GRPO vs PPO

- Both GRPO and PPO are very effective online RL algorithms!

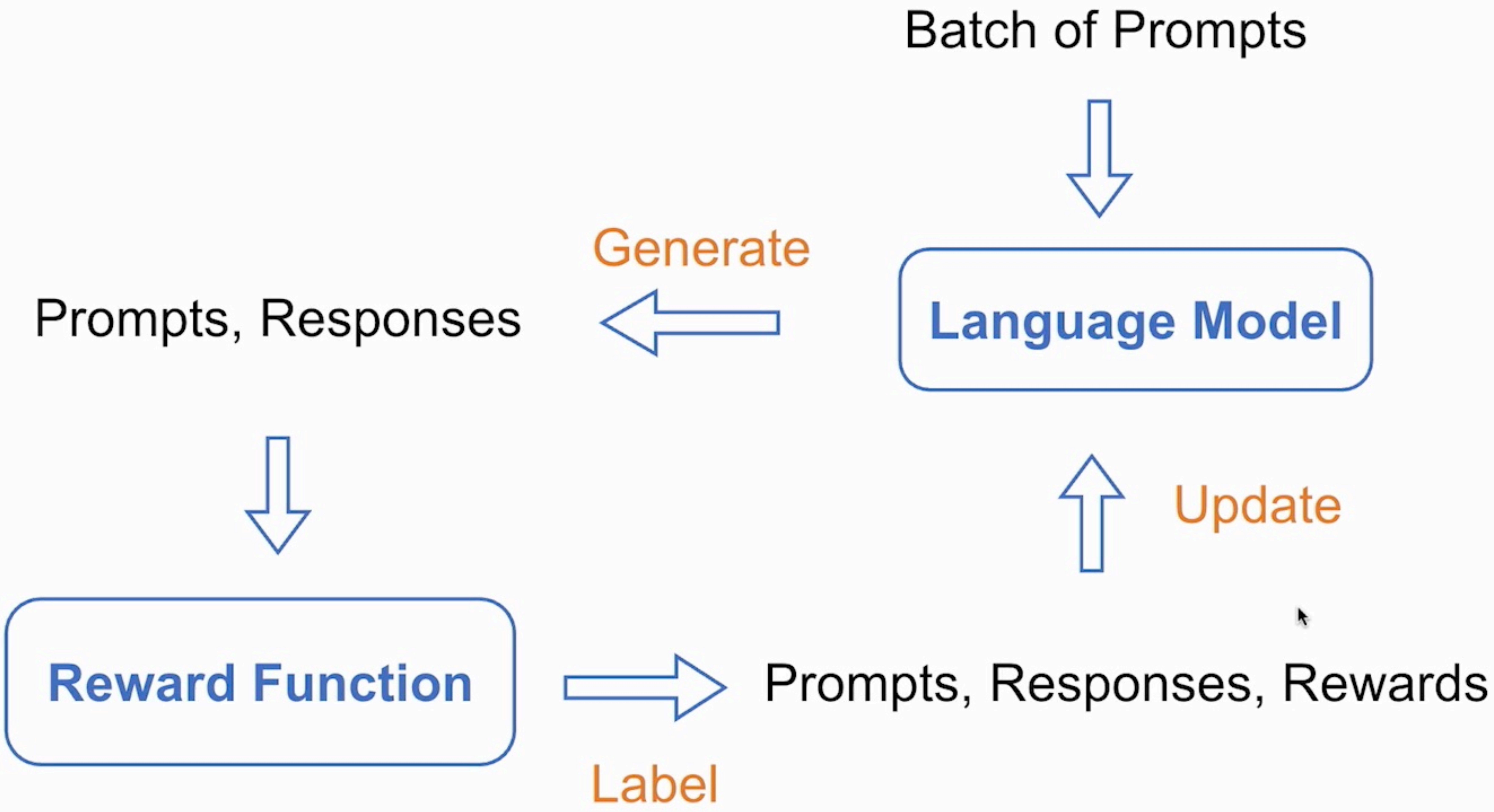
- **GRPO:**
  - Well-suited for binary (often correctness-based) reward
  - Requires larger amount of samples
  - Requires less GPU memory (no value model needed)

- **PPO:**
  - Works well with reward model or binary reward
  - More sample efficient with a well-trained value model
  - Requires more GPU memory (value model)

Reinforcement Learning for LLMs: Online vs Offline

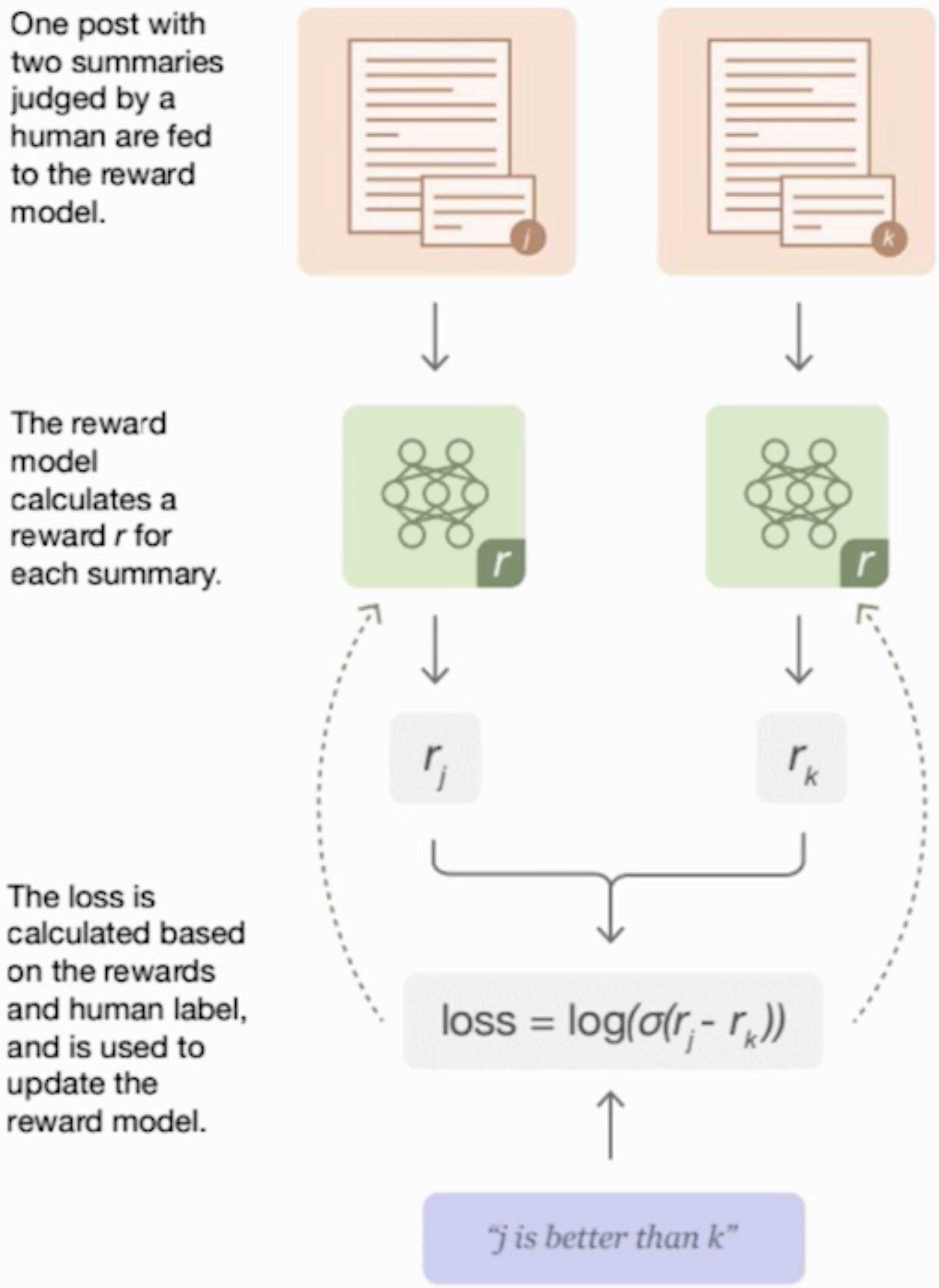
- **Online Learning:**
  - The model learns by generating new responses in real time — it iteratively collects new responses and their reward, updates its weights, and explores new responses as it learns.
- **Offline Learning:**
  - The model learns purely from a pre-collected prompt - response (-reward) tuple. No fresh responses generated during the learning process.

Online RL: Let Model Explore Better Responses by Itself



Reward Function in Online RL

Option 1: Trained Reward Model



- Usually initialized from an existing instruct model, then trained on large-scale human / machine generated preferences data
- Works for any open-ended generations;
- Good for improving chat & safety
- Less accurate for correctness-based domains like coding, math, function calling etc.

Reward Function in Online RL

Option 2: Verifiable Reward

Math: Check if the response matches ground truth

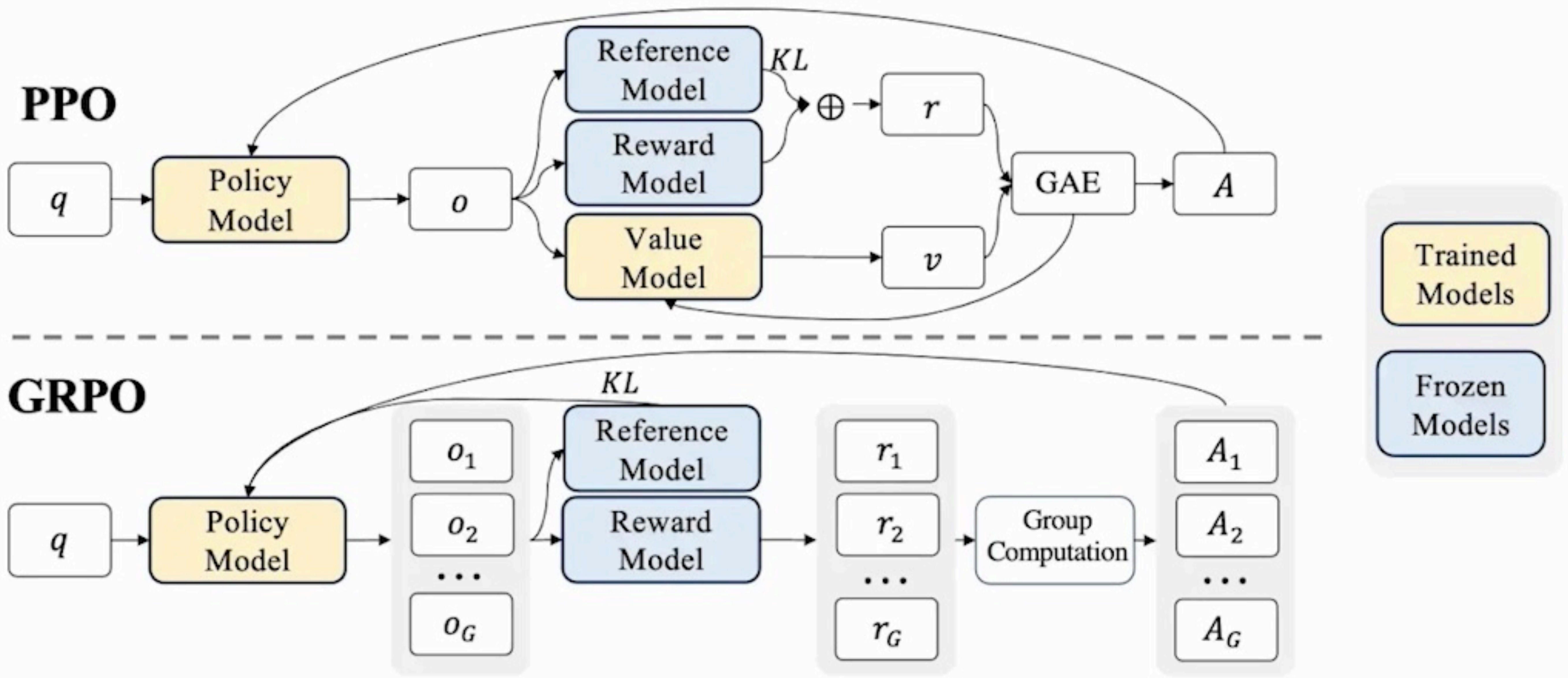
Prompt: What is 1+1-1+1.1-1
Response: The answer is \box{1.1}. ✅
Ground truth: 1.1

Coding: Running unit tests

Prompt: Given a string S, return the longest substring that occurs at least twice.
Response: import …
Test Input 1: “ABCDABCDBC”
Test Output 1: “ABCD”

- Requires preparation of ground truth for math, unit tests for coding, or sandbox execution environment for multi-turn agentic behavior
- More reliable than reward model in those domains
- Used more often for training reasoning models

Policy Training in Online RL



$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip}\left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right]$$

Source: Shao, Zhihong, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).