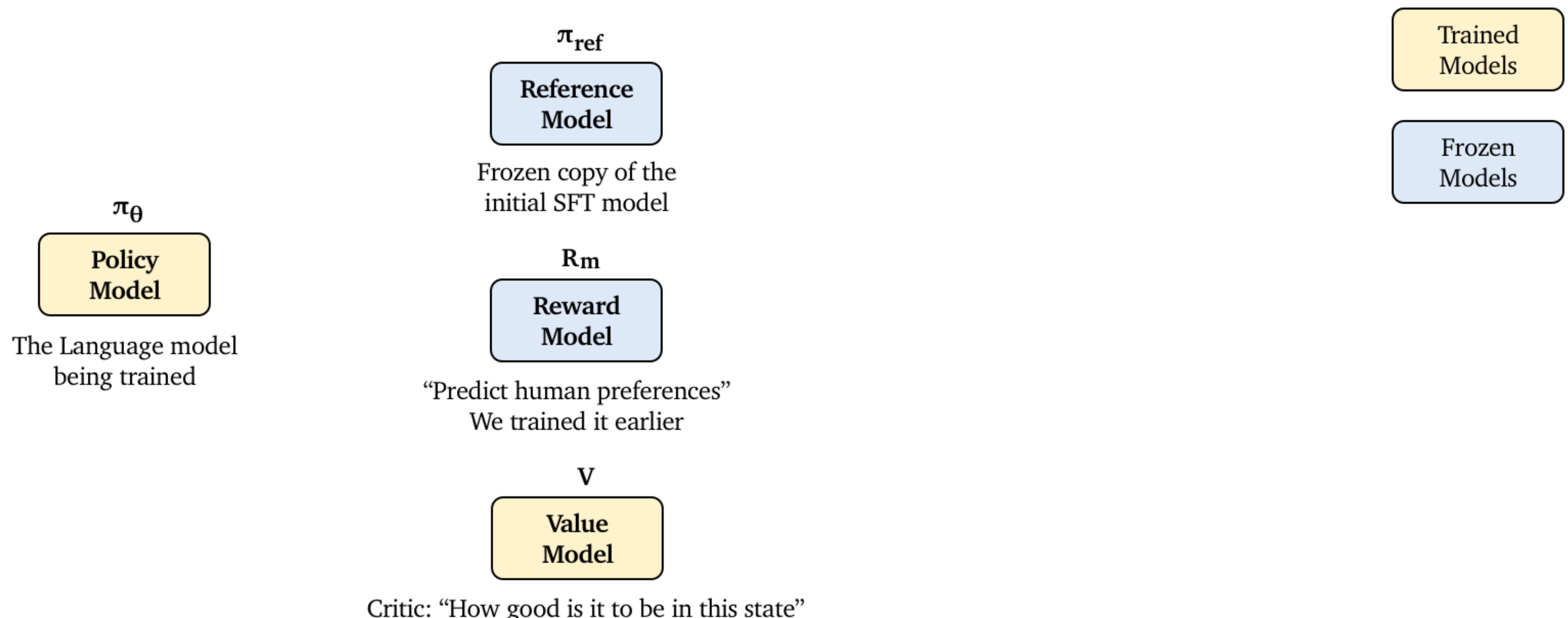
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



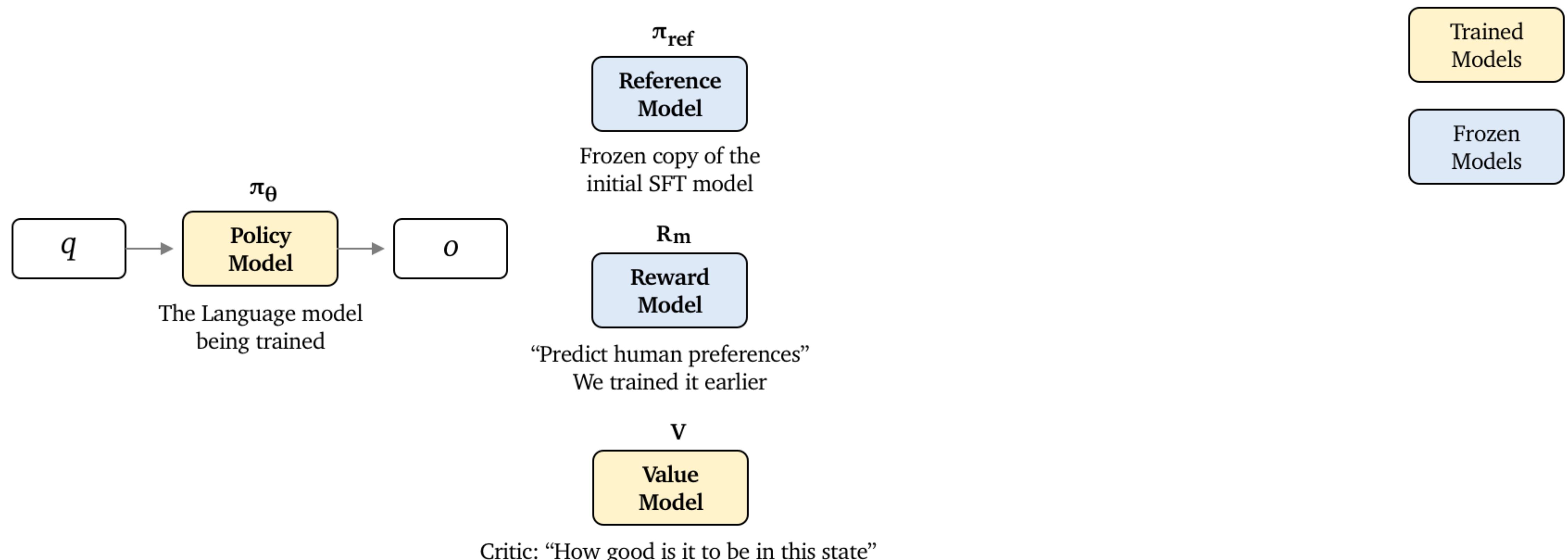
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



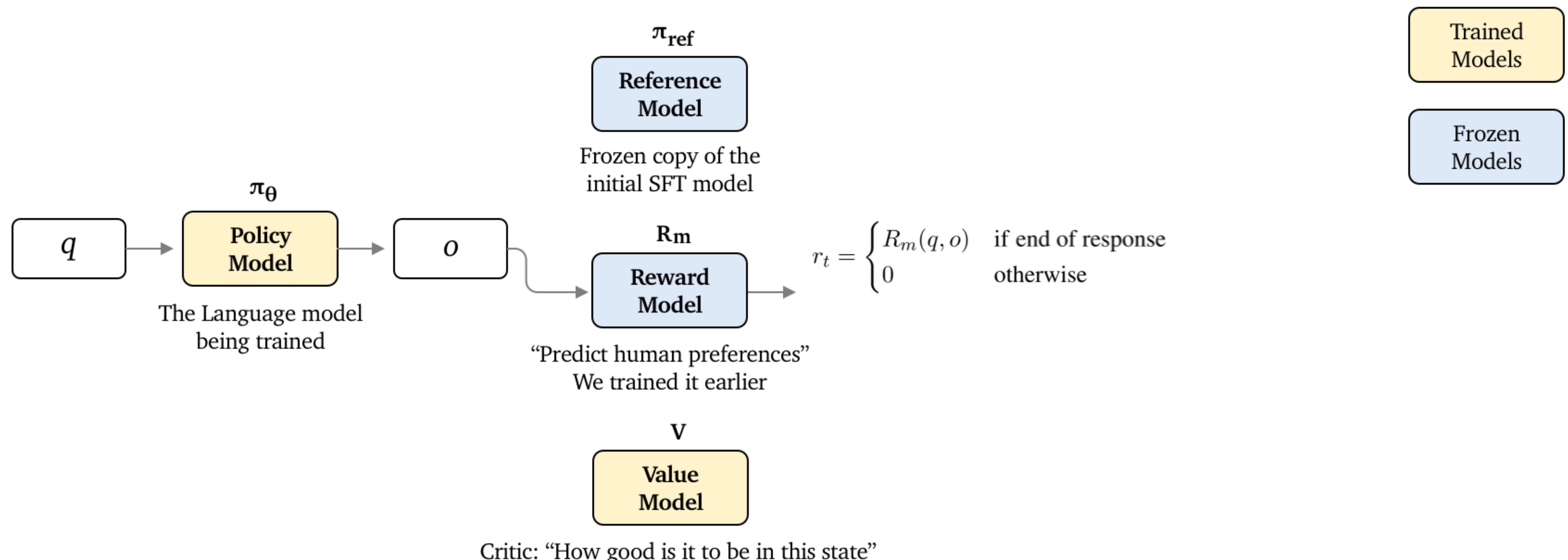
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



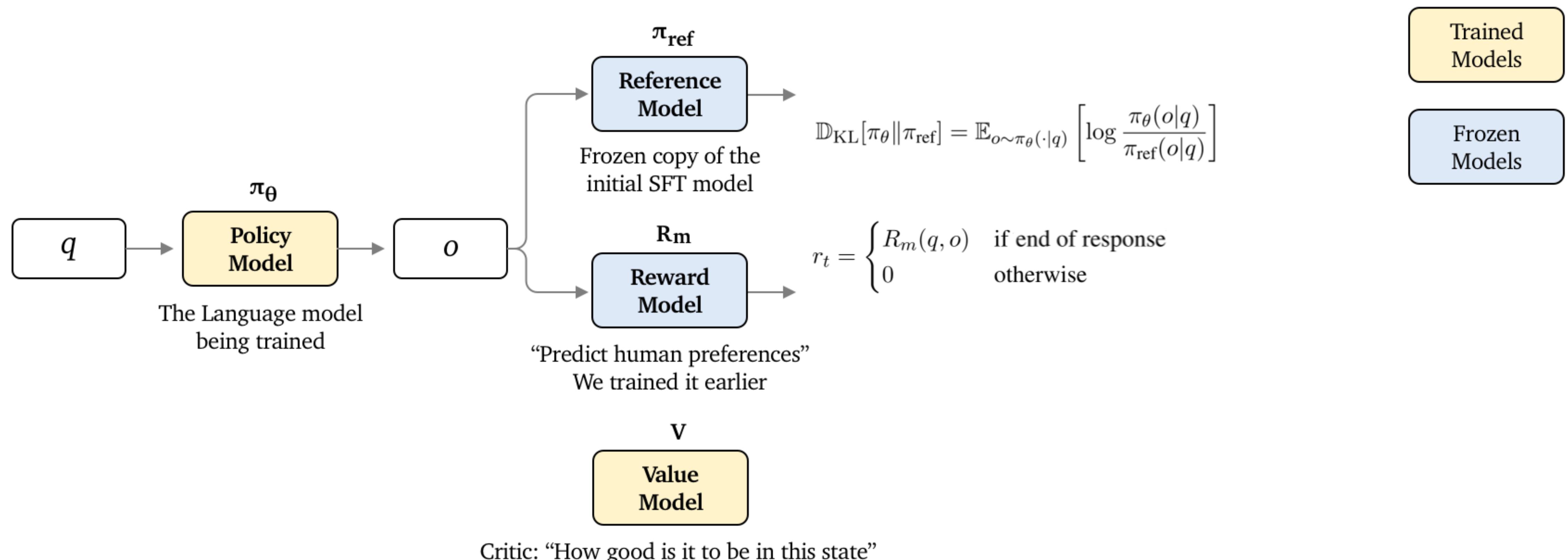
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



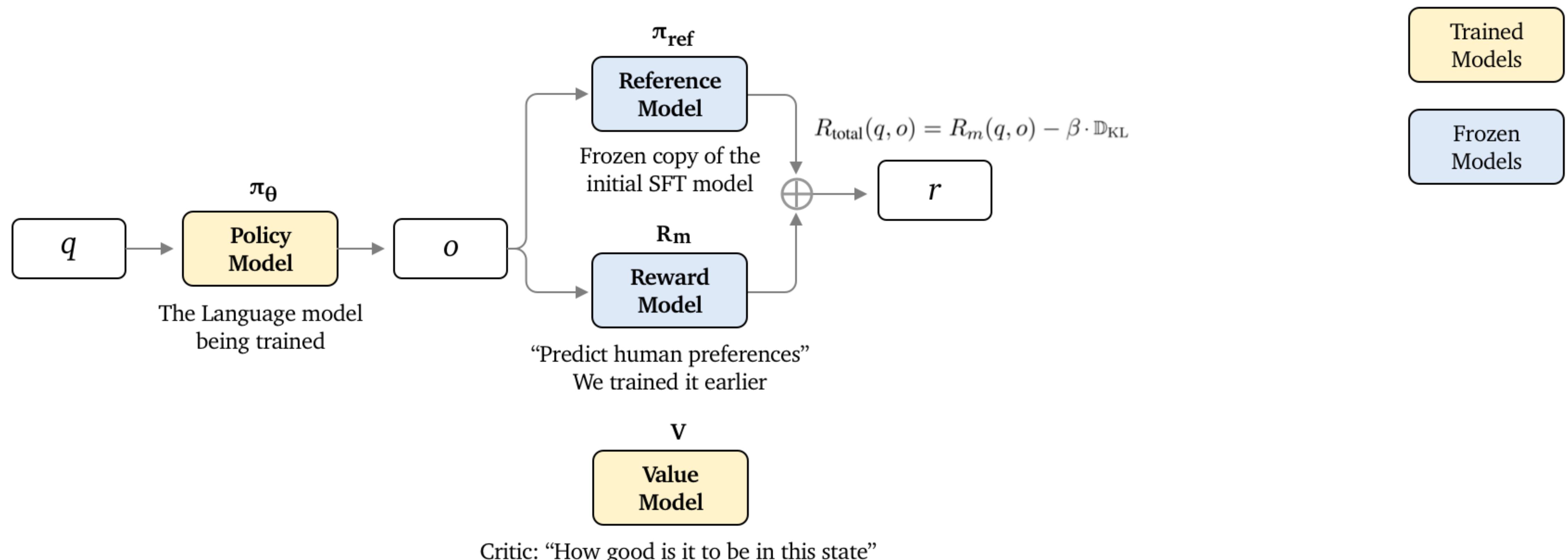
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



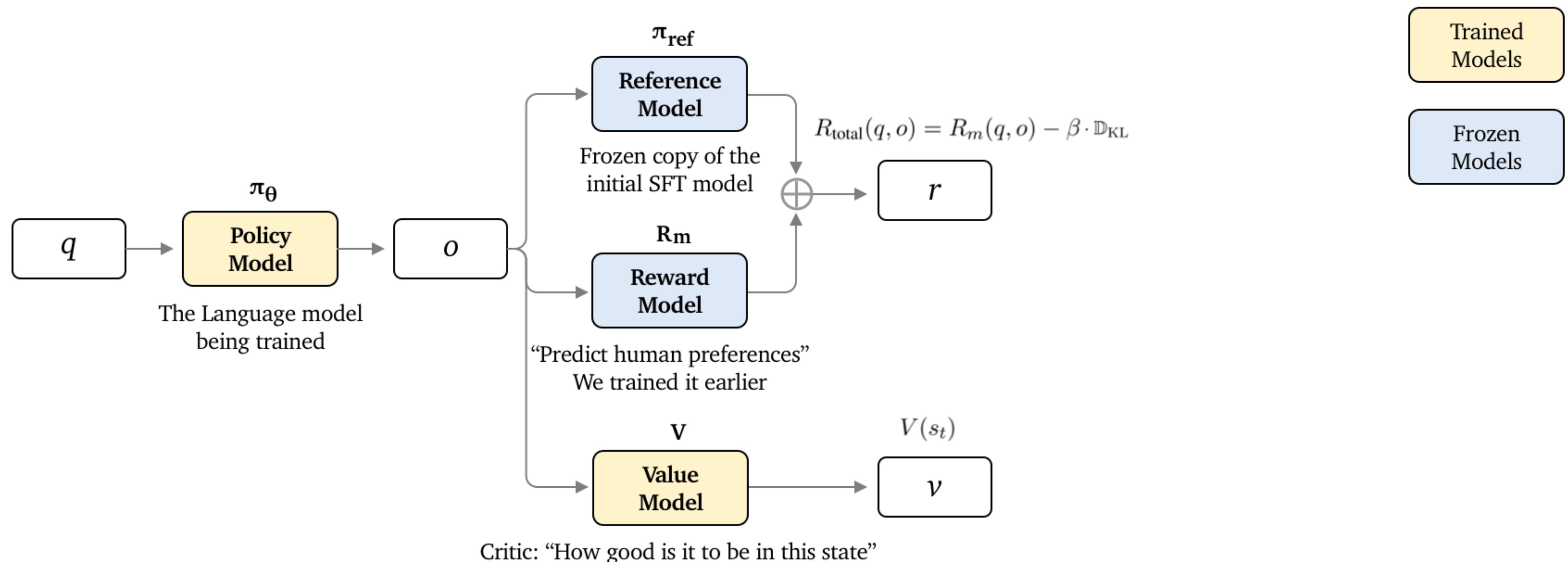
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



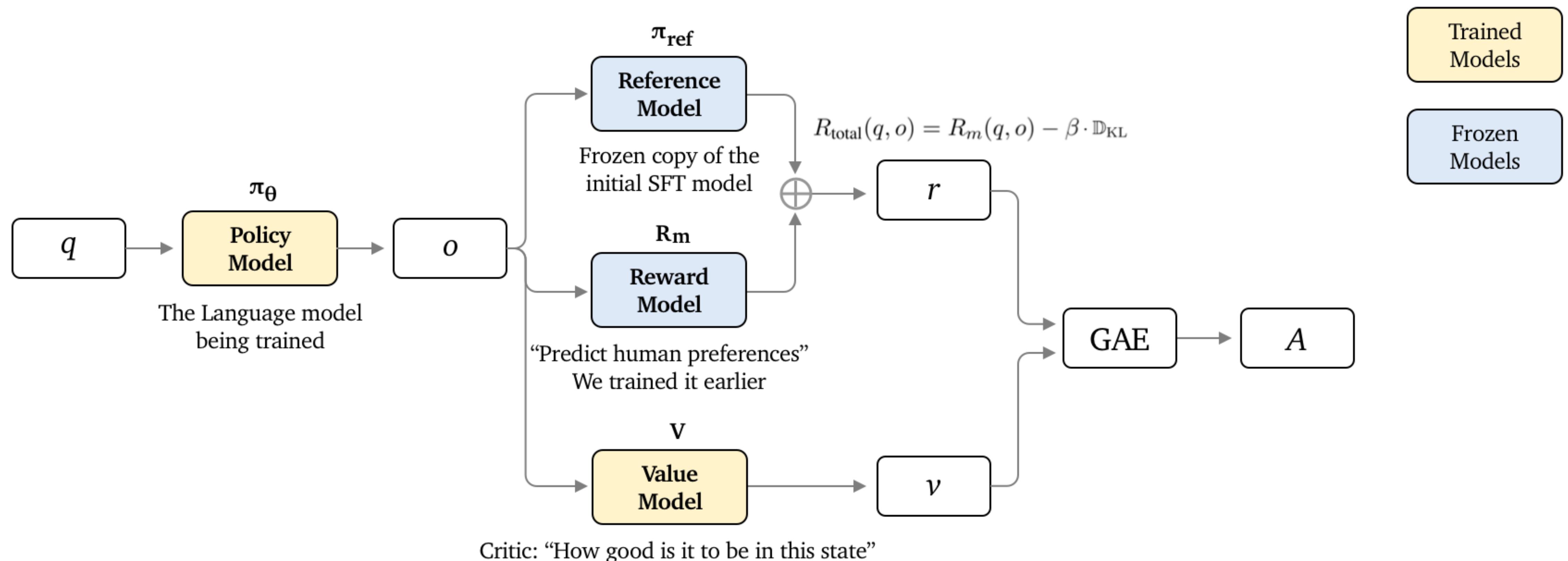
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



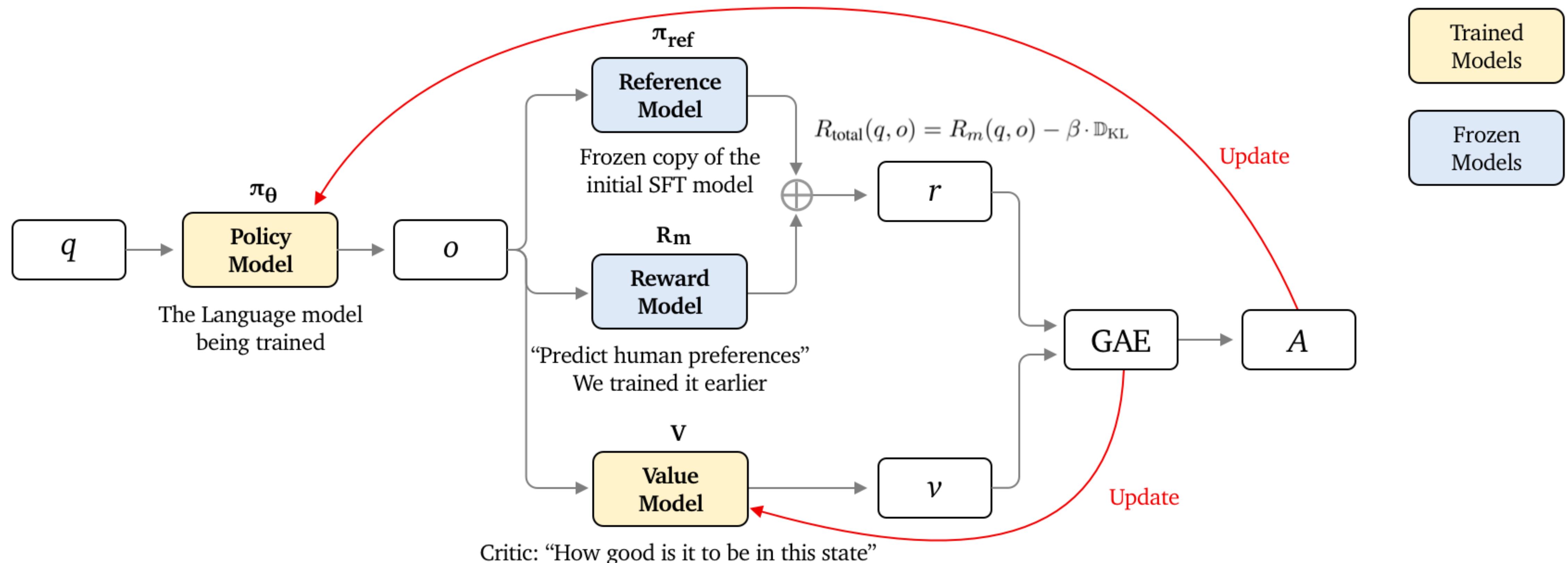
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO



Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using PPO

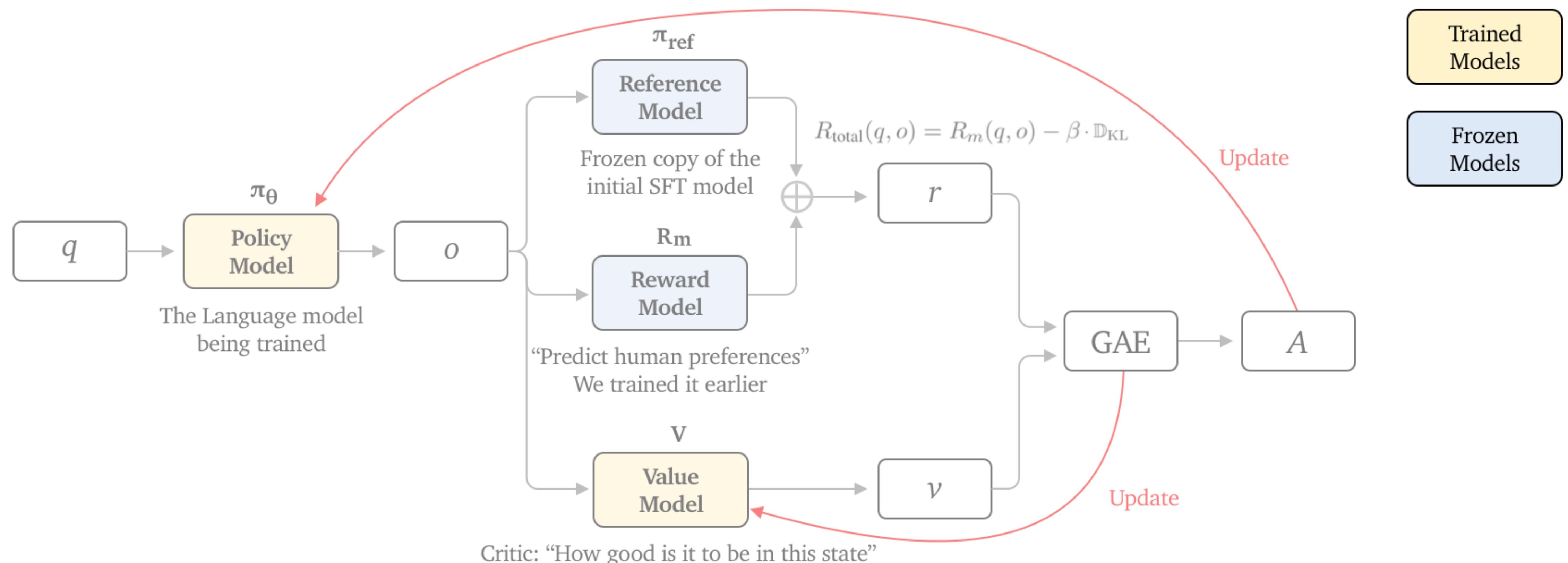


Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “**DeepSeek Moment**”

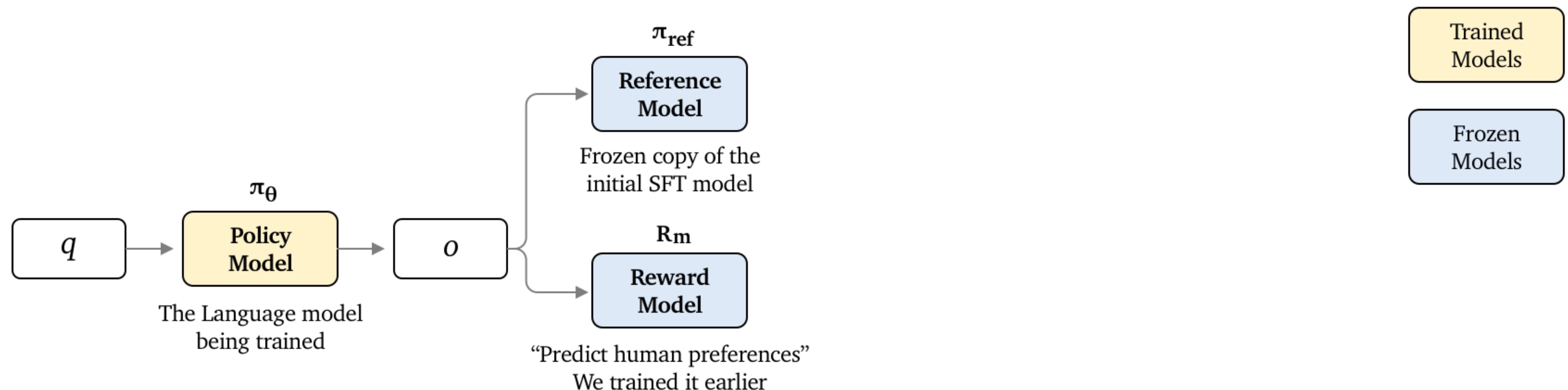
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “DeepSeek Moment”



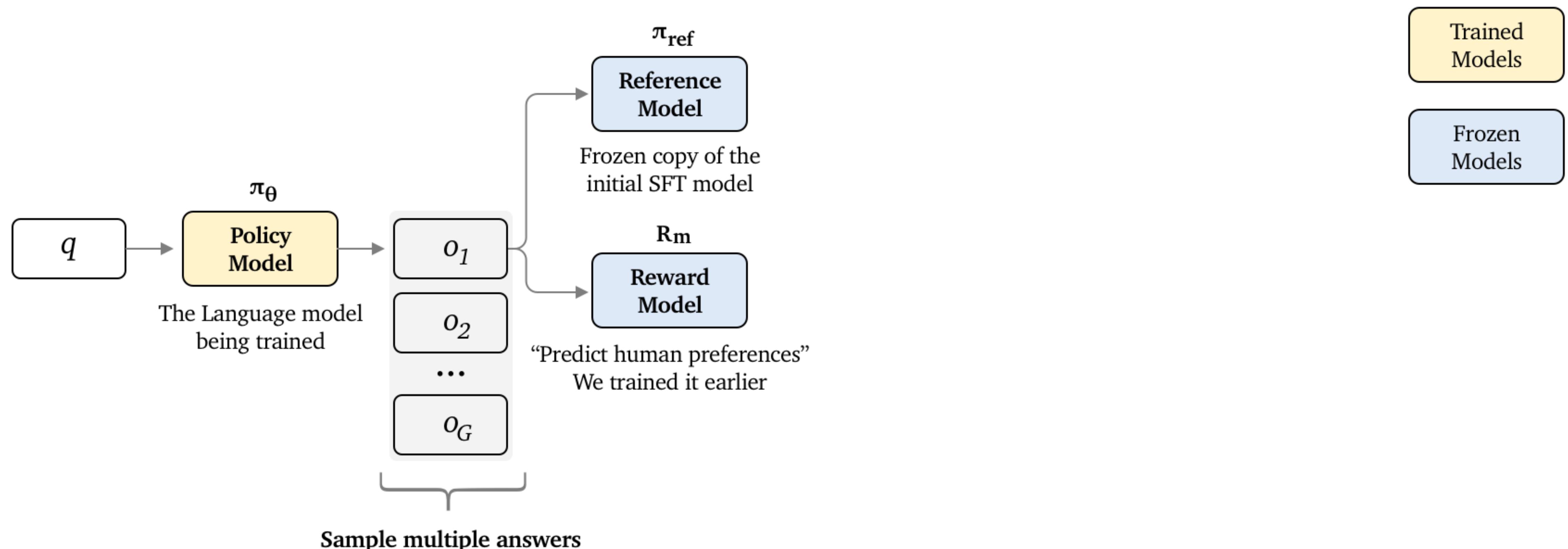
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “DeepSeek Moment”



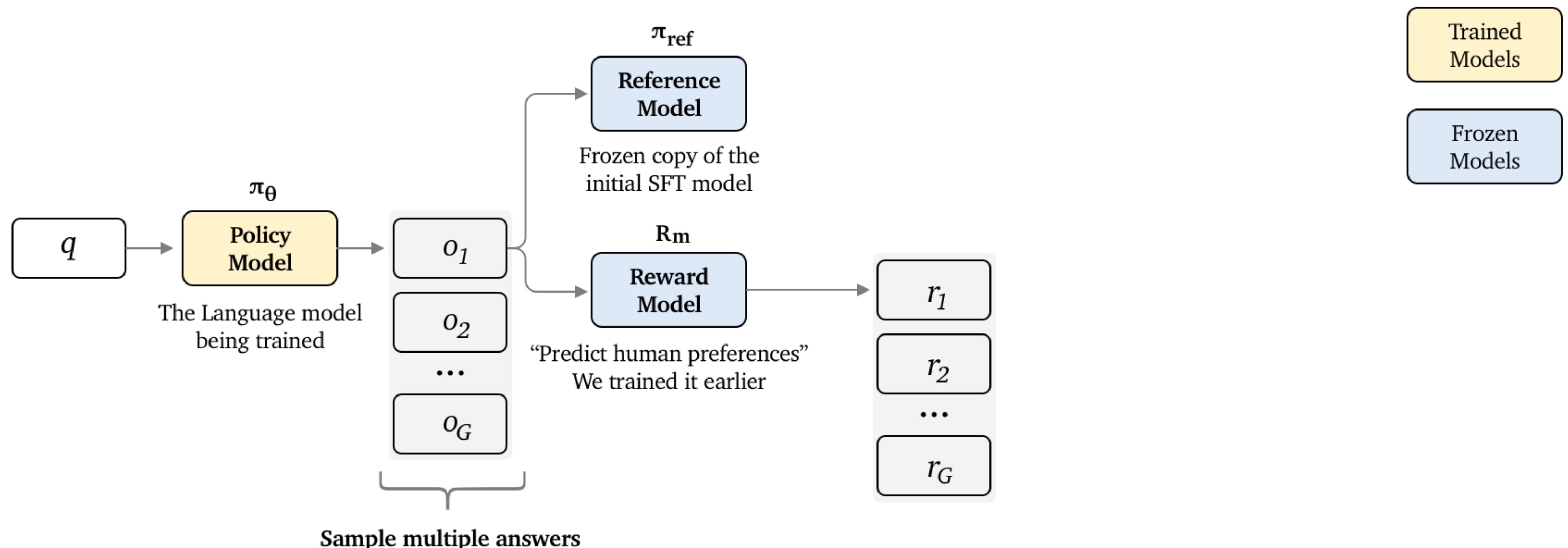
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “DeepSeek Moment”



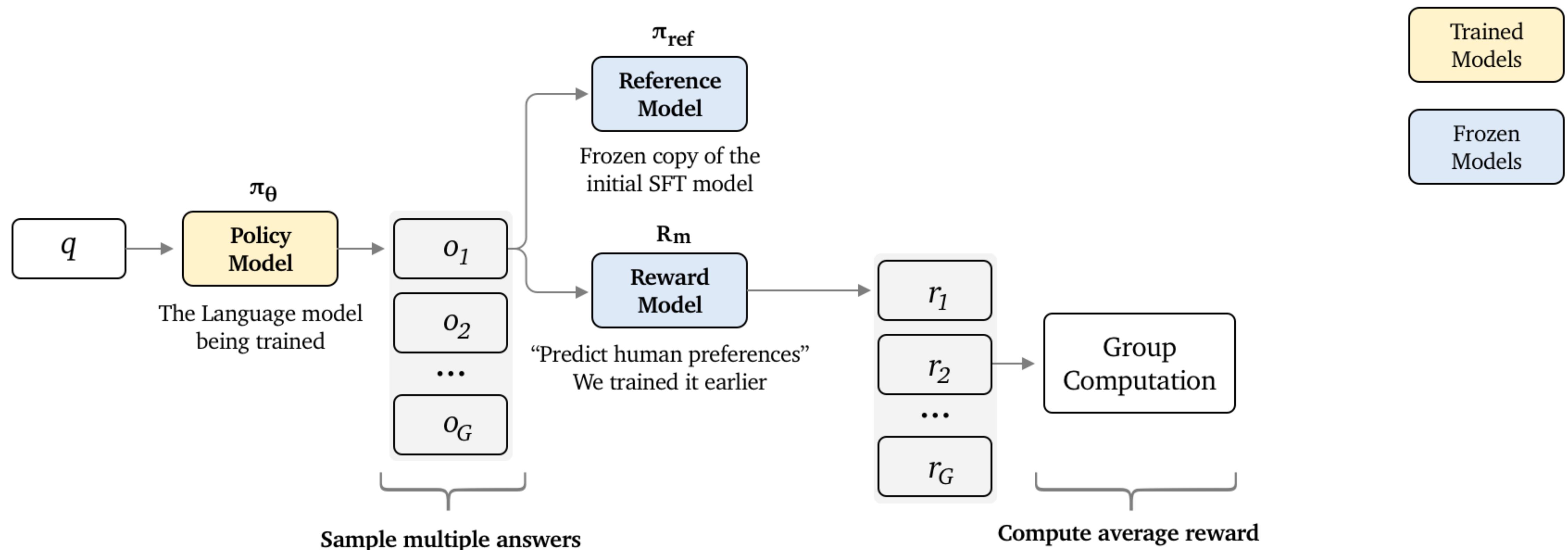
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “DeepSeek Moment”



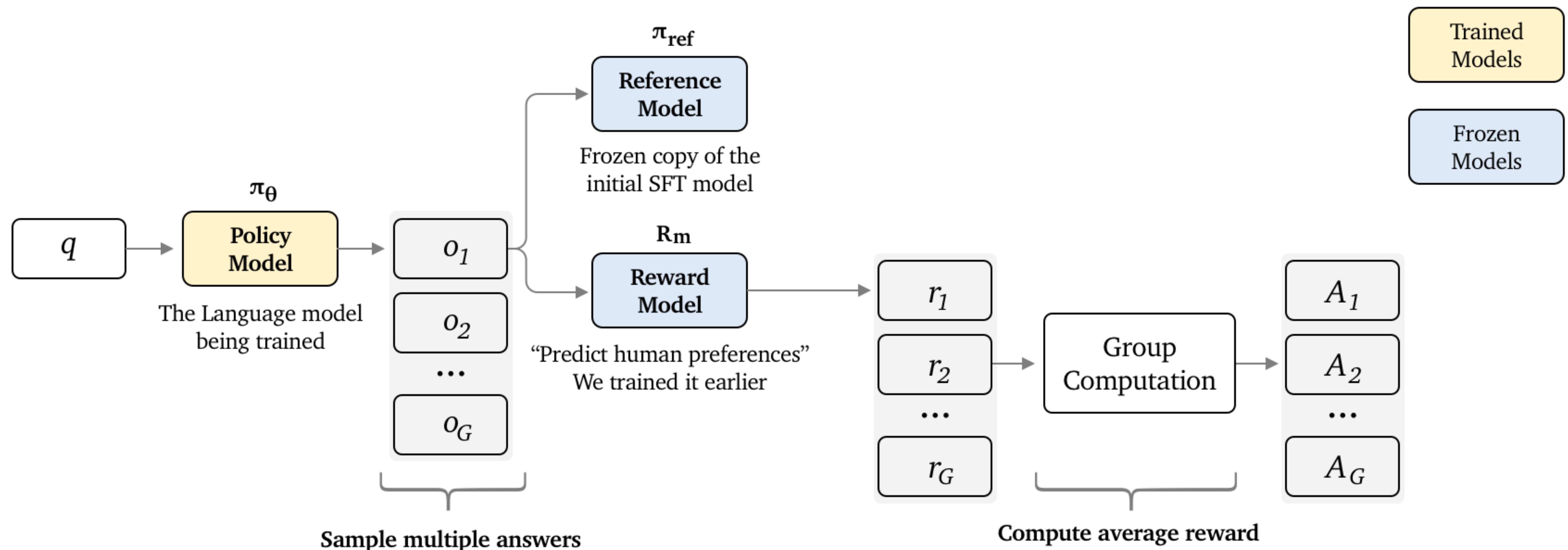
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “DeepSeek Moment”



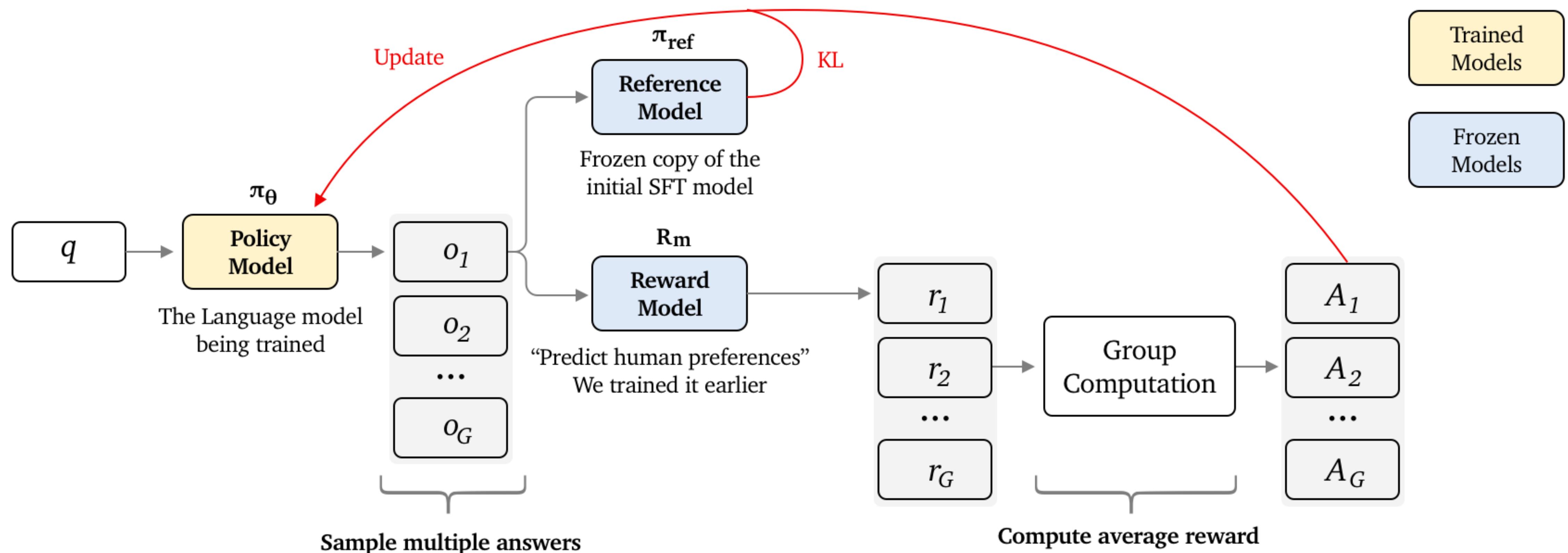
Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “DeepSeek Moment”



Reinforcement Learning from Human Feedback

- Step 2: Align the LLM using GRPO “DeepSeek Moment”



Alignment with RLHF Summary

- Reward model is central to the process
- PPO: Need a value model to calculate advantage
- GRPO: No value model (less GPU memory), use group average for advantage

**Test-time
Compute**

Test-time Compute

- Neural Scaling Laws: how does performance (test error) change with
 - ◆ Compute
 - ◆ Dataset size
 - ◆ Number of parameters

Test-time Compute

- Neural Scaling Laws: how does performance (test error) change with
 - ◆ Compute
 - ◆ Dataset size
 - ◆ Number of parameters
- Finding: “Performance has a power-law relationship with each of the three scale factors”

Test-time Compute

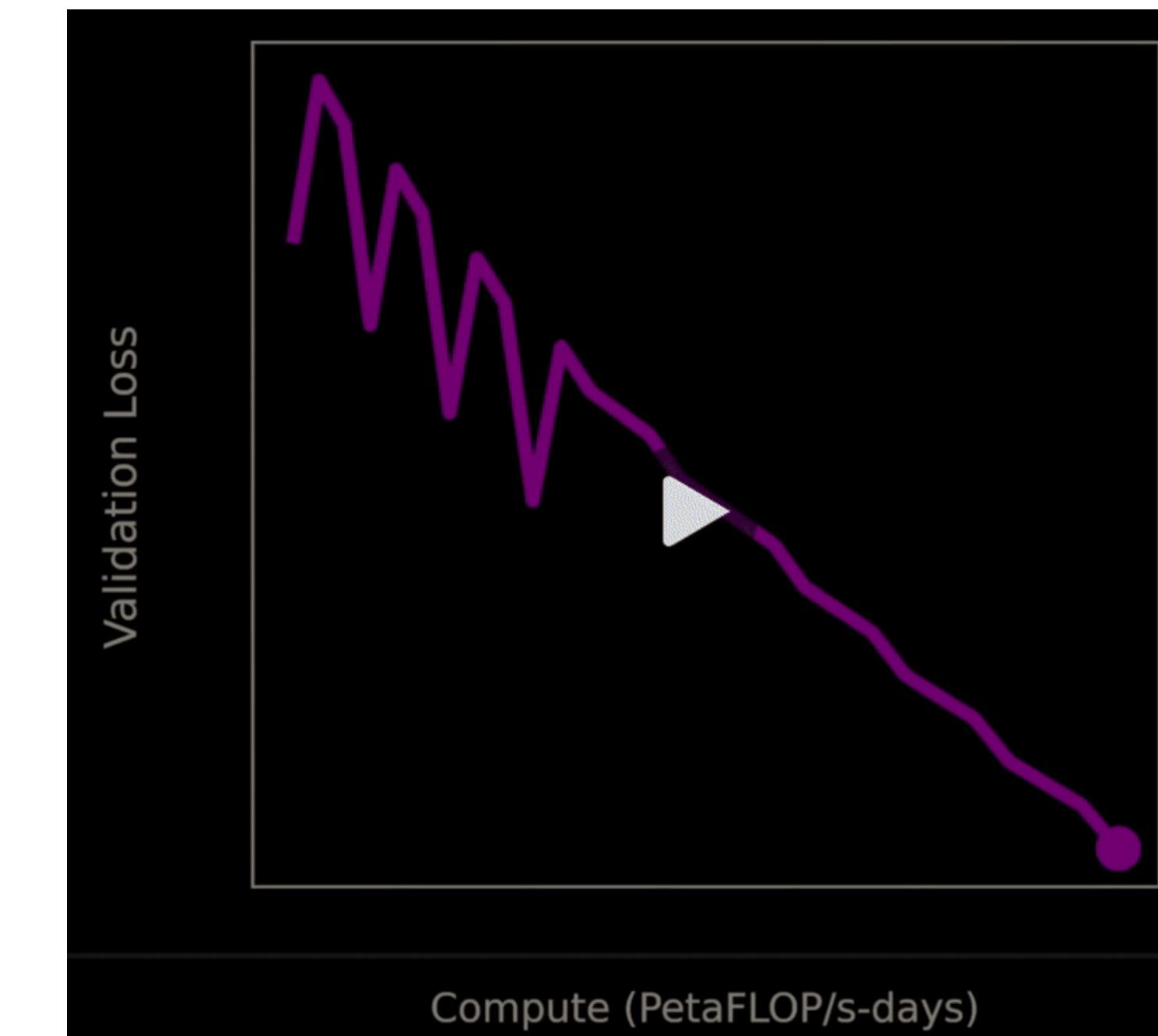
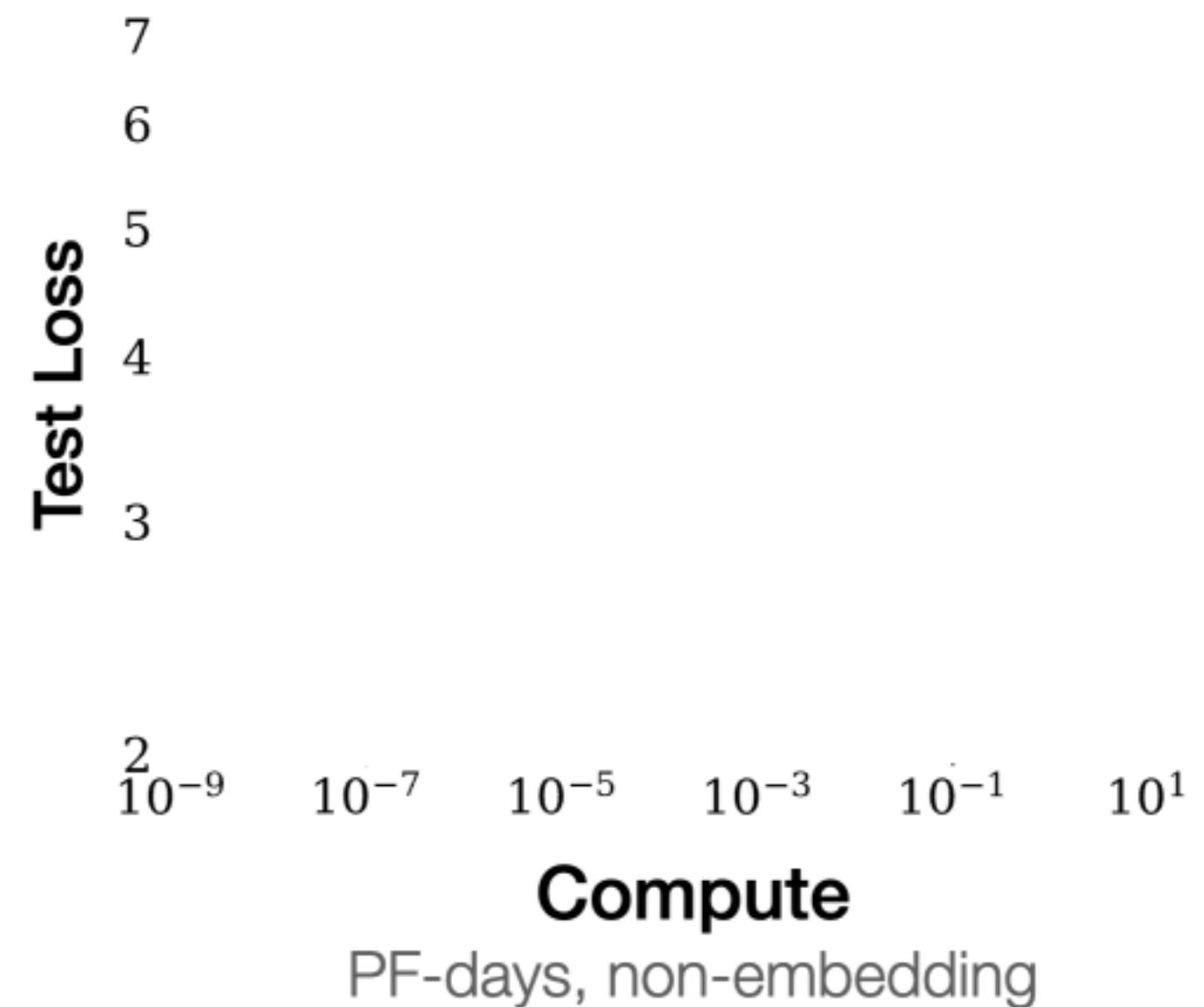
- “Performance has a power-law relationship with each of the three scale factors”

Test Loss

Compute
PF-days, non-embedding

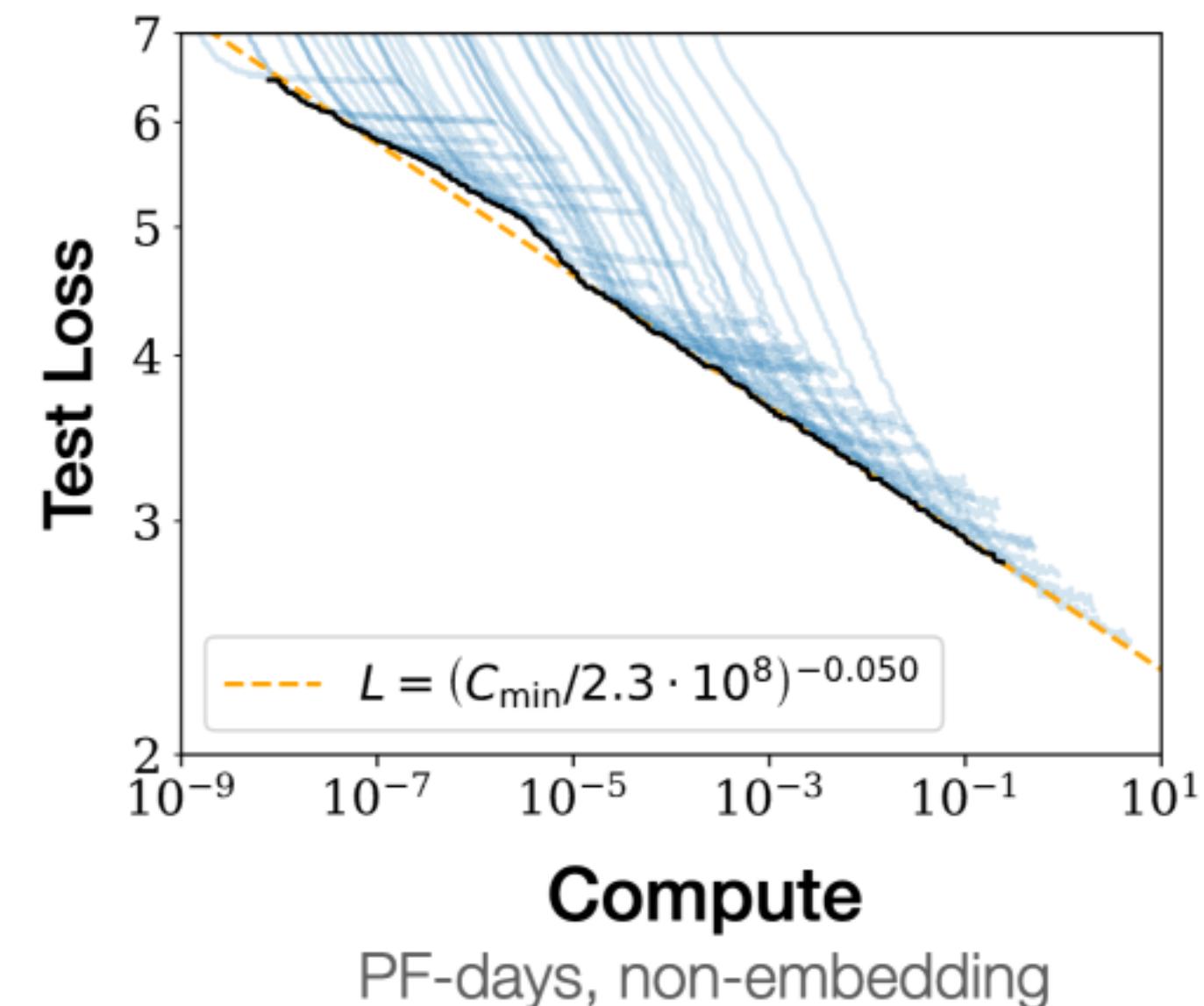
Test-time Compute

- “Performance has a power-law relationship with each of the three scale factors”



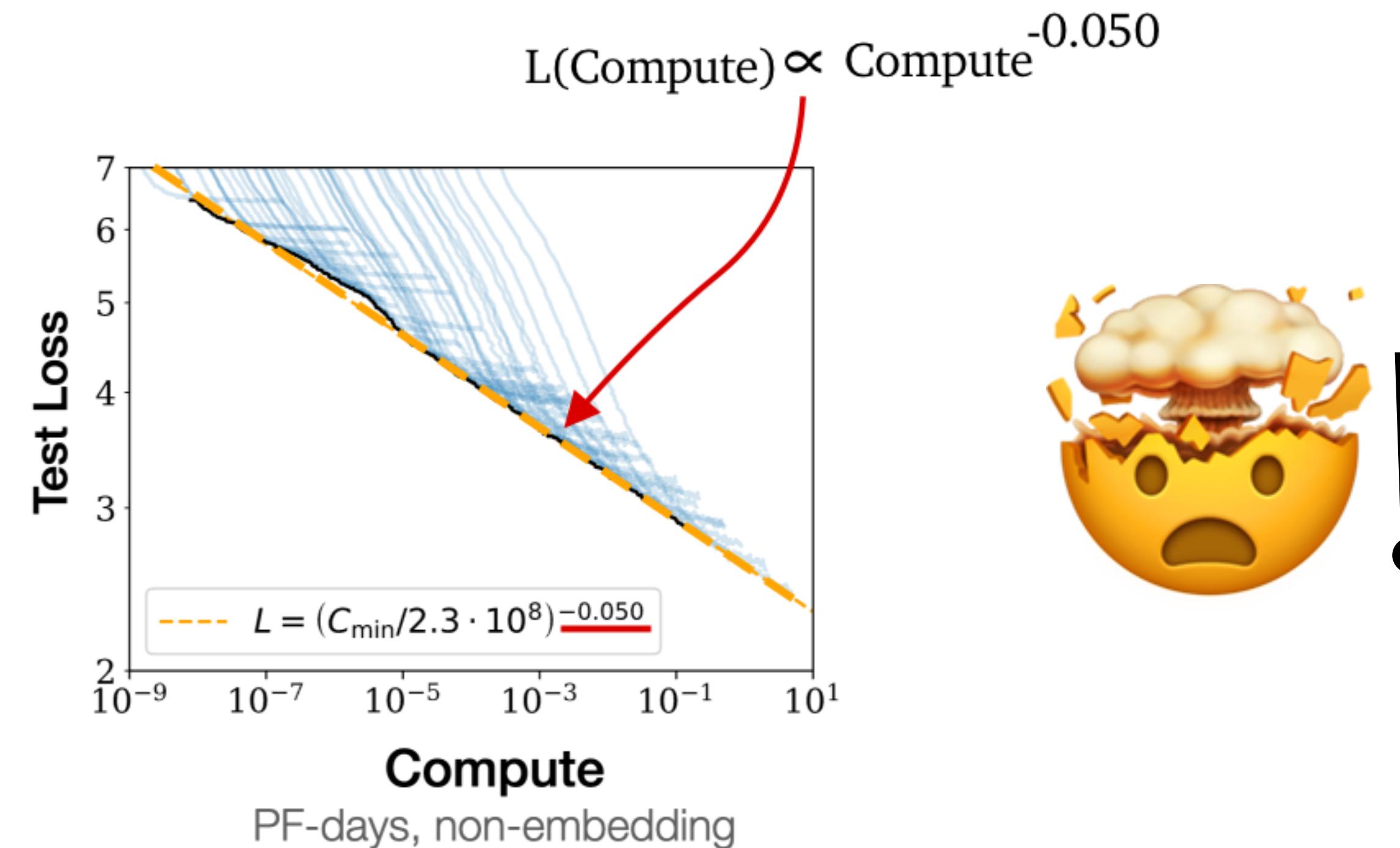
Test-time Compute

- “Performance has a power-law relationship with each of the three scale factors”



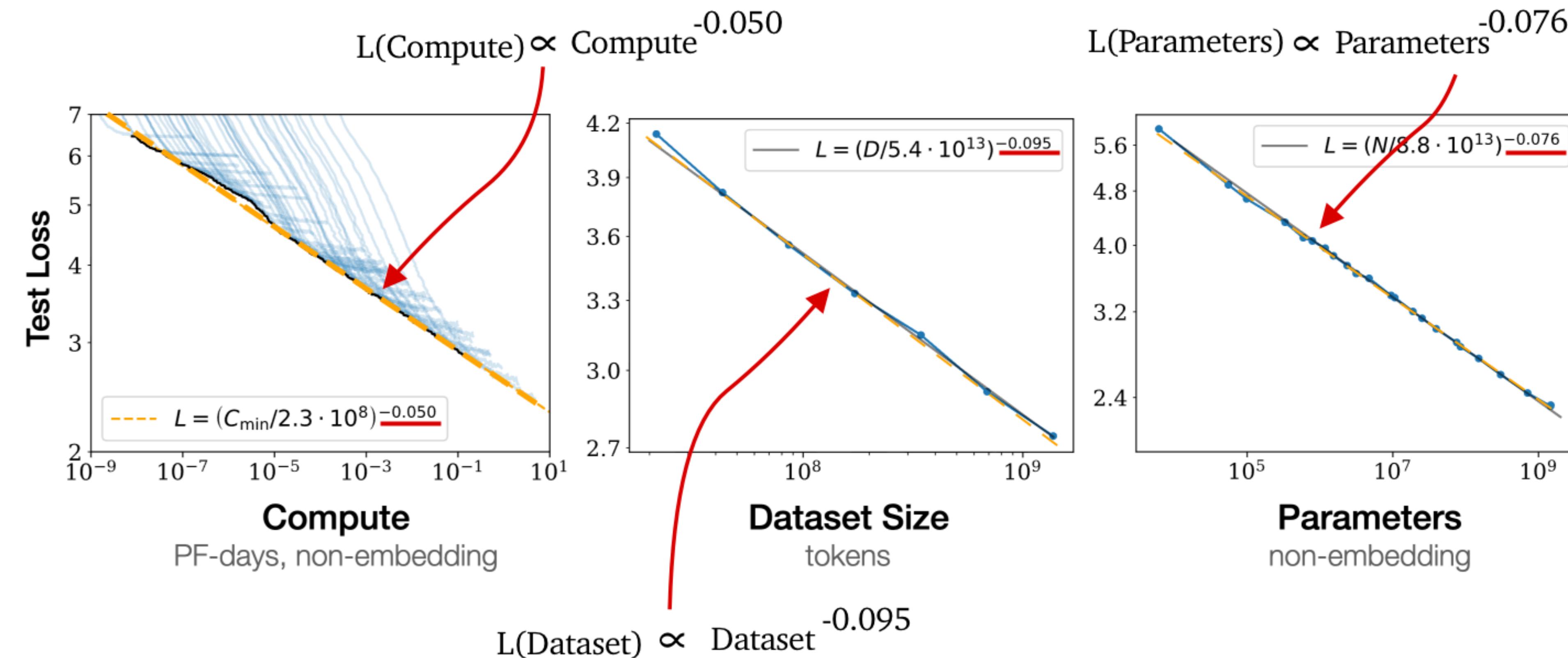
Test-time Compute

- “Performance has a power-law relationship with each of the three scale factors”



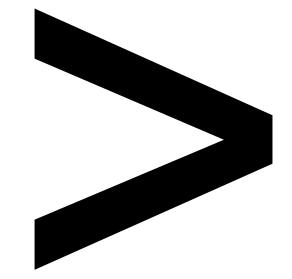
Test-time Compute

- “Performance has a power-law relationship with each of the three scale factors”



Test-time Compute

Computation
at Scale



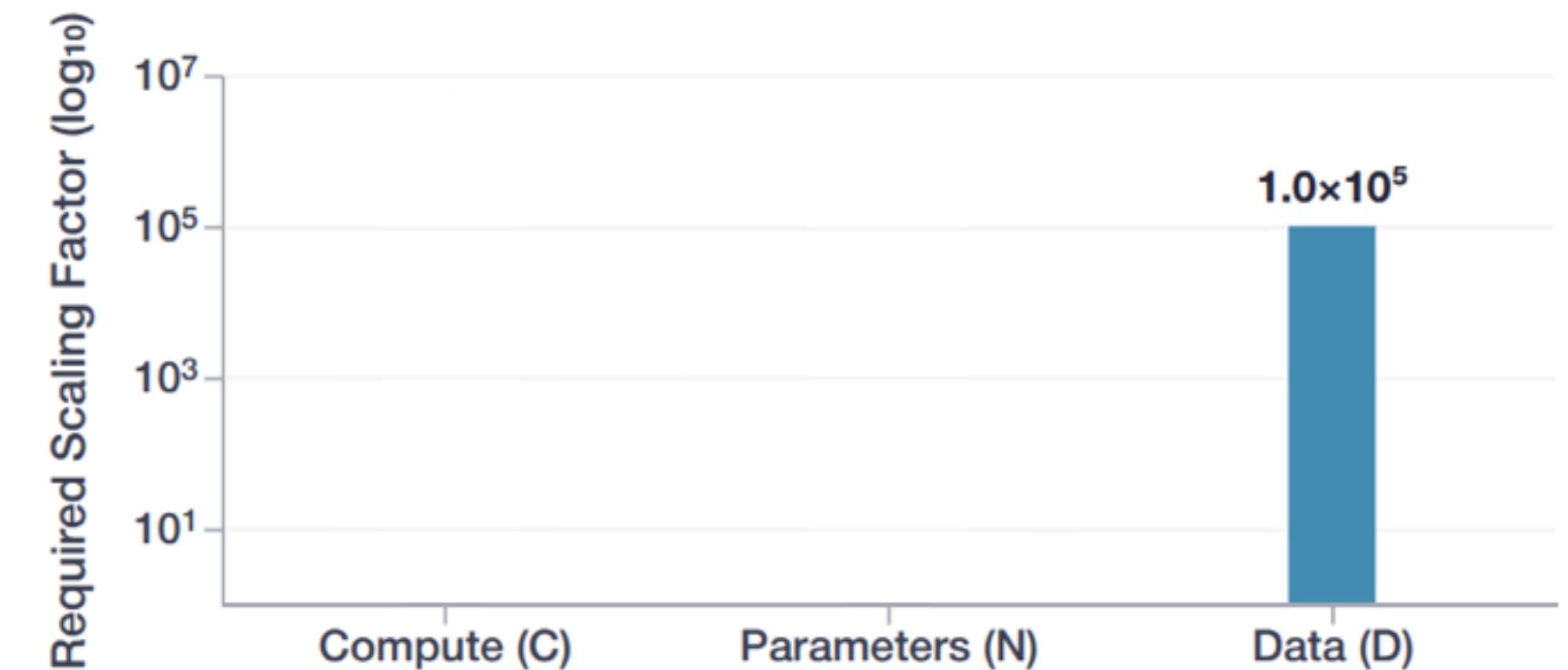
Algorithmic *
Breakthrough

Test-time Compute (~3 years later)

- But... napkin math

Scaling Requirements for 2x Performance Improvement

Factor increase required to halve test loss under power law scaling



Scaling Laws:

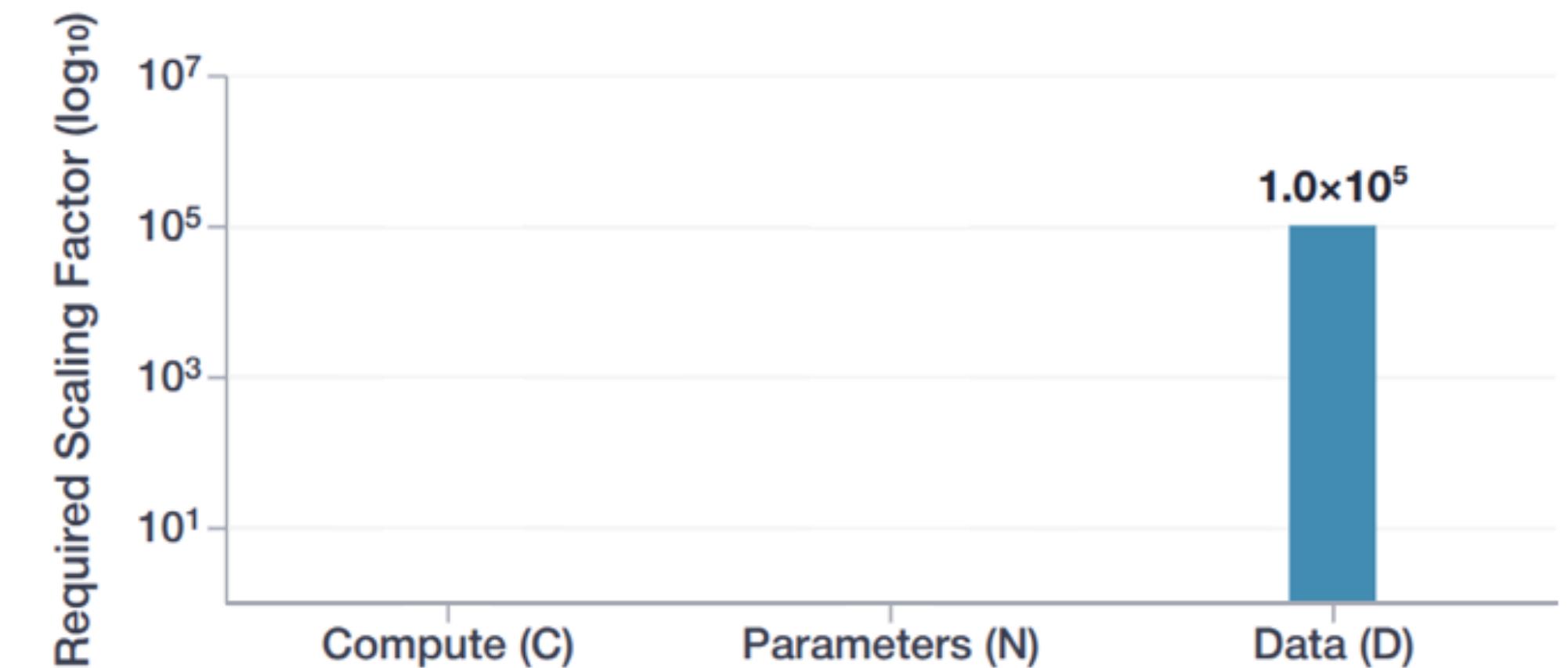
$$L(D) = (D/D_0)^{-0.095}$$

Test-time Compute (~3 years later)

- But... napkin math
- Useful tokens from internet $\sim 10T$ tokens
- GPT-4 and Llama 3.1 $\sim 15T$ tokens
- Increase that by $10^5 = 1.5$ Quintillion

Scaling Requirements for 2x Performance Improvement

Factor increase required to halve test loss under power law scaling



Scaling Laws:

$$L(D) = (D/D_0)^{-0.095}$$

Test-time Compute (~3 years later)

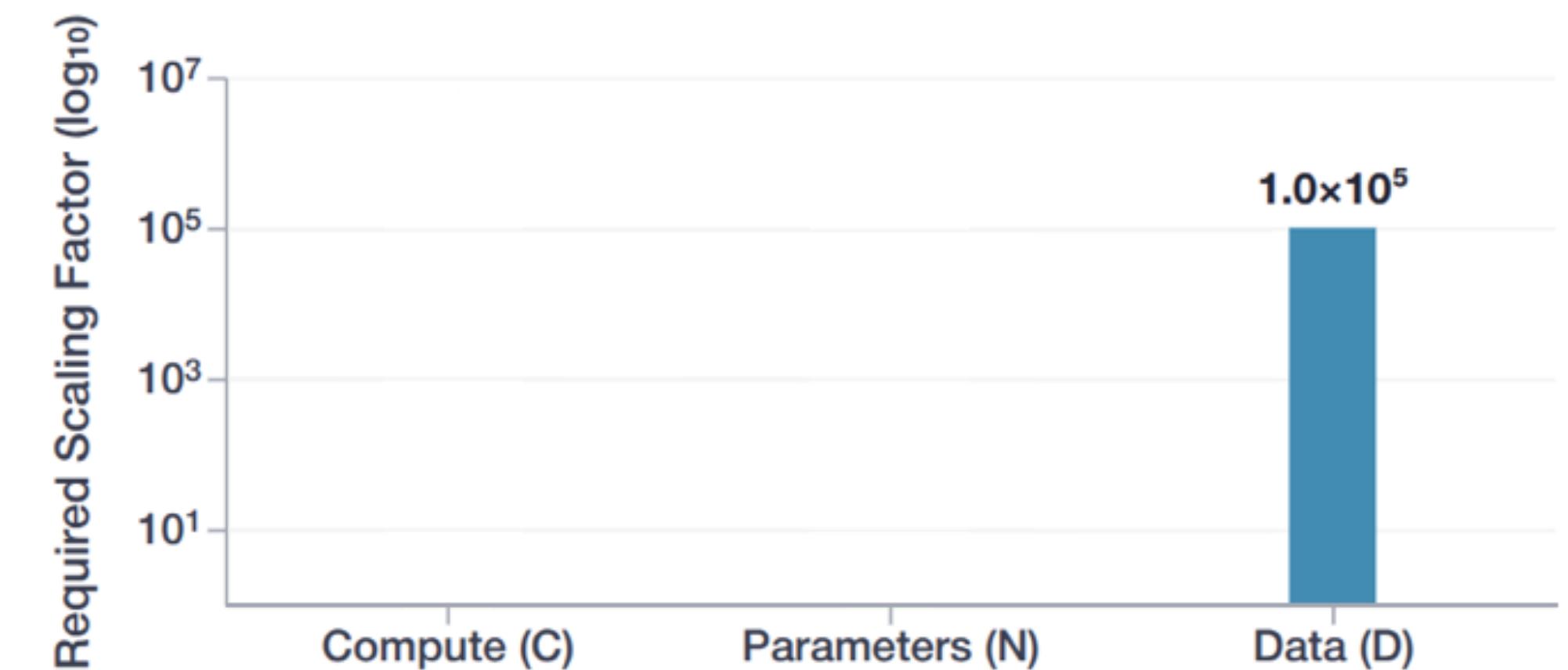
- But... napkin math
- Useful tokens from internet $\sim 10T$ tokens
- GPT-4 and Llama 3.1 $\sim 15T$ tokens
- Increase that by $10^5 = 1.5$ Quintillion

“the data is not growing,
we have but one internet”

“we have achieved peak data,
there will be no more”

Scaling Requirements for 2x Performance Improvement

Factor increase required to halve test loss under power law scaling



Scaling Laws:

$$L(D) = (D/D_0)^{-0.095}$$

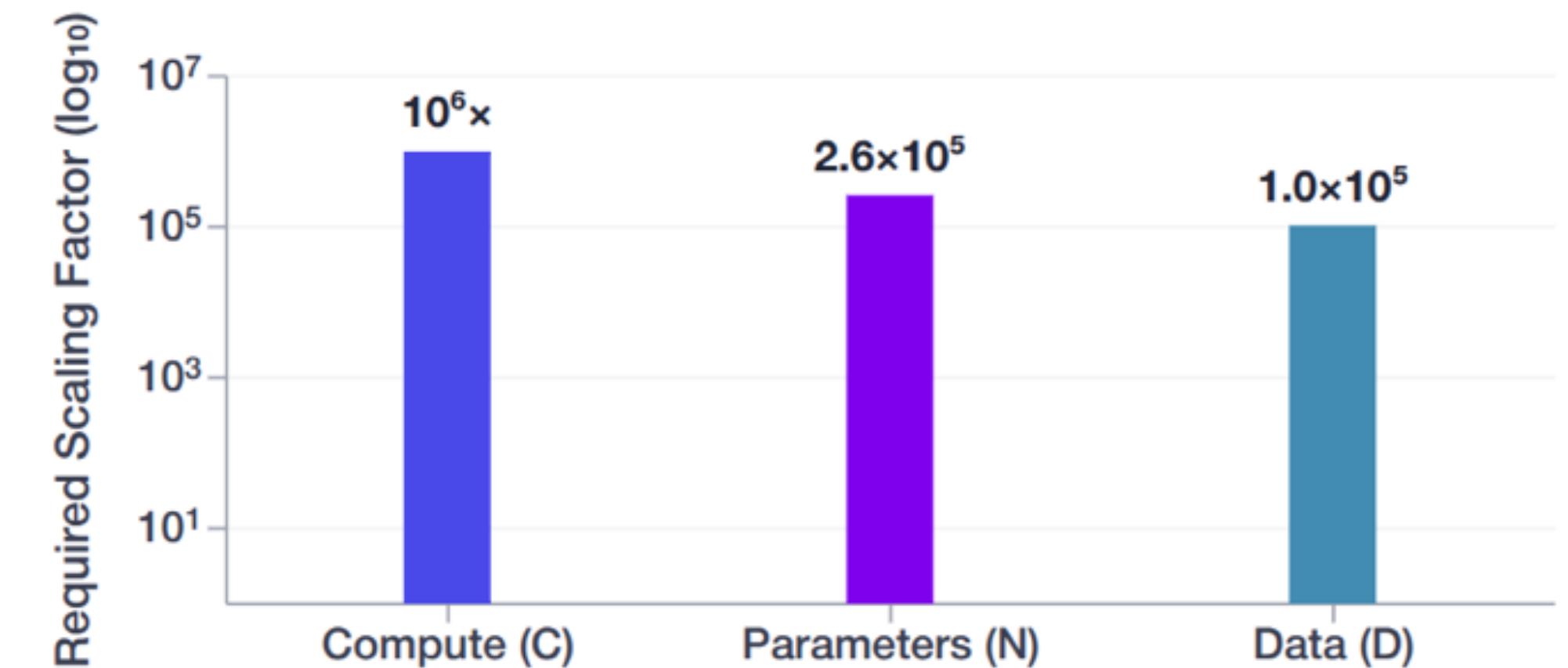
Test-time Compute (~3 years later)

- But... napkin math
- Scaling other dimensions, diminishing returns

“Intelligence scales with the log of compute”

Scaling Requirements for 2x Performance Improvement

Factor increase required to halve test loss under power law scaling



Scaling Laws:

$$L(C) = (C/C_0)^{-0.050}$$

$$L(N) = (N/N_0)^{-0.076}$$

$$L(D) = (D/D_0)^{-0.095}$$

Note: Achieving 2x improvement requires ~1,000,000x more compute, ~260,000x more parameters, or ~100,000x more training data when scaling each dimension independently. Based on Kaplan et al. (2020).

Ilya Sutskever: "Sequence to sequence learning with neural networks: what a decade", NeurIPS 2024
<https://www.youtube.com/watch?v=1yvBqasHLZs>

Test-time Compute (~3 years later)

- But... napkin math
- Scaling other dimensions, diminishing returns

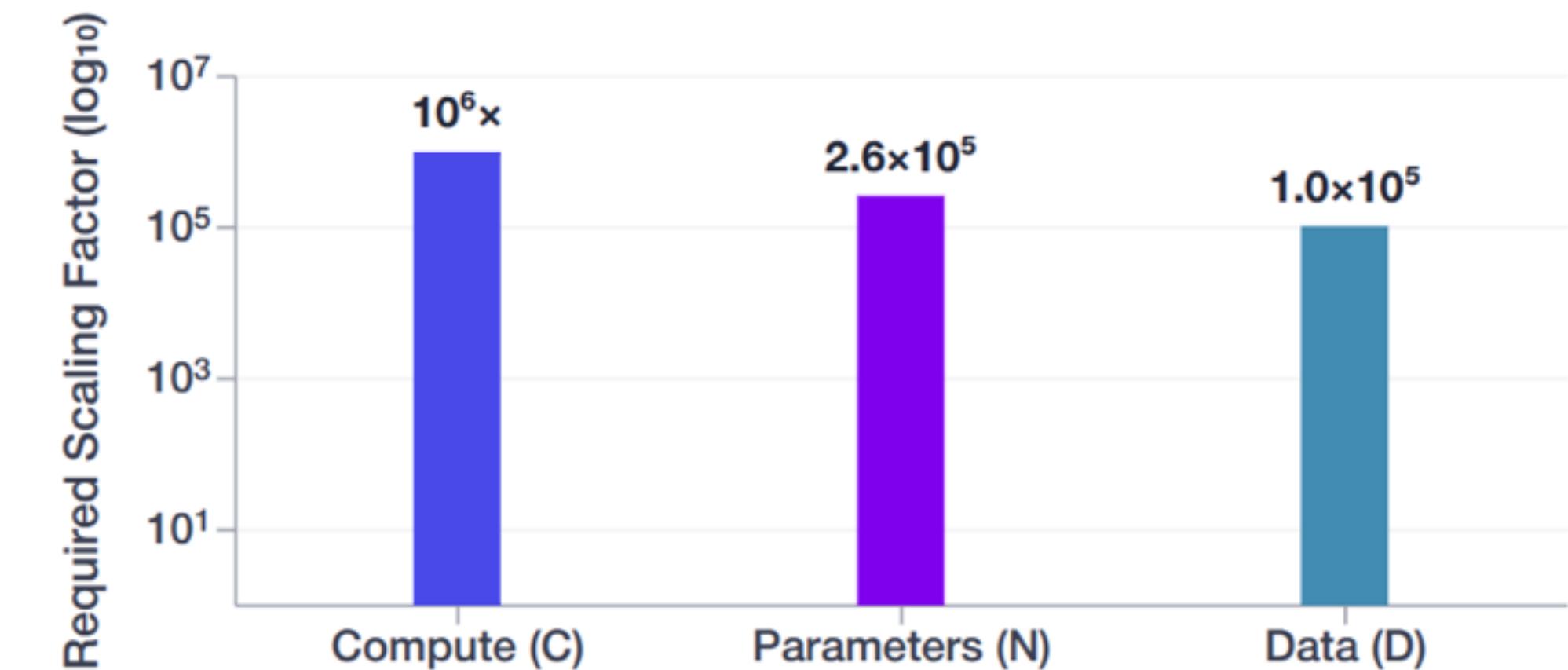
“Intelligence scales with the log of compute”

“The age of pre-training scaling has ended”

“Deep learning has hit a wall”

Scaling Requirements for 2x Performance Improvement

Factor increase required to halve test loss under power law scaling



Scaling Laws:

$$L(C) = (C/C_0)^{-0.050}$$

$$L(N) = (N/N_0)^{-0.076}$$

$$L(D) = (D/D_0)^{-0.095}$$

Note: Achieving 2x improvement requires ~1,000,000x more compute, ~260,000x more parameters, or ~100,000x more training data when scaling each dimension independently. Based on Kaplan et al. (2020).

Test-time Compute

- Until... RL to the rescue

Test-time Compute

- OpenAI o1, a new kind of model: “*Spend more time thinking before they respond*”
- “*Can reason through complex tasks and solve harder problems than previous models*”

The screenshot shows a white web page with a light gray header bar. In the top left corner of the bar, the date "September 12, 2024" is displayed next to the word "Release". Below the header, the main title "Learning to reason with LLMs" is centered in a large, bold, black font. Underneath the title, there are two small, dark gray buttons: "Contributions" on the left and "Use o1" with a right-pointing arrow on the right. The main content area contains a paragraph of text about the model's capabilities and its performance across various fields like competitive programming and academic competitions. At the bottom of this section, there is another paragraph describing the reinforcement learning process used to train the model. The entire page has a clean, minimalist design with a white background.

September 12, 2024 Release

Learning to reason with LLMs

Contributions Use o1 ↗

OpenAI o1 ranks in the 89th percentile on competitive programming questions (Codeforces), places among the top 500 students in the US in a qualifier for the USA Math Olympiad (AIME), and exceeds human PhD-level accuracy on a benchmark of physics, biology, and chemistry problems (GPQA). While the work needed to make this new model as easy to use as current models is still ongoing, we are releasing an early version of this model, OpenAI o1-preview, for immediate use in ChatGPT and to [trusted API users](#).

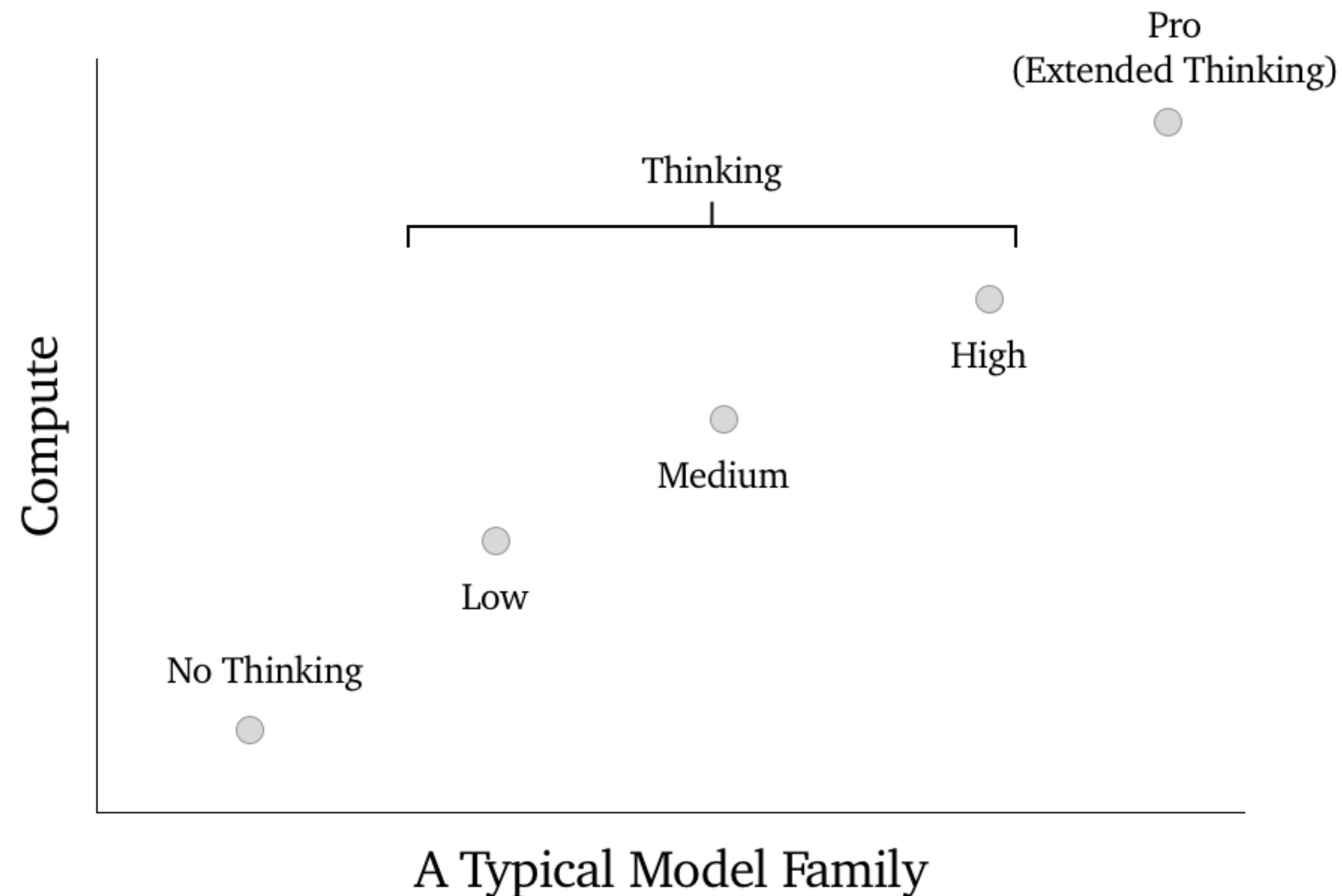
Our large-scale reinforcement learning algorithm teaches the model how to think productively using its chain of thought in a highly data-efficient training process. We

Test-time Compute

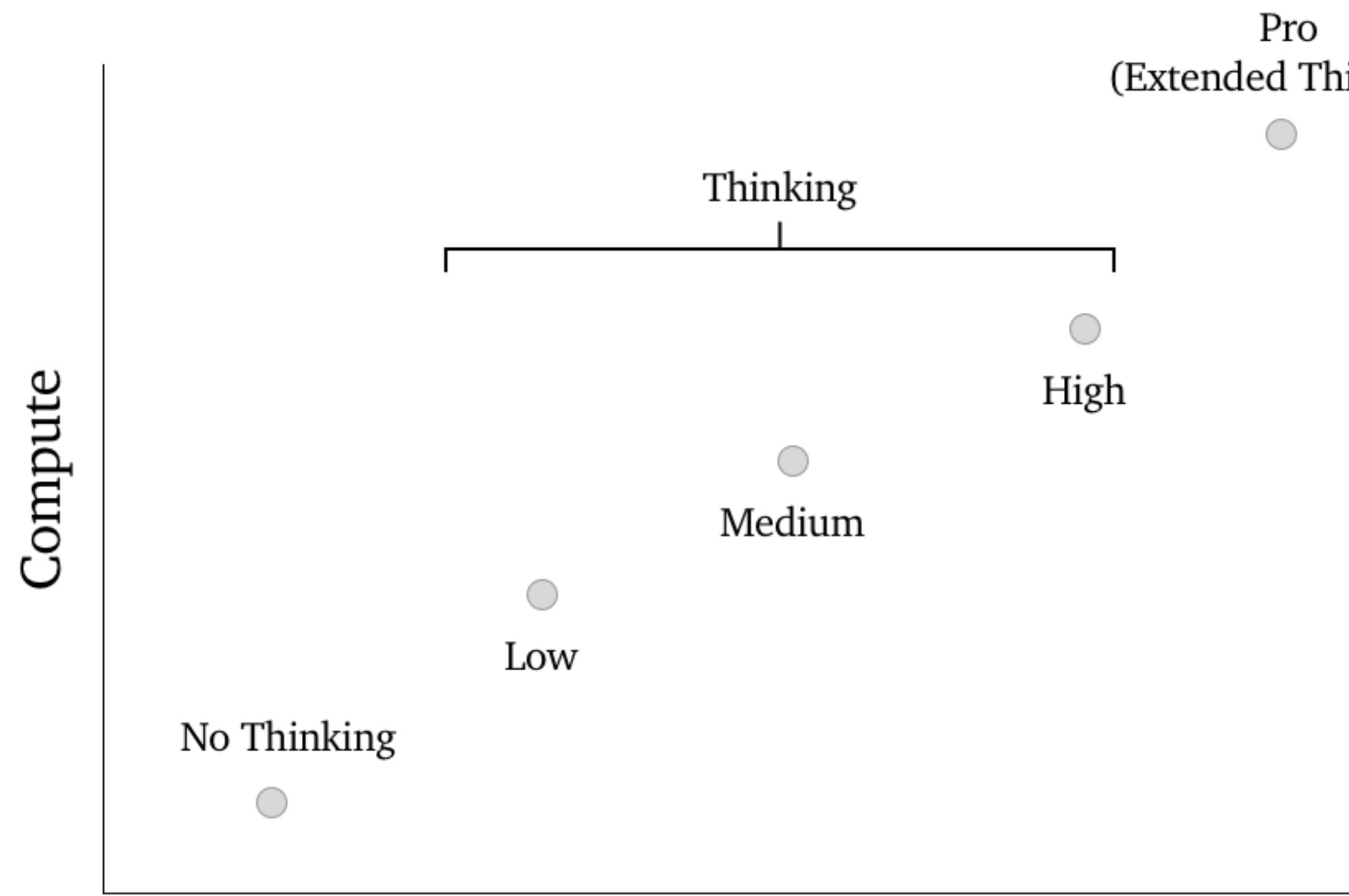
- OpenAI o1, a new kind of model: “*Spend more time thinking before they respond*”
- “*Can reason through complex tasks and solve harder problems than previous models*”
- A new scaling law:

“Performance scales with the amount of compute applied during generation of an answer”

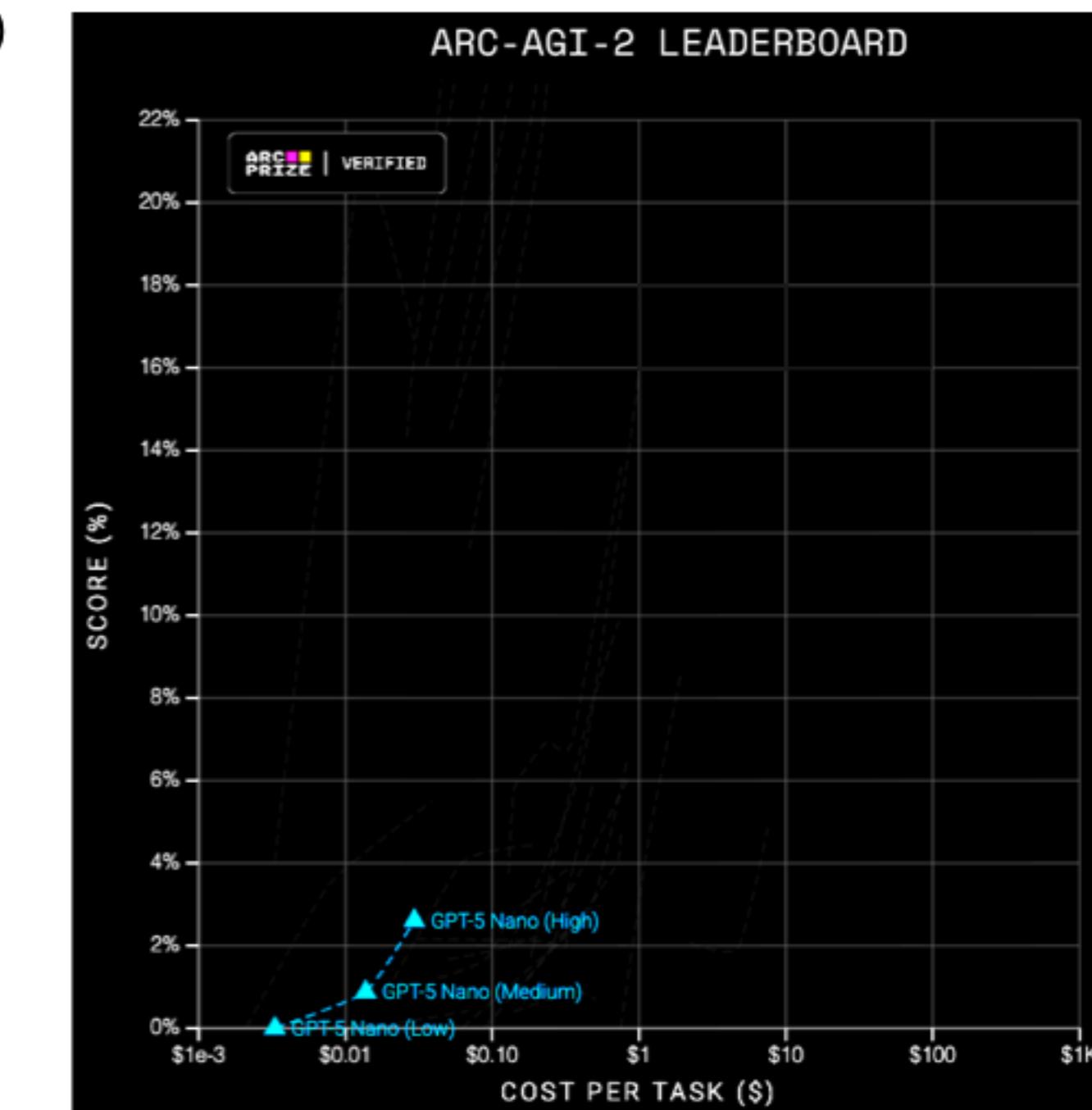
Test-time Compute



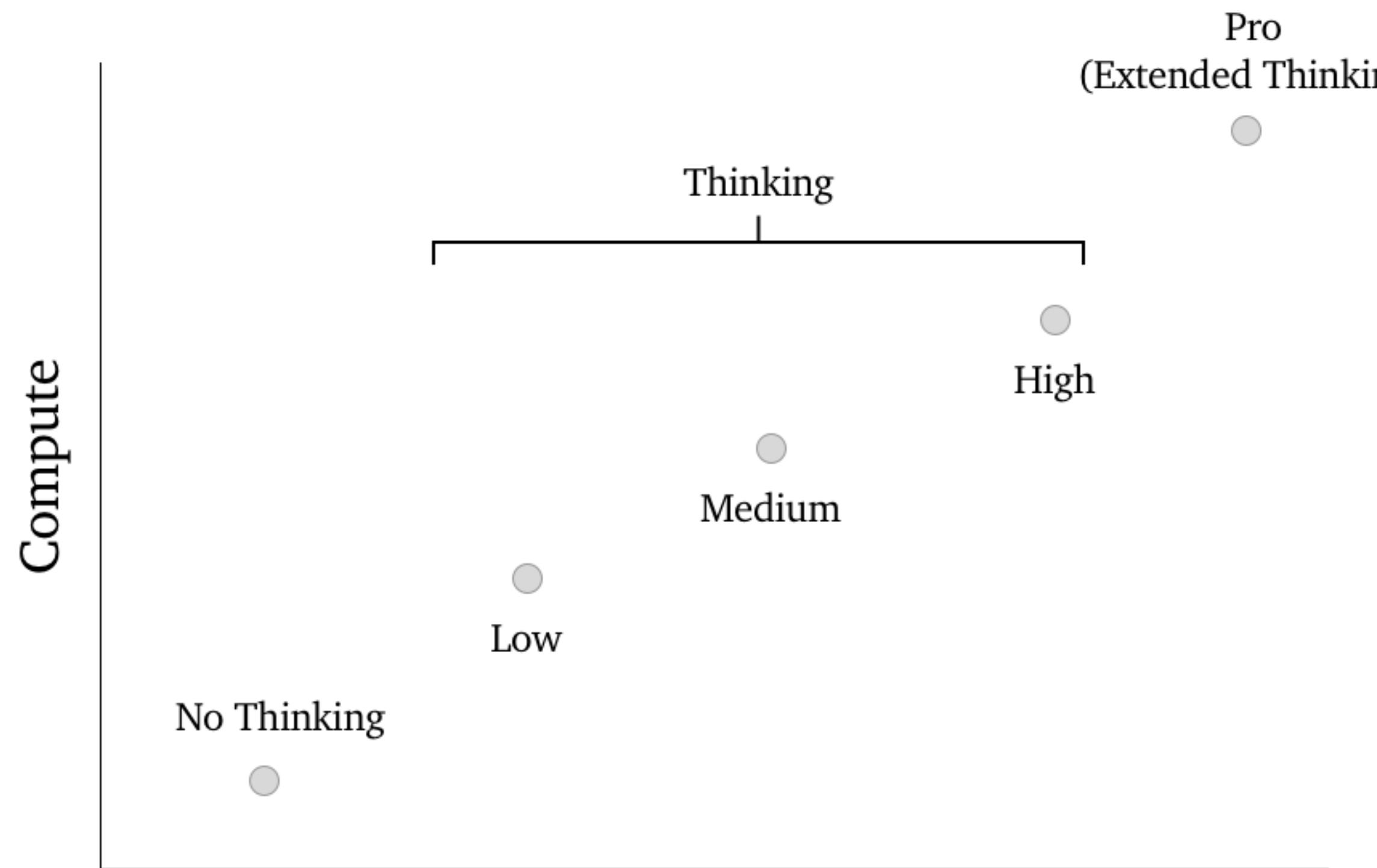
Test-time Compute



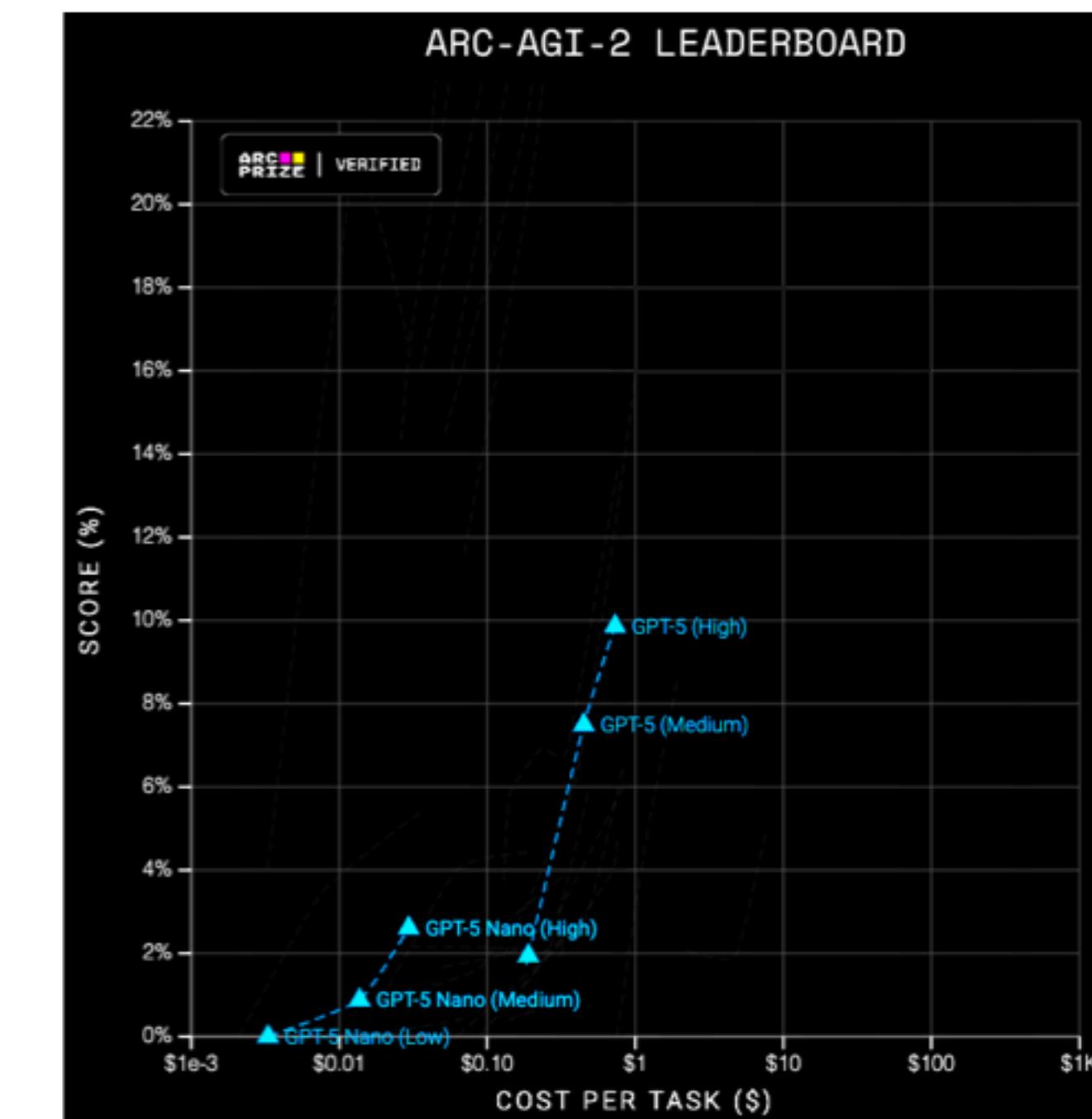
A Typical Model Family



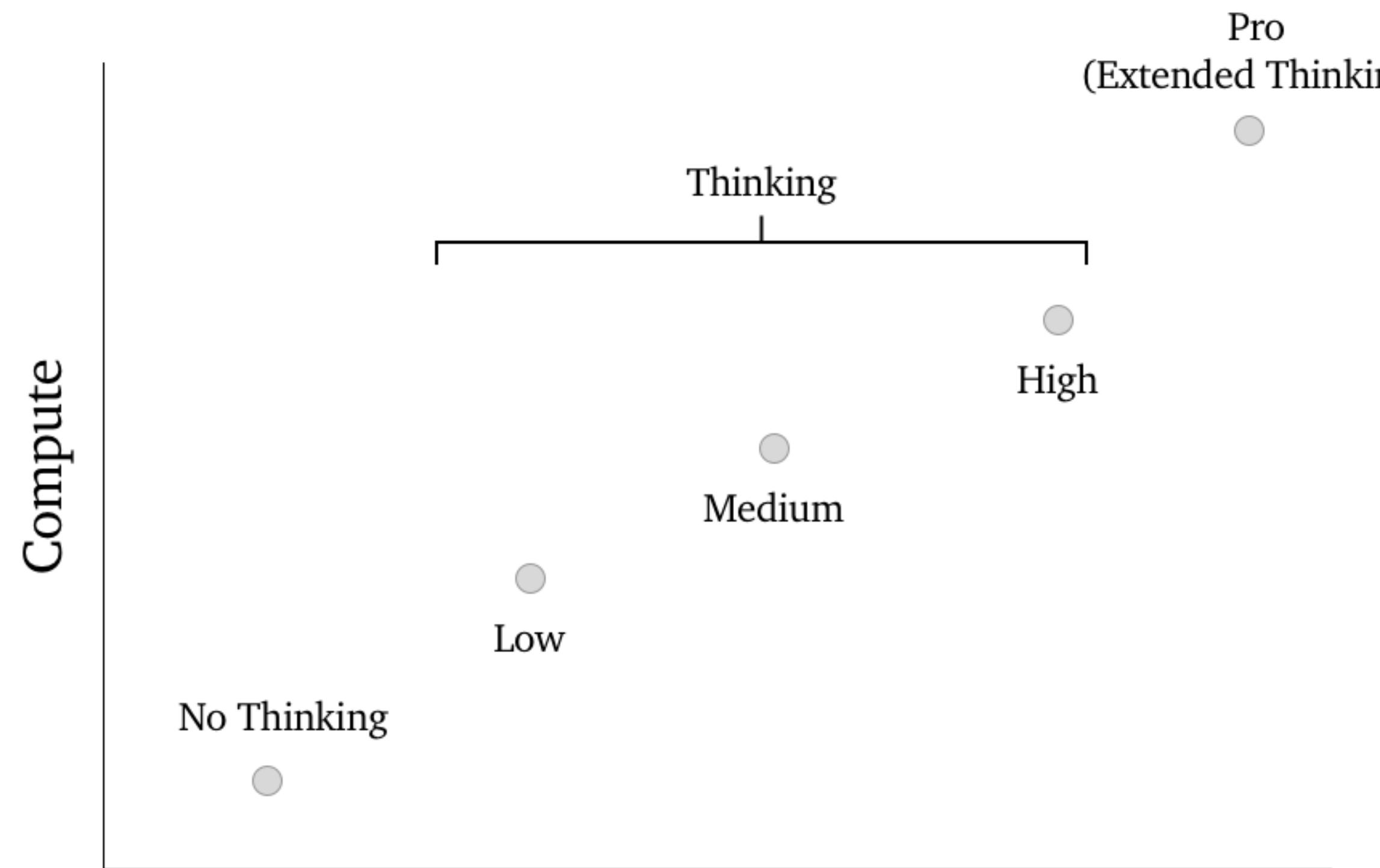
Test-time Compute



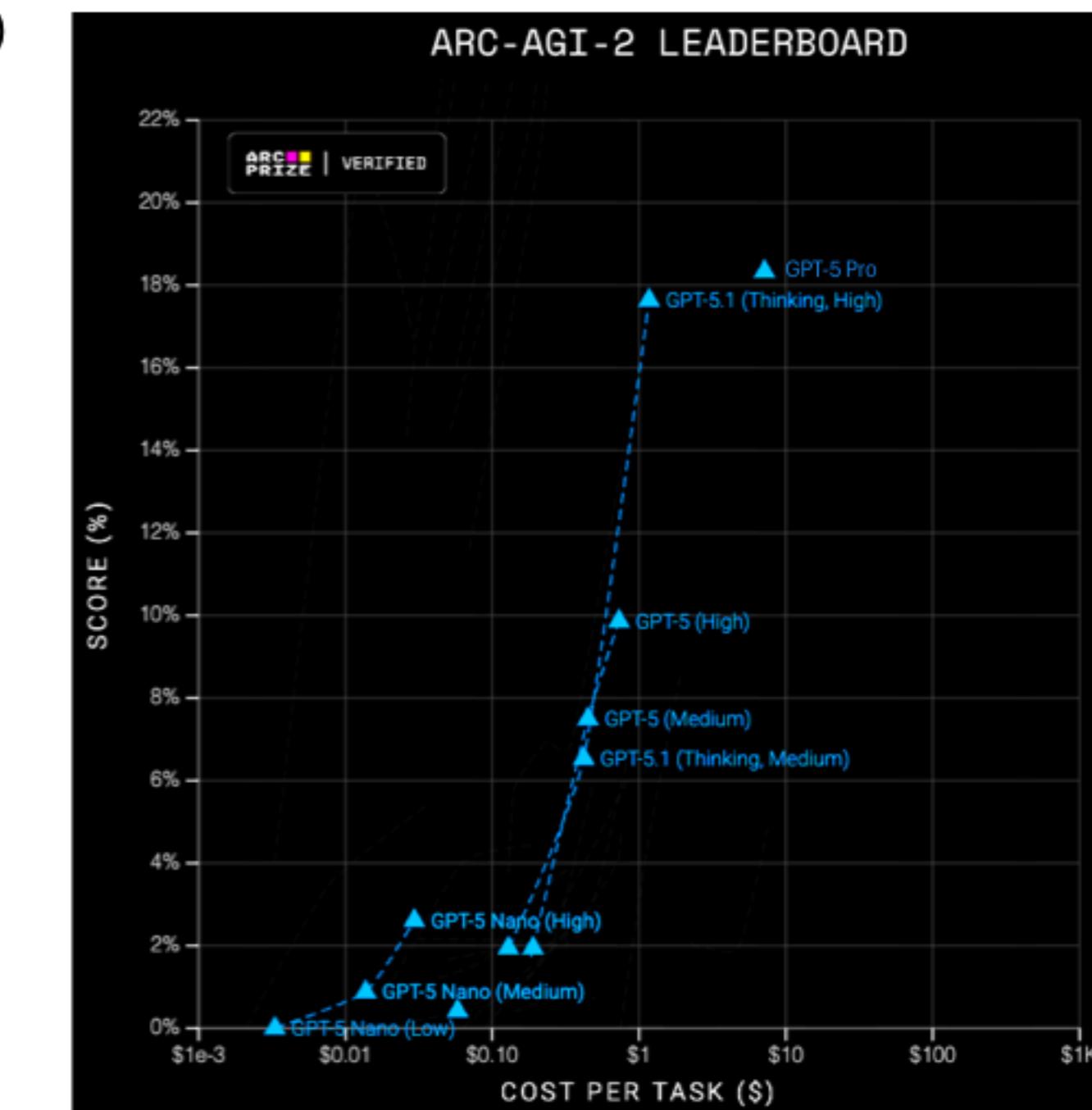
A Typical Model Family



Test-time Compute



A Typical Model Family

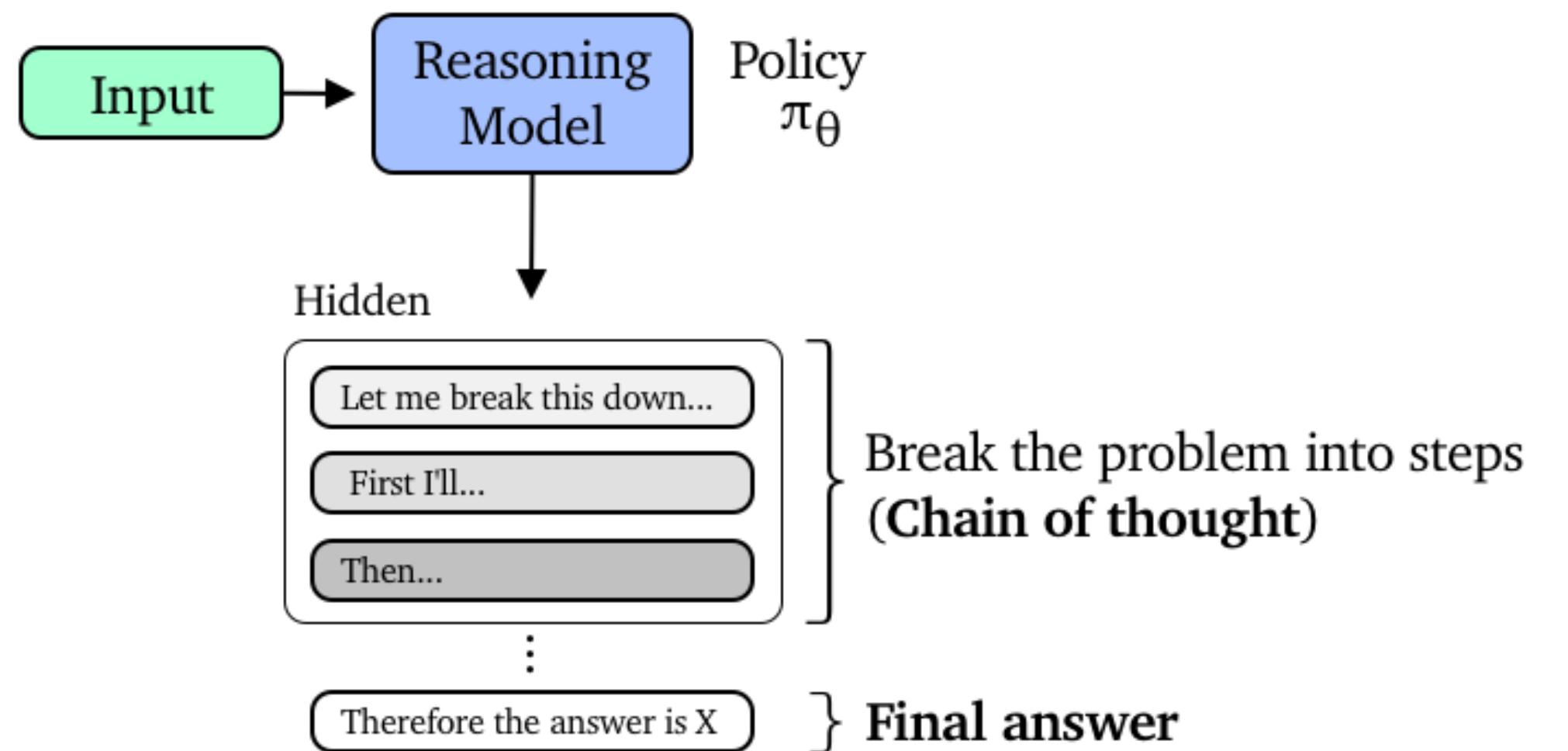


Test-time Compute

- But where is the RL?

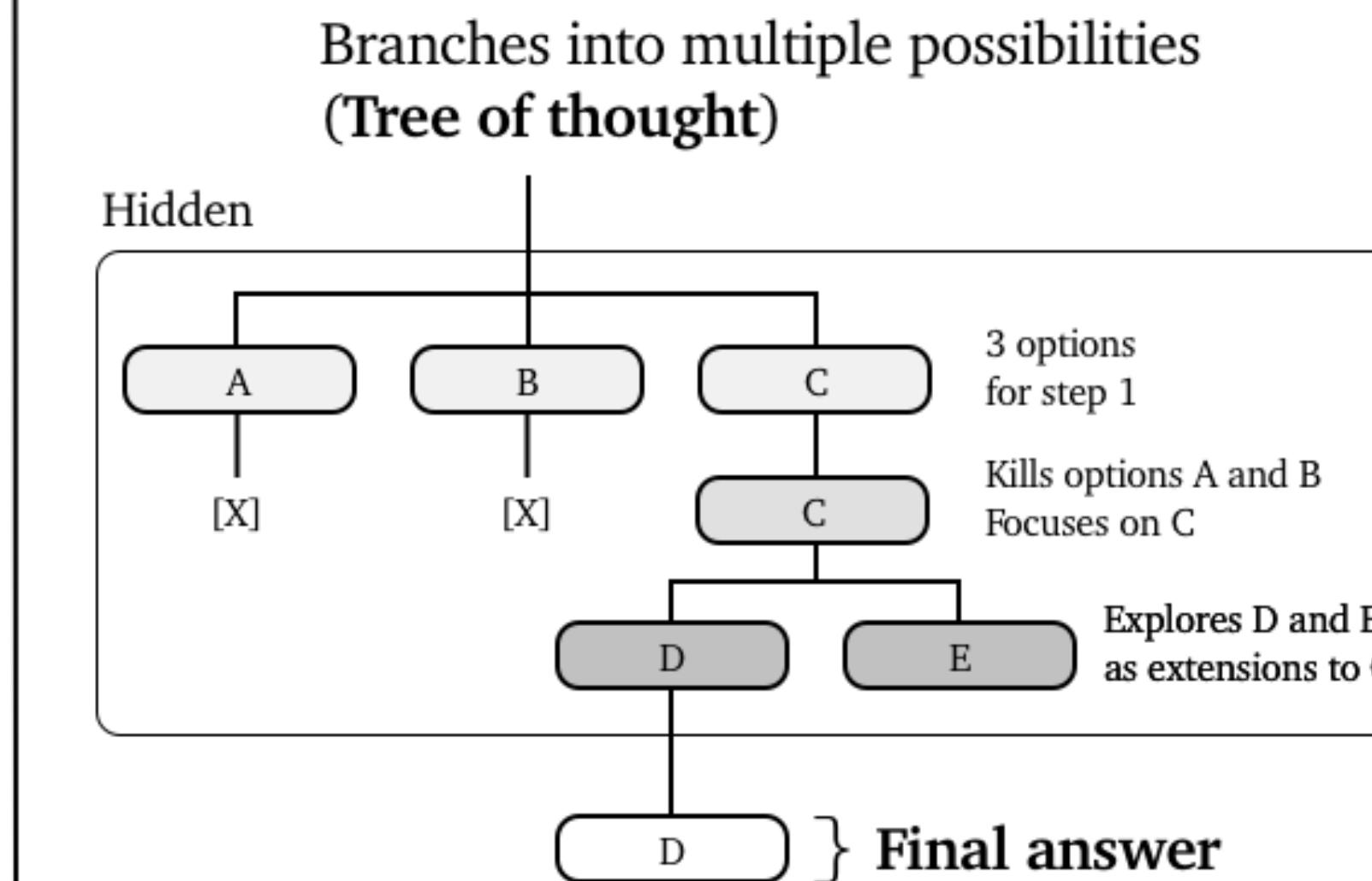
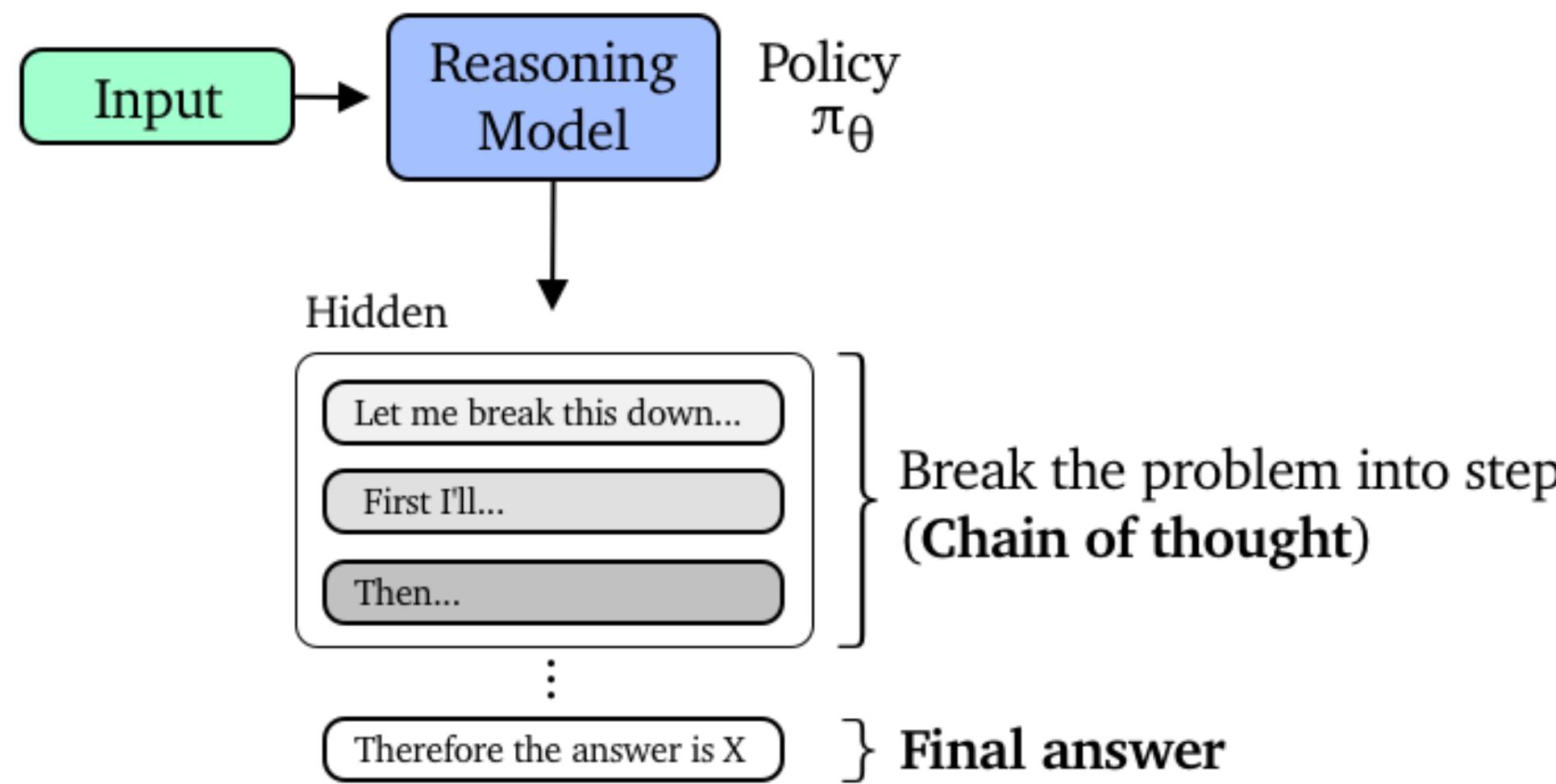
Test-time Compute

- But where is the RL?



Test-time Compute

- But where is the RL?

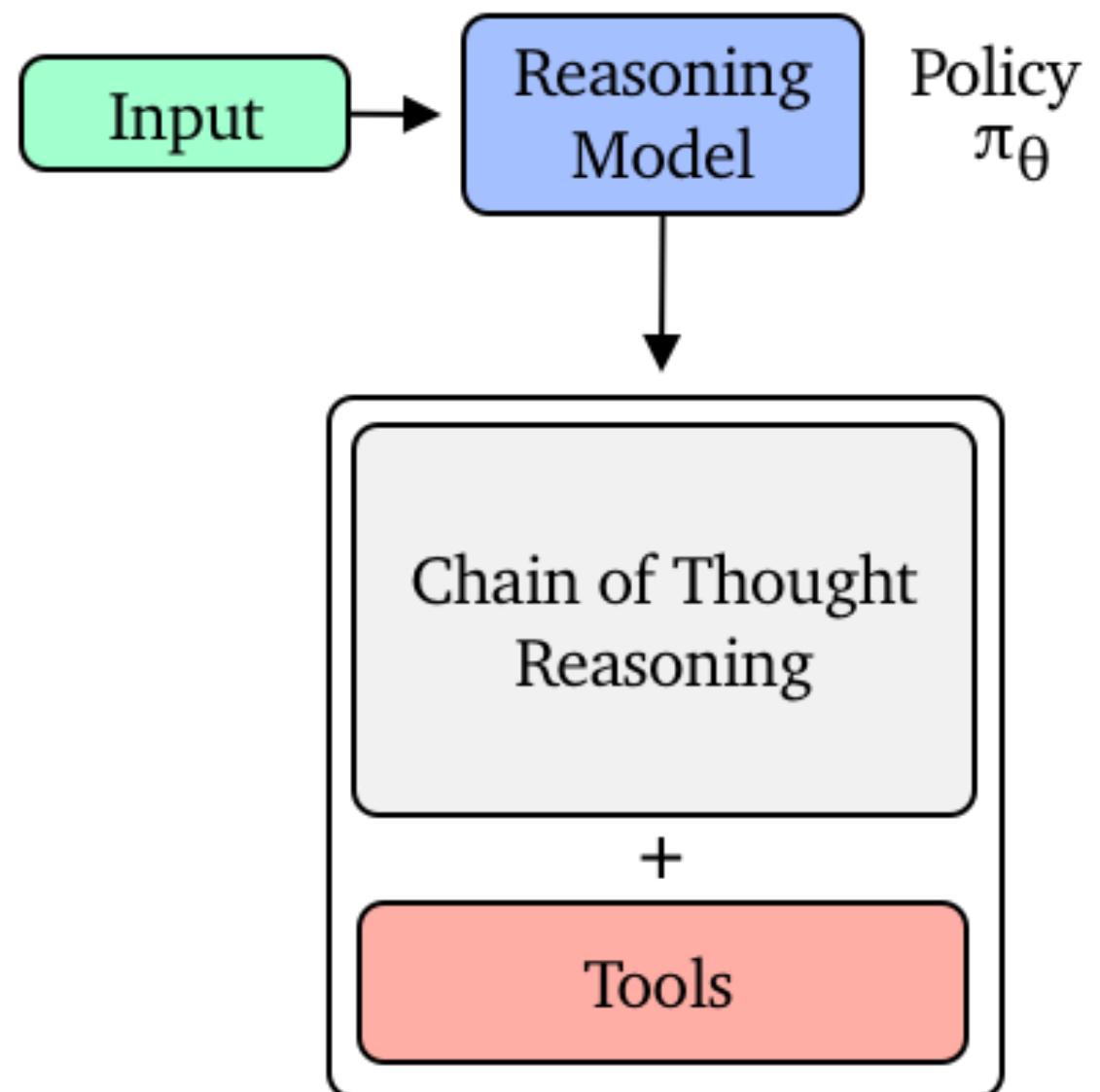


More complex traces
(Graph of thought)

- Self-correction
- Merge
- Backtrack

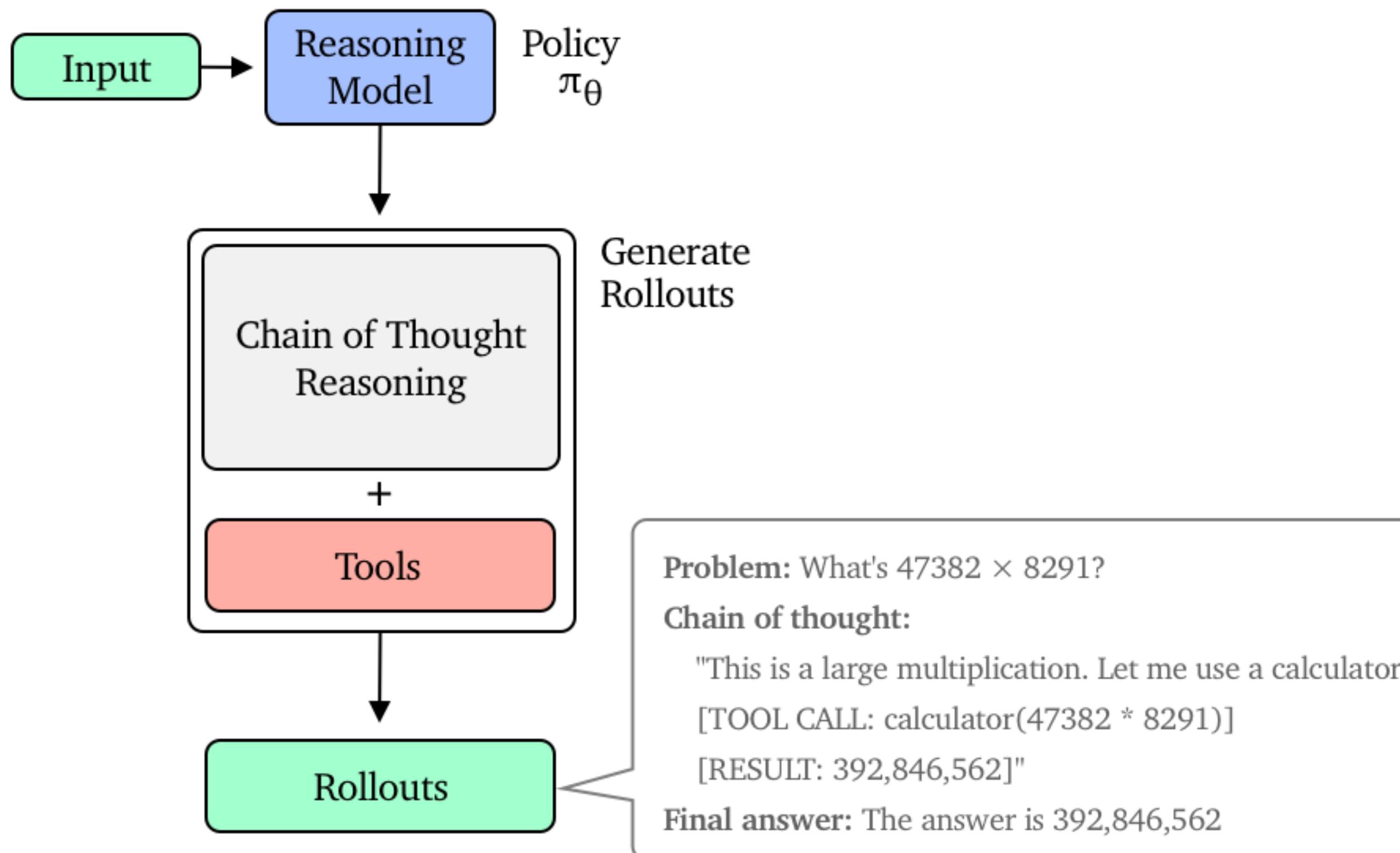
Test-time Compute

- But where is the RL?



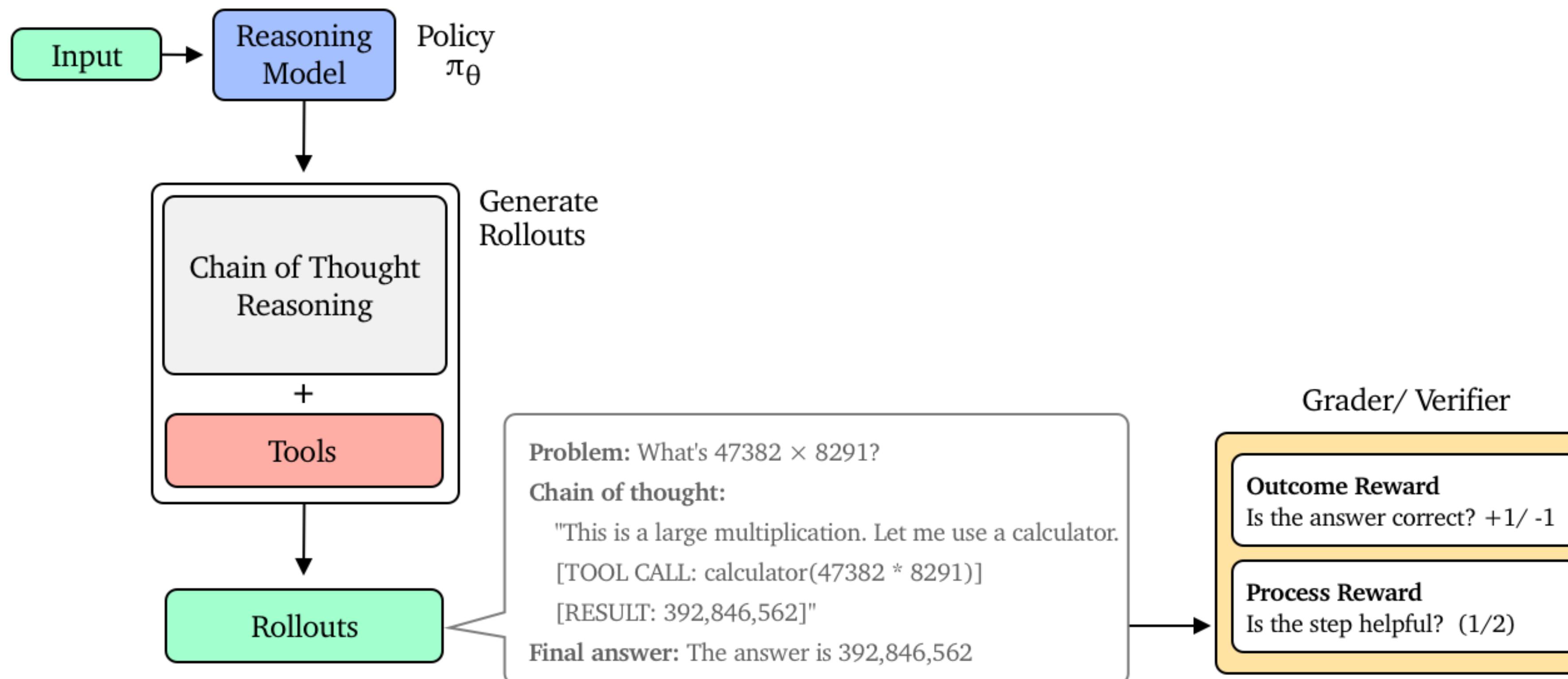
Test-time Compute

- But where is the RL?



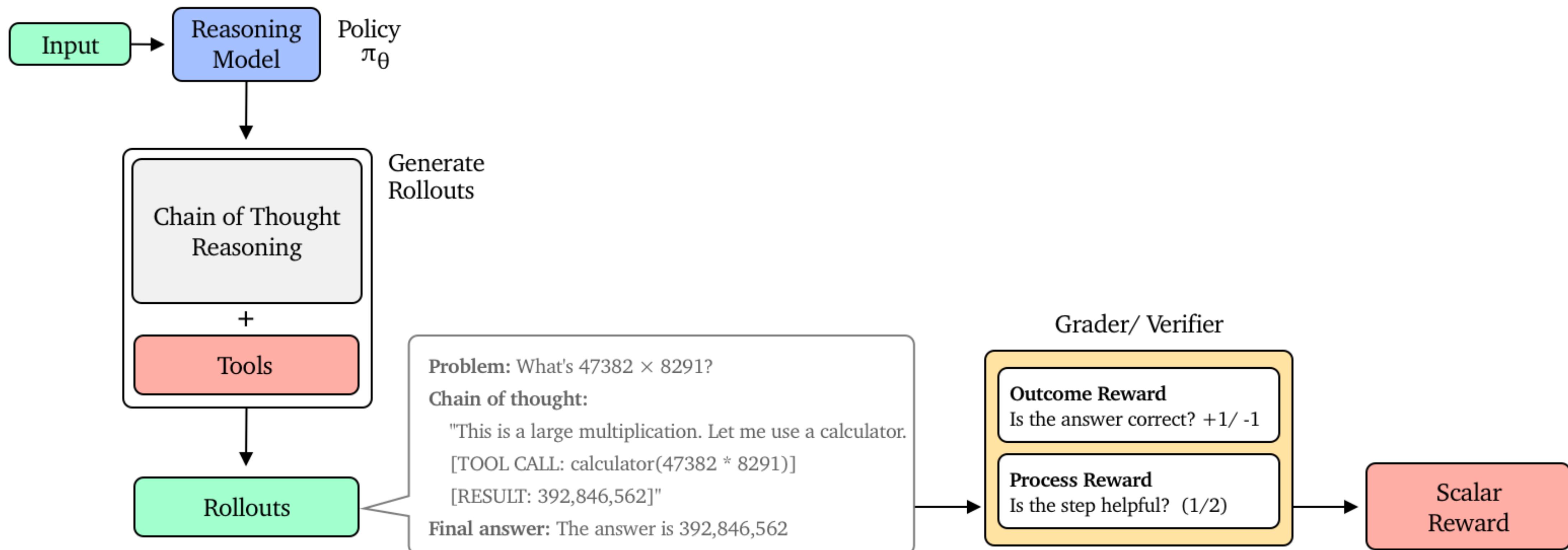
Test-time Compute

- But where is the RL?



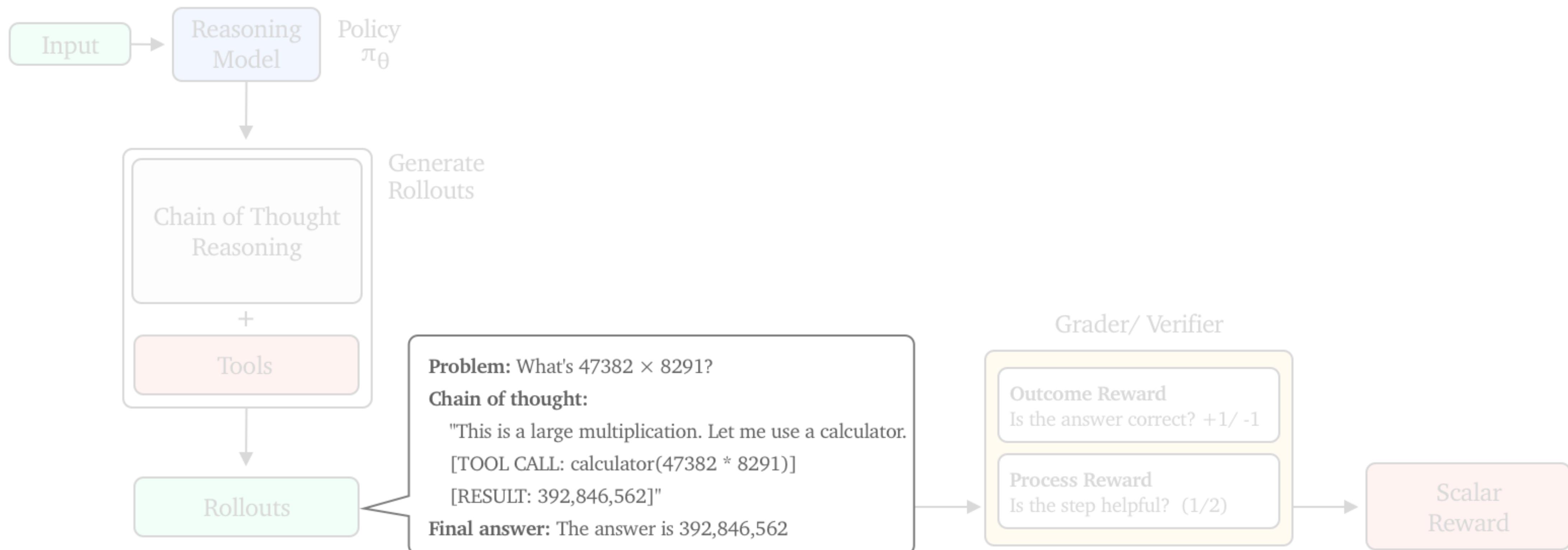
Test-time Compute

- But where is the RL?



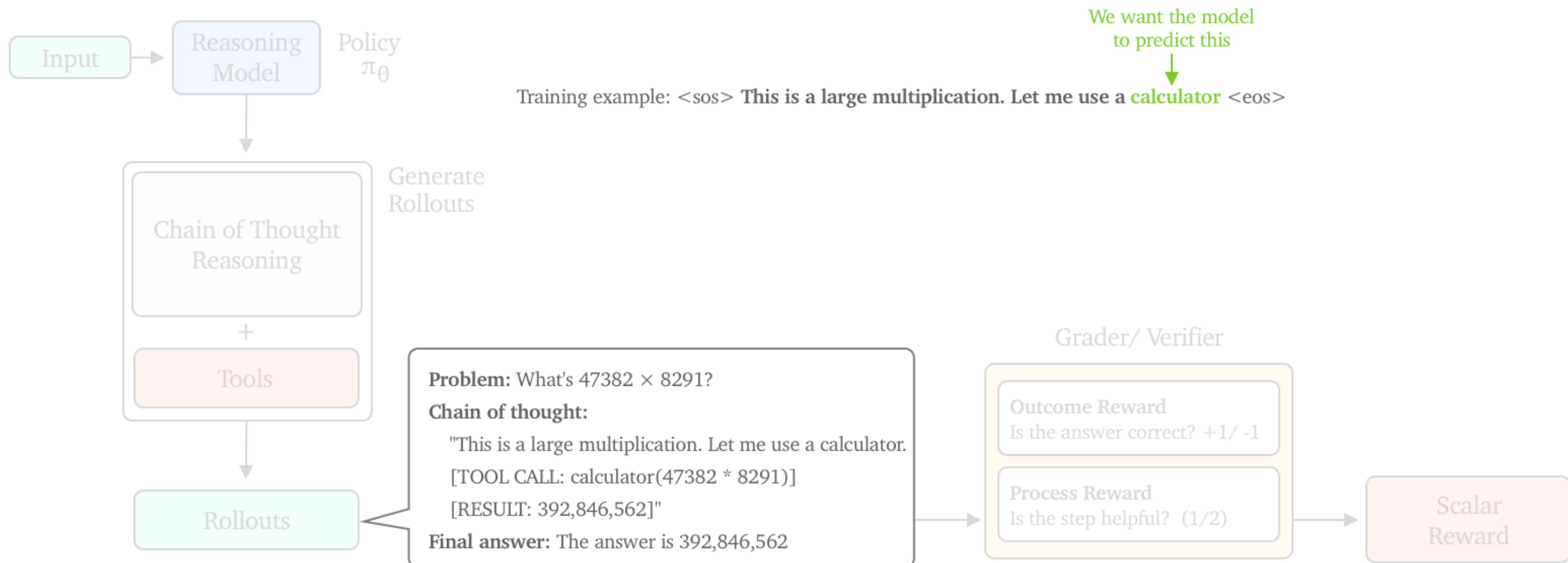
Test-time Compute

- But where is the RL?



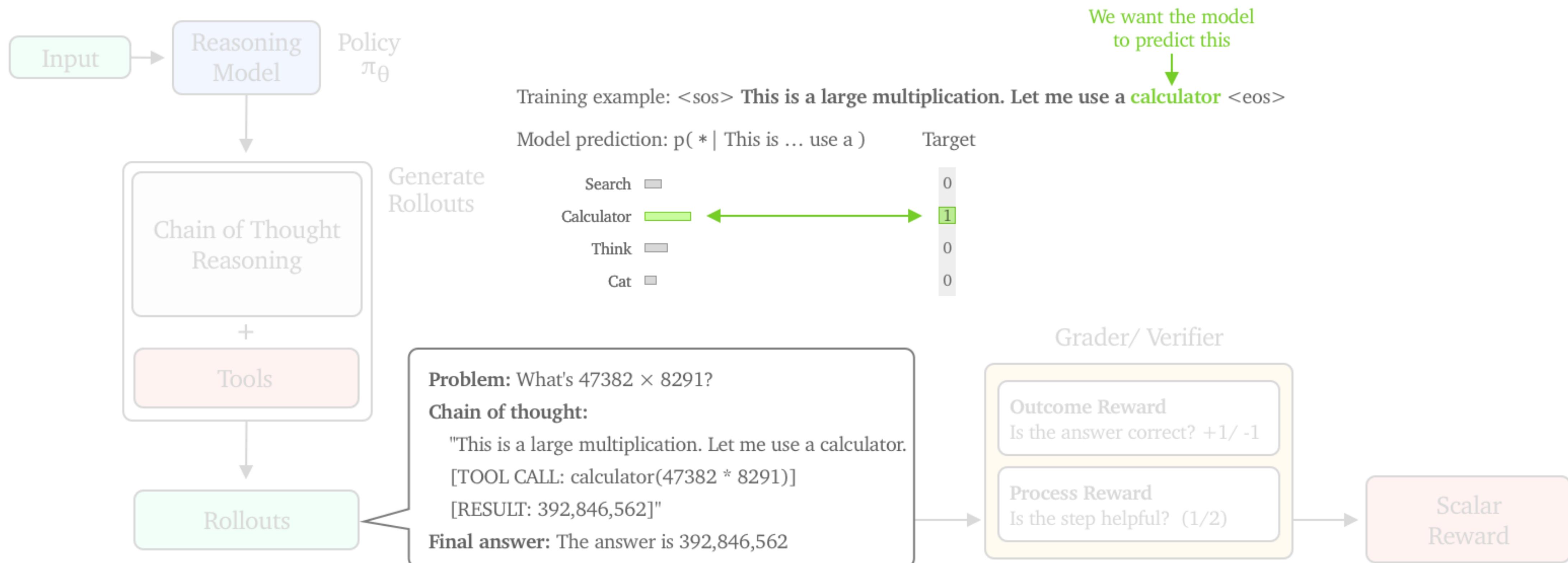
Test-time Compute

- But where is the RL?



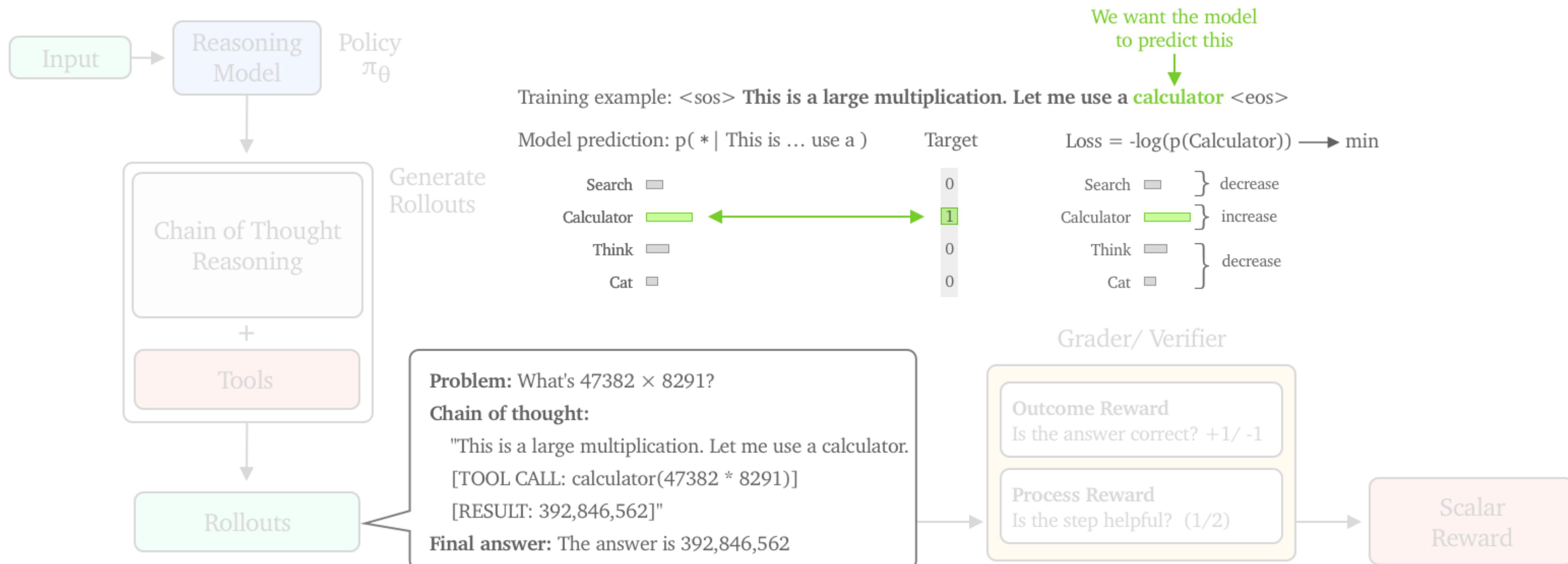
Test-time Compute

- But where is the RL?



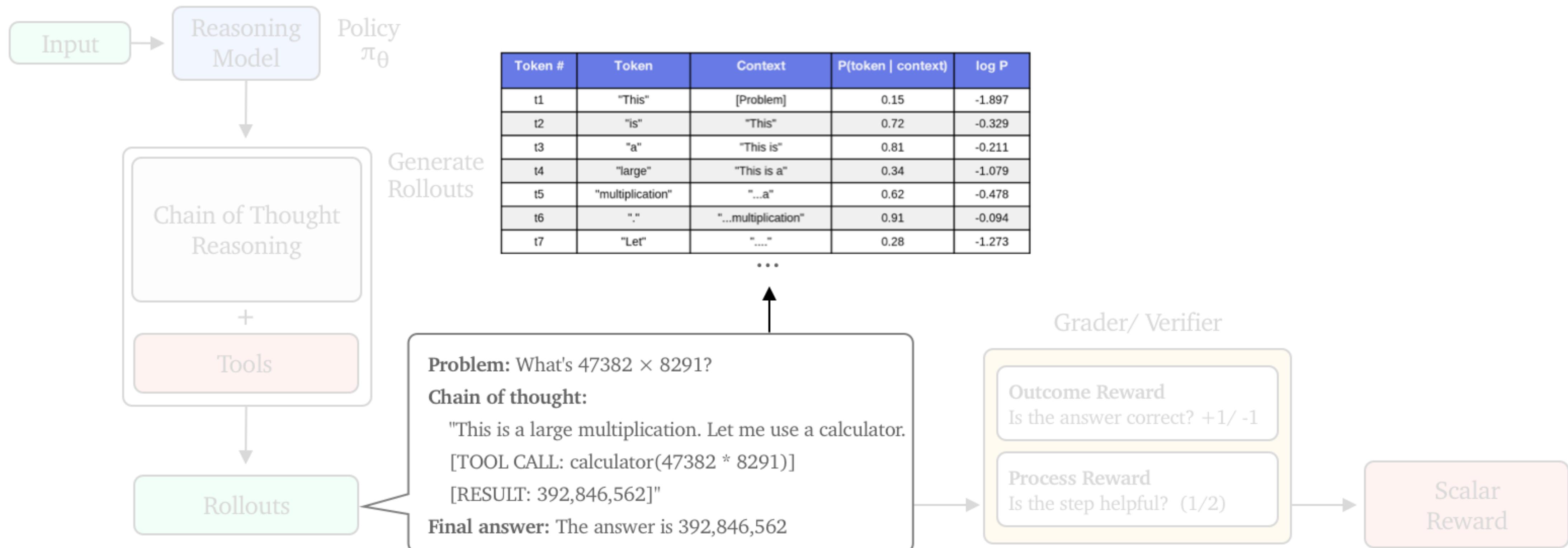
Test-time Compute

- But where is the RL?



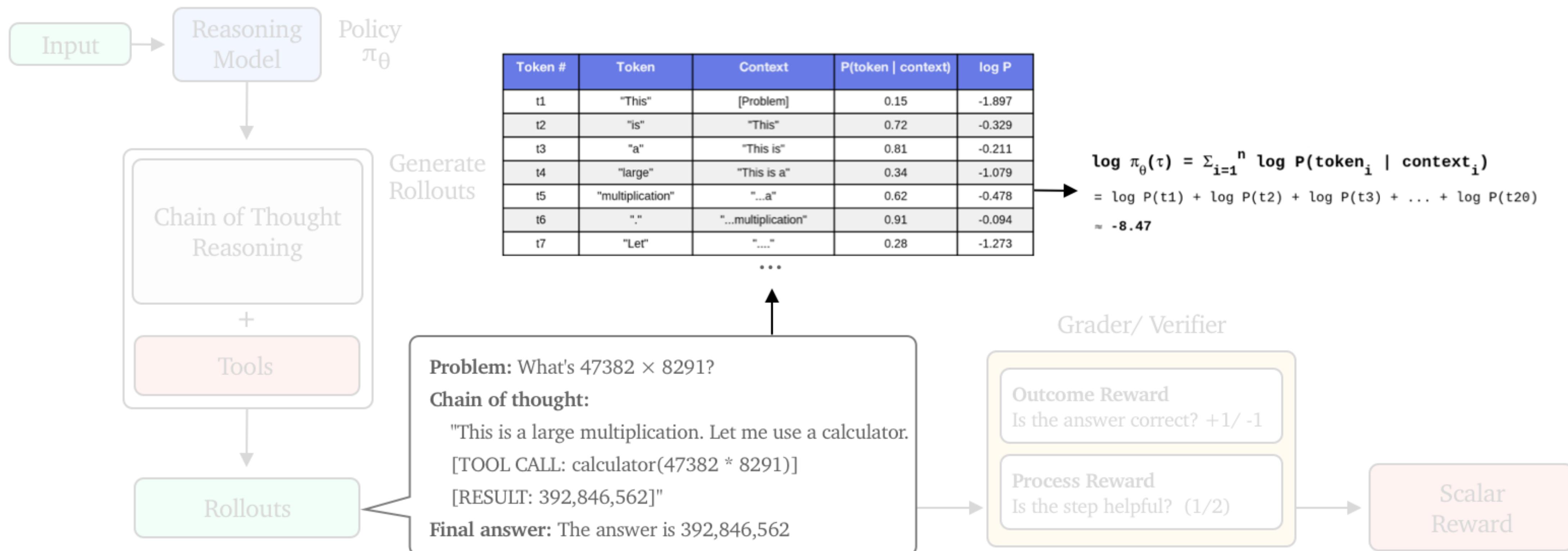
Test-time Compute

- But where is the RL?



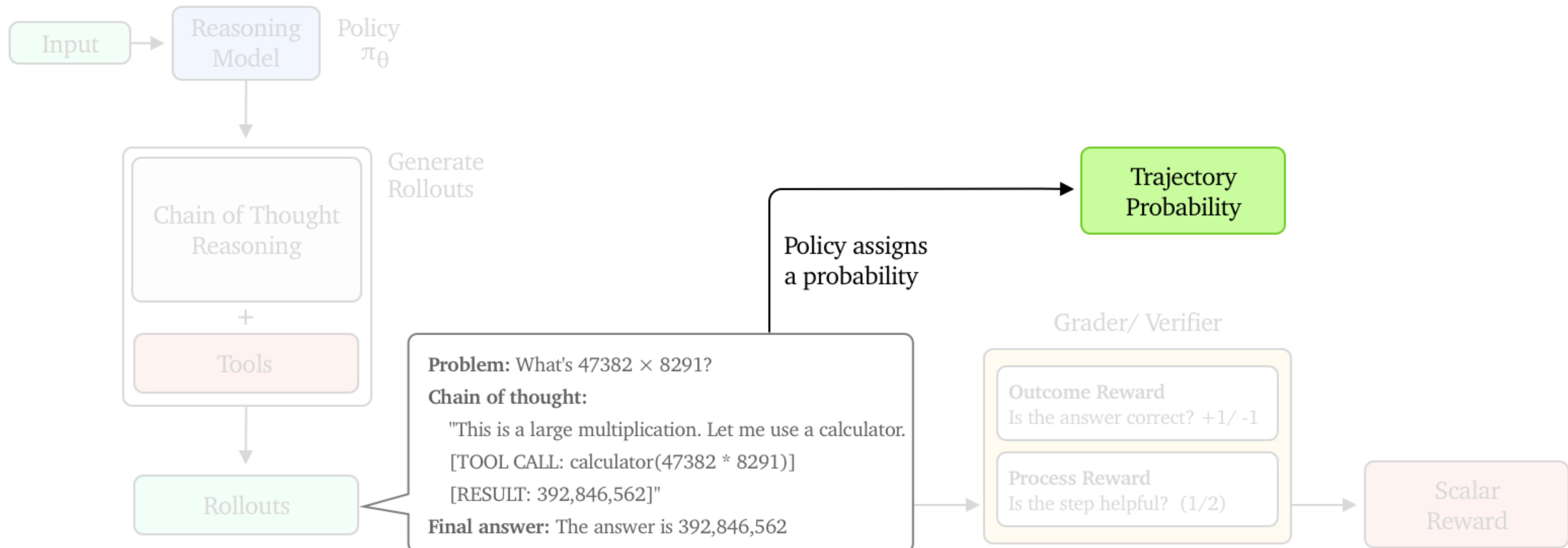
Test-time Compute

- But where is the RL?



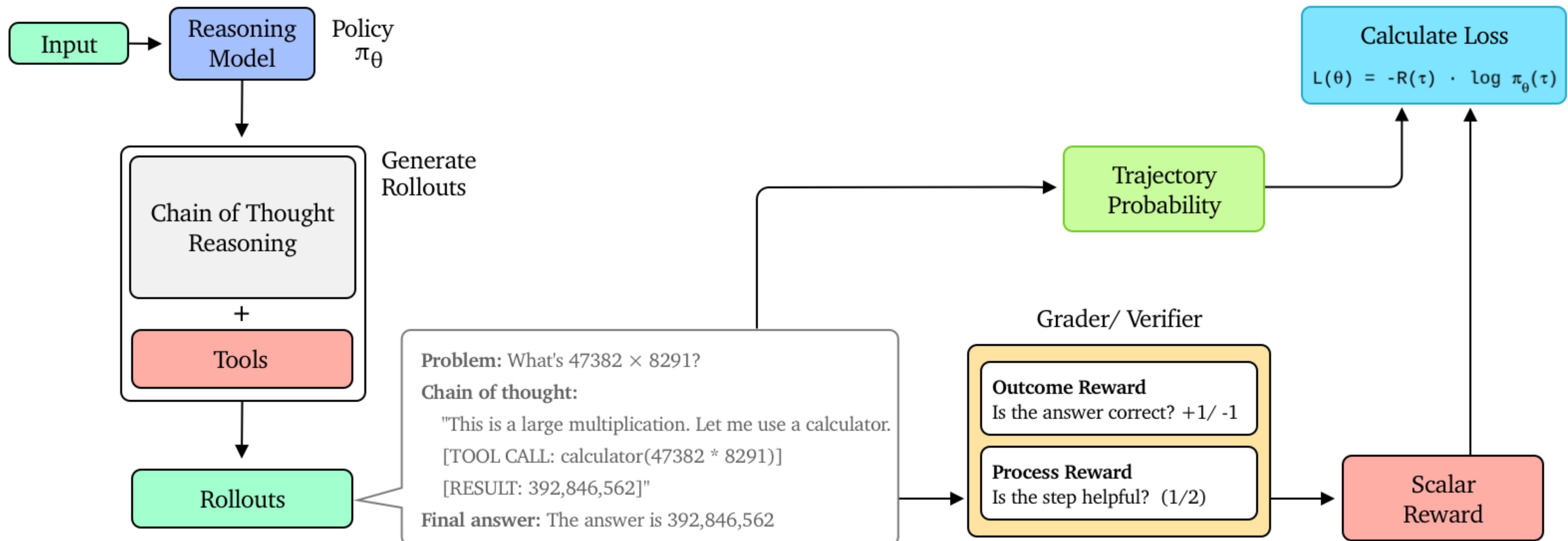
Test-time Compute

- But where is the RL?



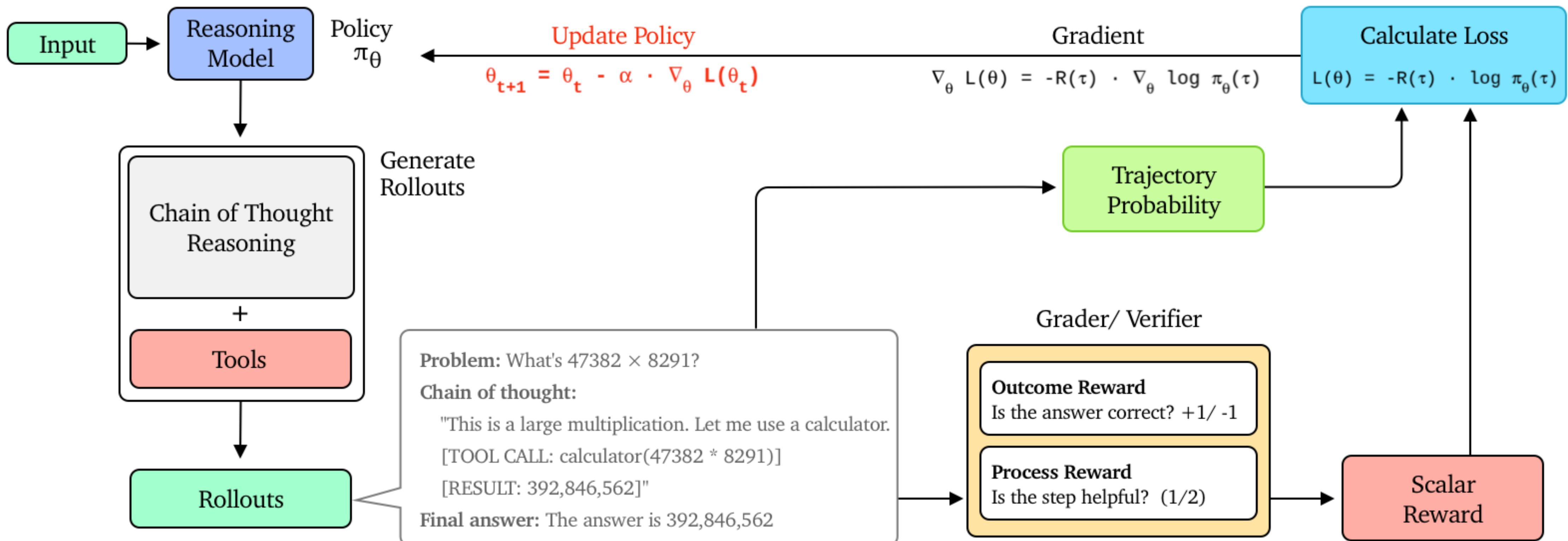
Test-time Compute

- But where is the RL?



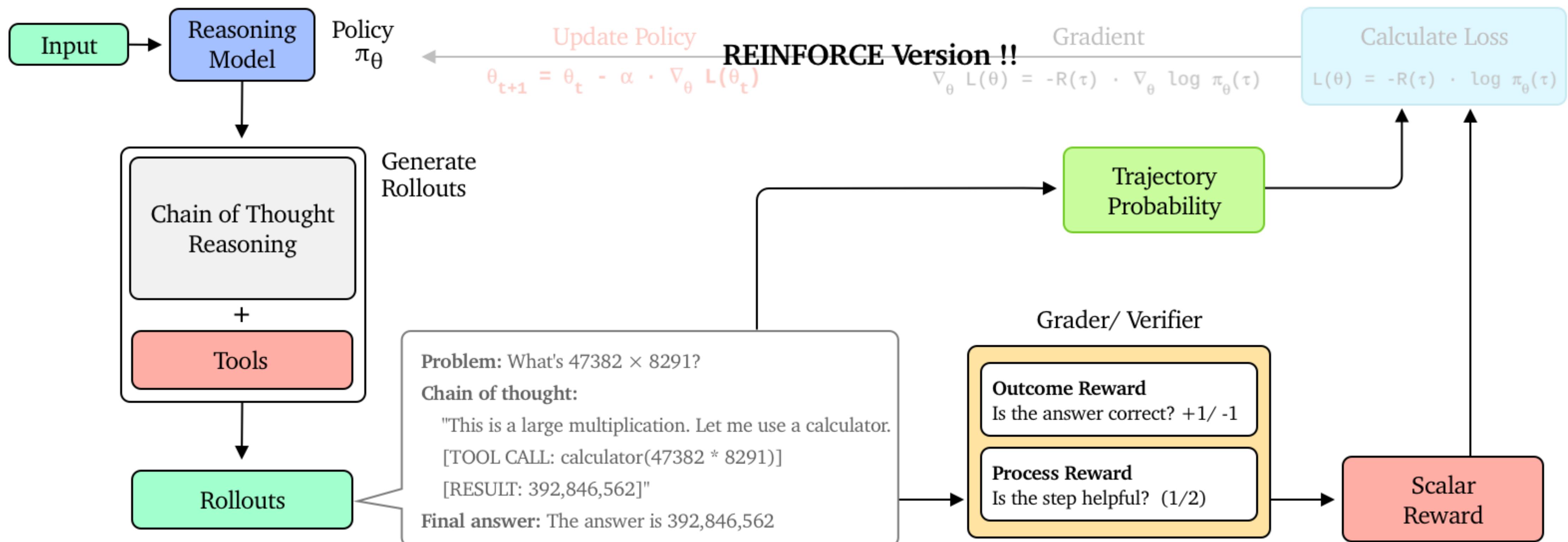
Test-time Compute

- But where is the RL?



Test-time Compute

- But where is the RL?

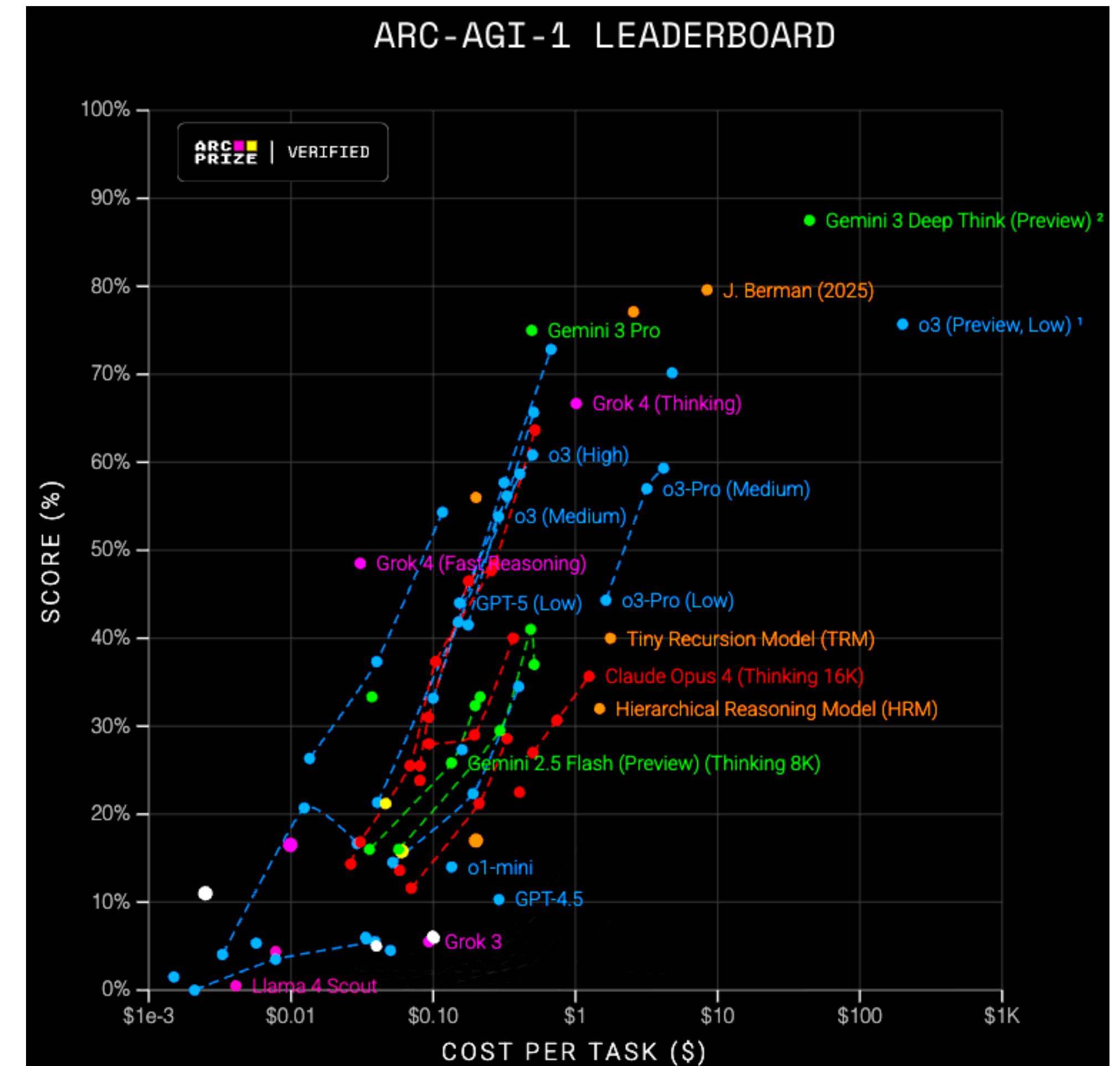


Test-time Compute Summary

- RL can be used to reinforce the thinking process of models
- Using REINFORCE, PPO, GPRO algorithms

Test-time Compute Summary

- RL can be used to reinforce the thinking process of models
- Using REINFORCE, PPO, GPRO algorithms
- Does it work?



Thank You!