

# PA<sup>3</sup>: Policy-Aware Agent Alignment through Chain-of-Thought

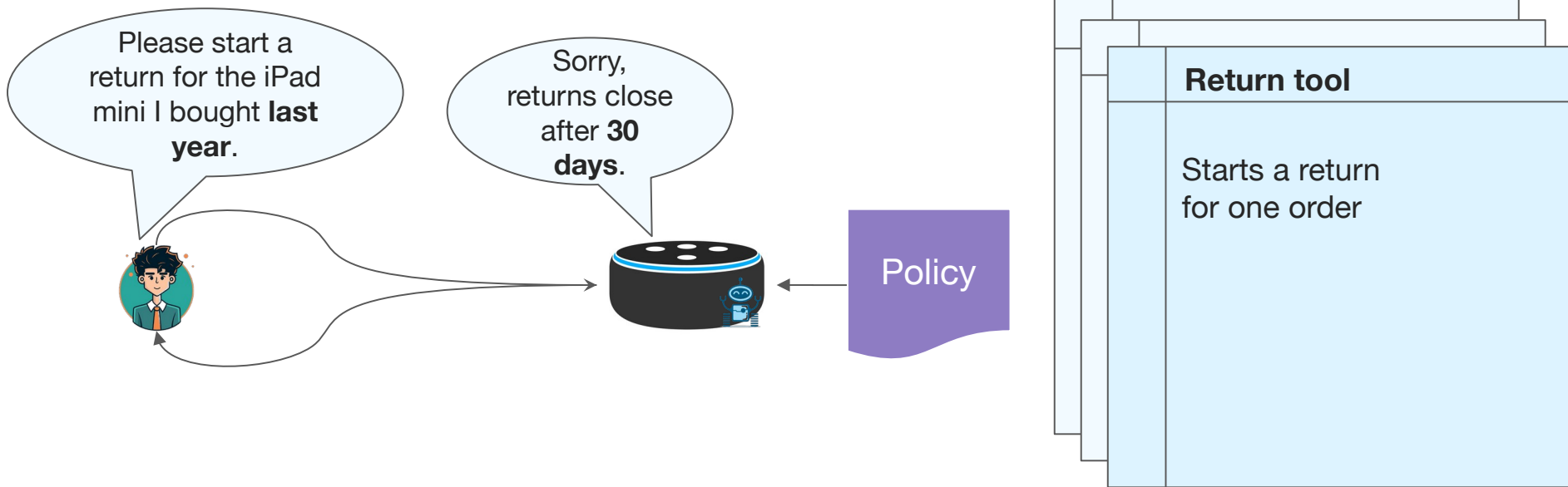
Recall the policy instead of reading it

**Shubhashis Roy Dipta**<sup>1</sup>, Daniel Biś<sup>2</sup>, Kun Zhou<sup>2</sup>,  
Lichao Wang<sup>2</sup>, Benjamin Yao<sup>2</sup>, Chenlei Guo<sup>2</sup>, Ruhi Sarikaya<sup>2</sup>

<sup>1</sup> University of Maryland, Baltimore County   <sup>2</sup> Amazon Alexa AI

sroydip1@umbc.edu

A customer-service agent must **follow the shop's policy** before it calls a tool



# Today we **paste the whole policy** into every prompt, and it works

## Retail Agent Policy:

As a retail agent, you can help users purchase, modify or return orders.

...

→ *You can return up to 30 days after purchase*

...

...



... Please return the iPad mini that I bought last year.

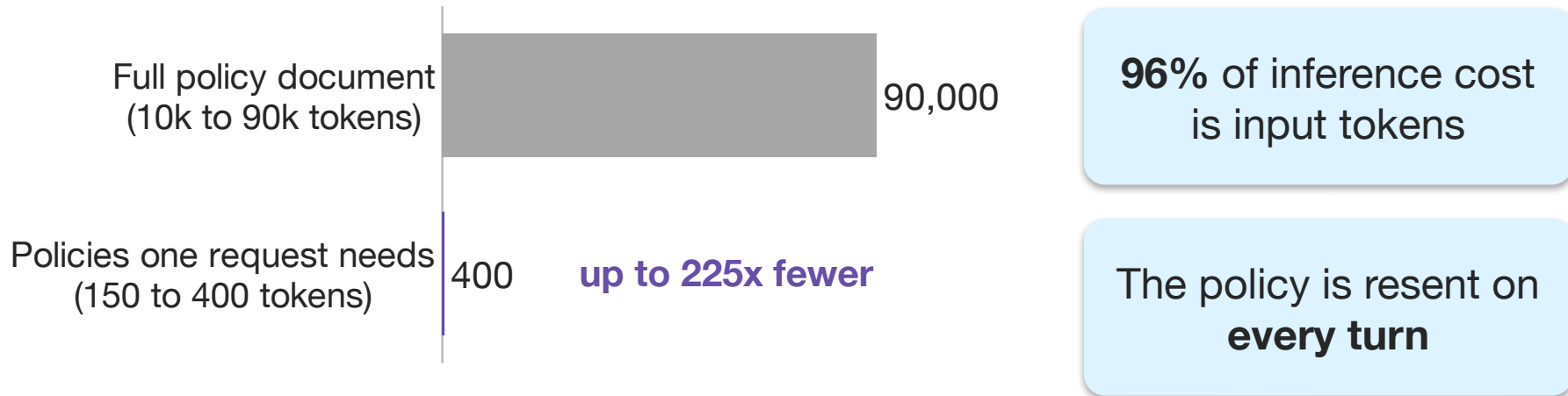


Our return policy doesn't allow that

...



But one request needs **a few rules**, and we resend **the whole document** every turn



Token counts for the upper end of each range.

★ Most of what we pay for is policy text the turn does not need

# The policies that matter are **the ones a model cannot guess**

Real airline policies from  $\tau$ -bench:

- Each reservation can have **at most five** passengers.
- The user can **add but not remove** checked bags.
- If the booking user is a **gold** member, 2 free checked bags for each basic economy passenger, 3 for each economy passenger...
- Basic economy or economy flights can be cancelled **only if travel insurance** is bought and the condition is met...

★ A model cannot know these numbers without the policy

# Fix 1, retrieve or compress: **the rule you need hides behind a condition** the user never says



## Retrieved (top 5)

Each extra baggage is 50 dollars.

...

I'd like to add **two checked bags** to my trip.



Two extra bags: that is **\$100**.



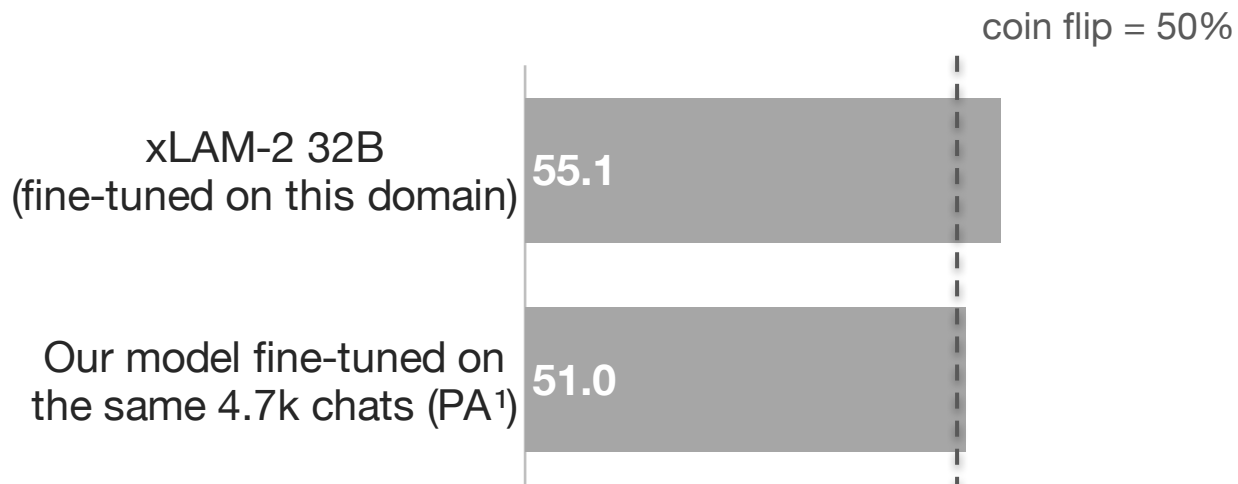
## Missed

If the booking user is a **gold** member, ... 3 free checked bags for each **economy** passenger...

- The request never says **gold** or **economy**, so a retriever has no reason to fetch this rule.
- Compression keeps the text short but can drop exactly these numbers.
- An LLM router (an LLM that picks the policies) must re-read the whole document every turn.

## Fix 2, fine-tune and drop the policy: **the model learns the moves, not the rules**

49 yes/no questions on the airline policy, written by the authors. No policy in the prompt.



★ 4.7k policy-following conversations of training, and still near a coin flip

## The idea

Teach the model to **recall the policy it needs**,  
in its own reasoning.

Then send **no policy** at all.

# At test time the prompt holds **no policy**; the reasoning **recalls the one rule** it needs

## Policy in the prompt

## Retail Agent Policy:  
As a retail agent, you can help  
users purchase, modify or return  
orders.

...  
→ *You can return up to 30 days  
after purchase*  
...



🔥 ~10K  
Tokens

... Please return the iPad mini that I  
bought last year.



Our return policy doesn't allow that  
...

## PA<sup>3</sup> (ours)

## System prompt:  
Respond in the following format:  
<reasoning>reasoning process  
here</reasoning>final response here

😬 <50  
Tokens



... Please return the iPad mini that I  
bought last year.



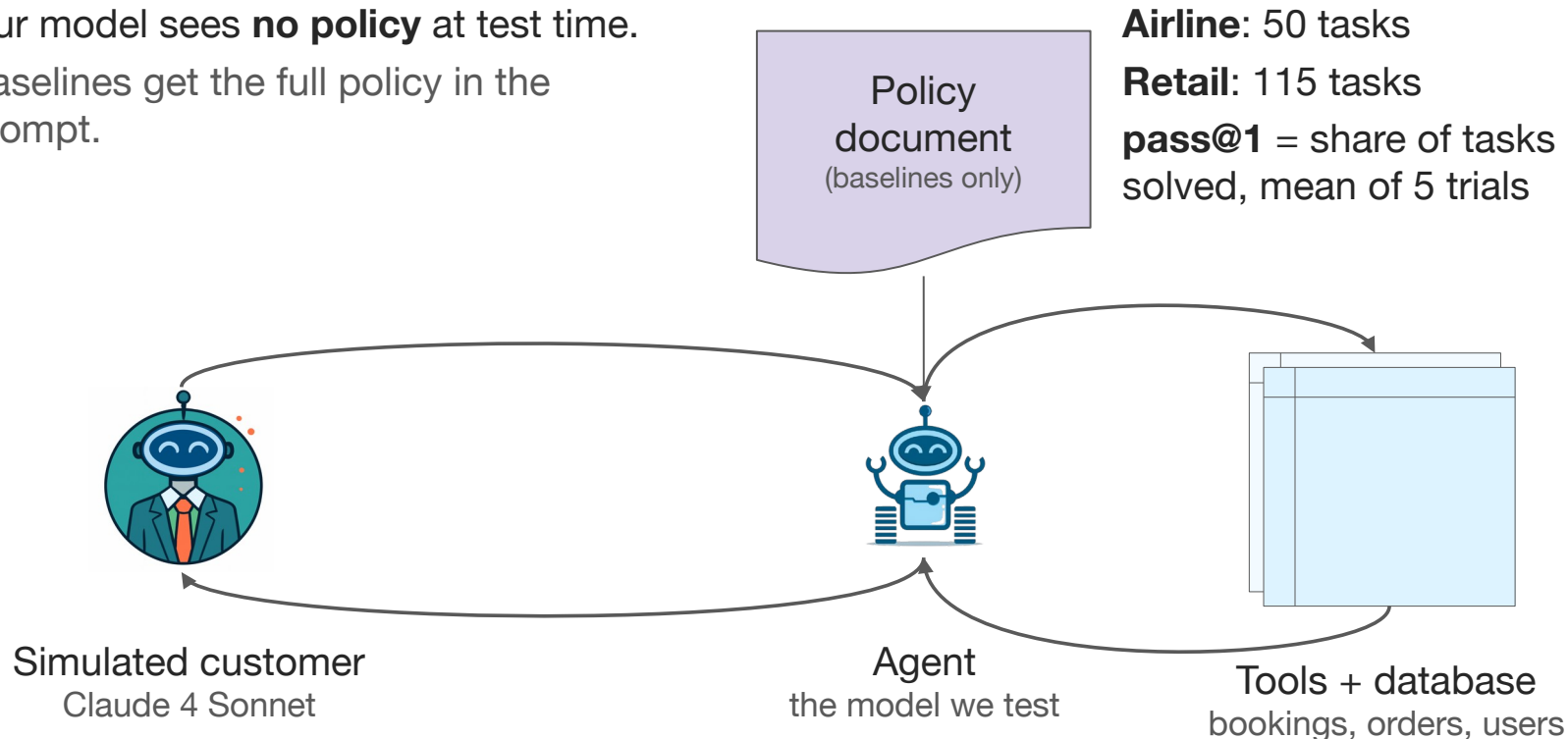
<reasoning> Okay, the user wants to  
return an iPad mini bought last year.  
The policy states that “**Return can  
be initiated in 30 days of  
purchase**”, so...

😊 ~150  
Tokens



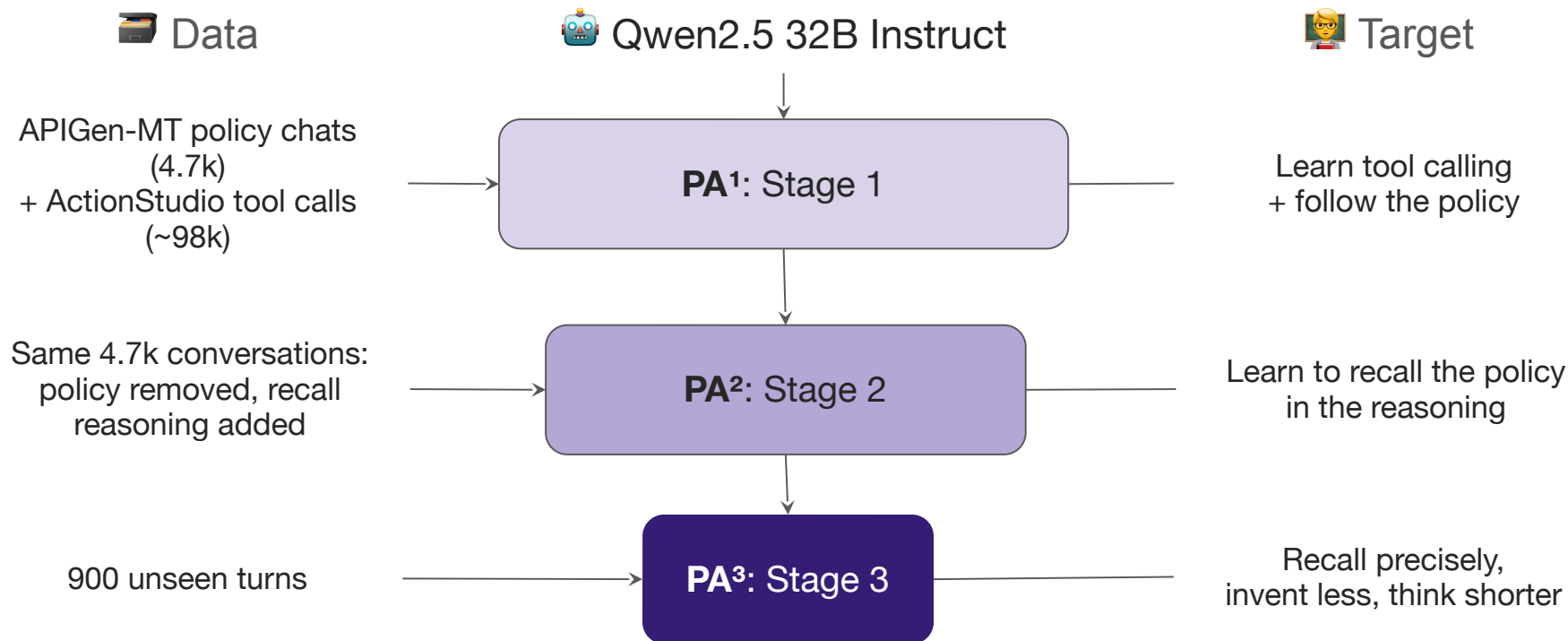
# $\tau$ -bench: a simulated customer chats with the agent; success means the **database ends right**

Our model sees **no policy** at test time.  
Baselines get the full policy in the prompt.

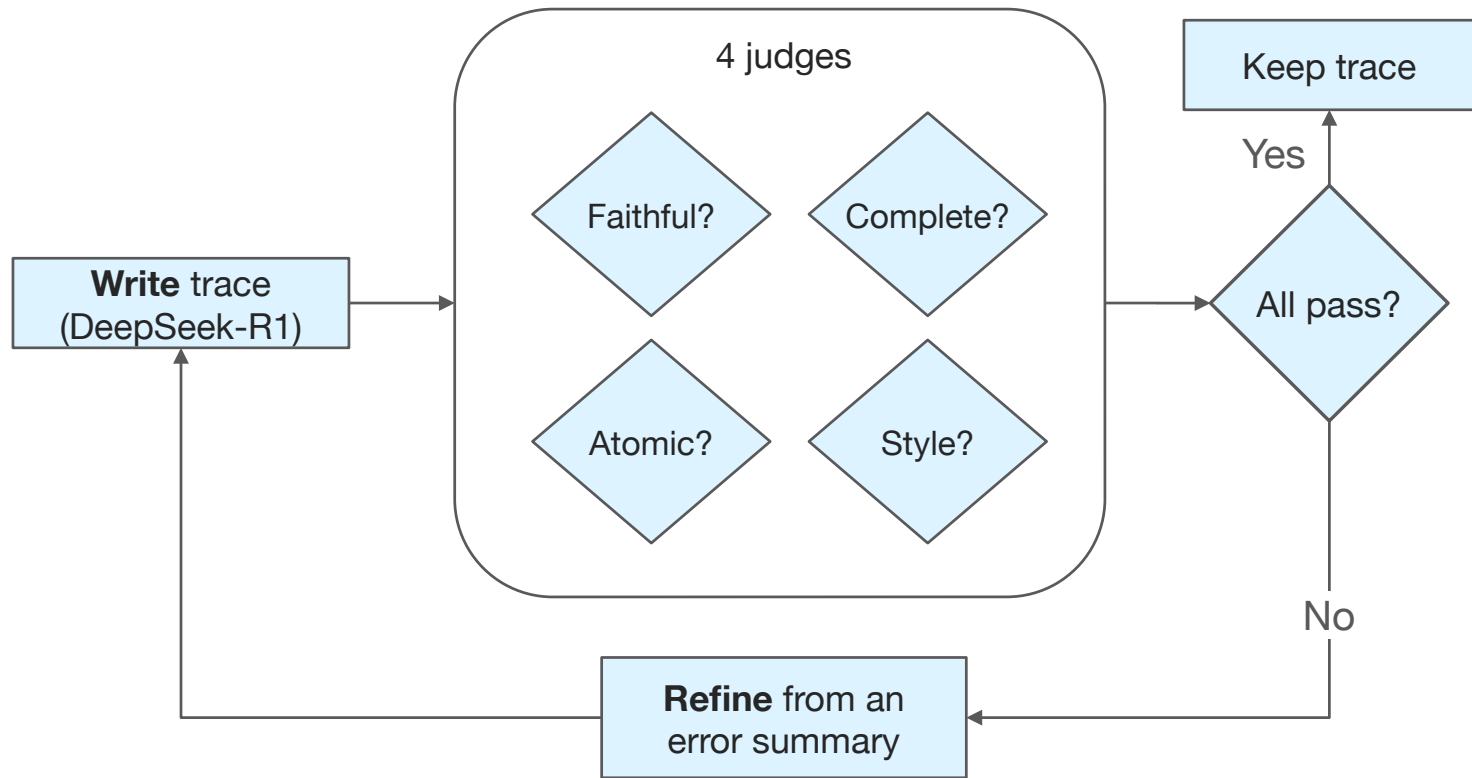


$\tau$ -bench: Yao et al. (2024). Task counts from its Table 1. Success = final database state matches the goal.

# We train one 32B model in **three stages**, and the policy leaves the prompt after Stage 1

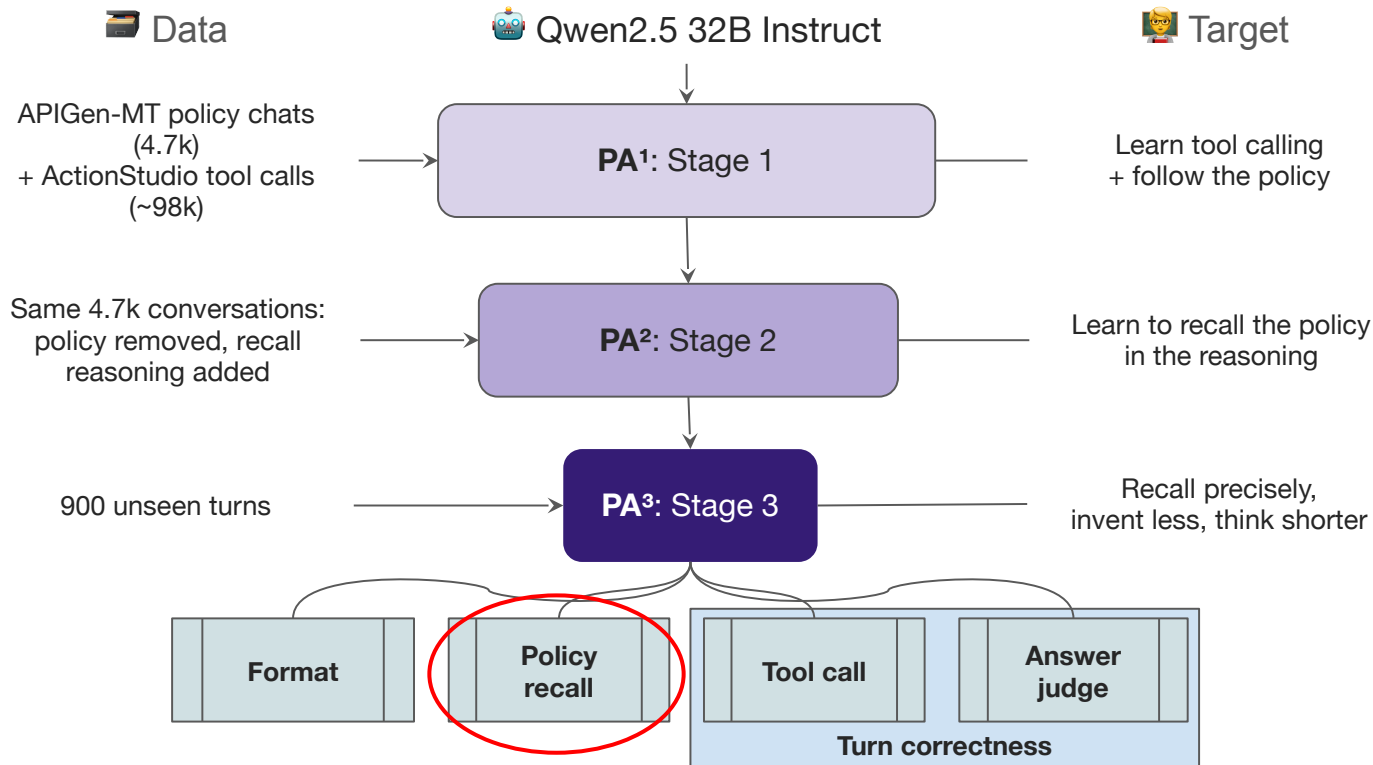


An LLM writes each trace, **four judges grade it**, and failures go back for a rewrite

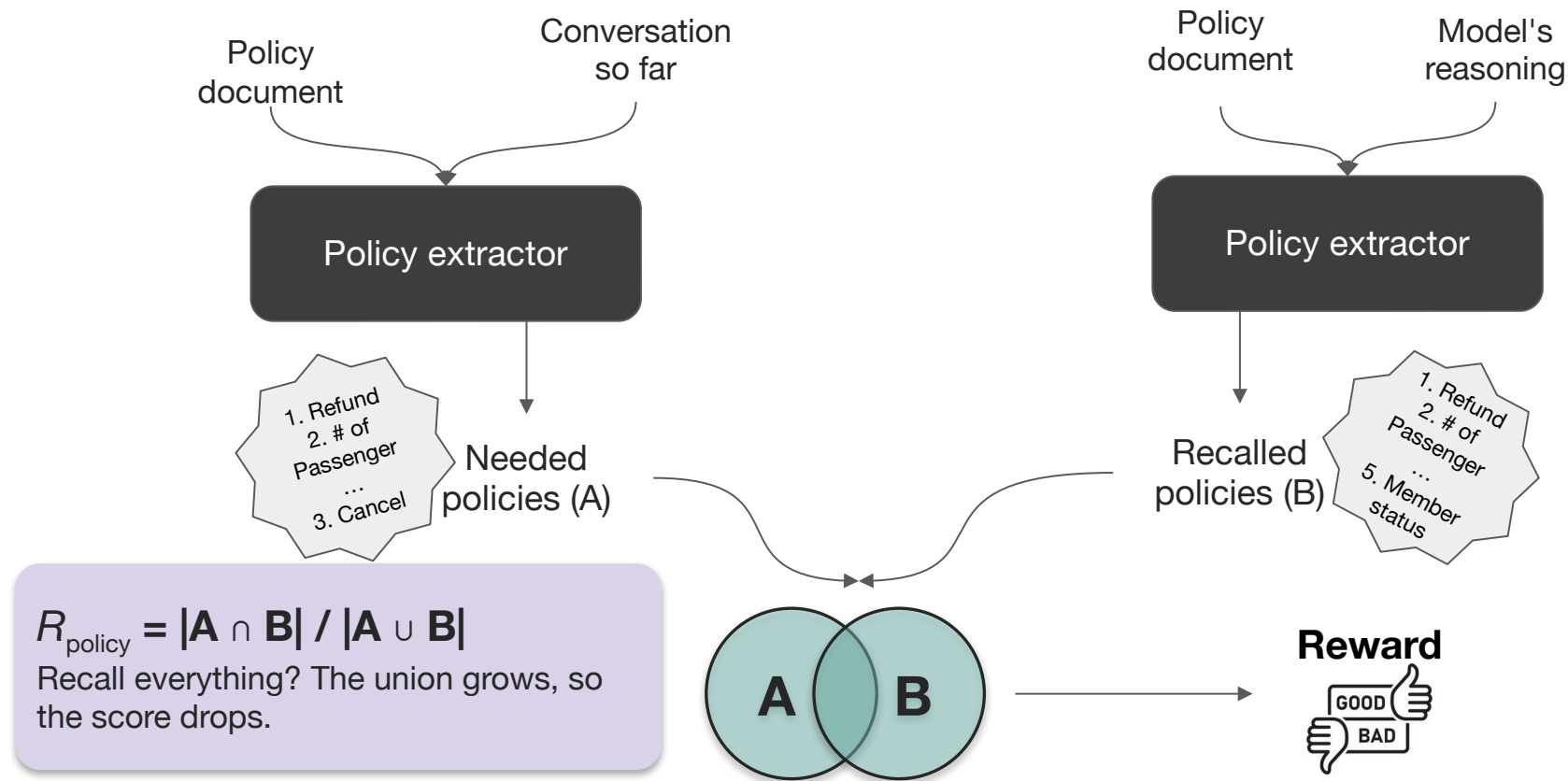


All four judges are Claude 3.5 Sonnet with one rubric each (analyze, then rate 1 to 10). All four bars must pass. At most 4 refine rounds, then the turn is dropped.

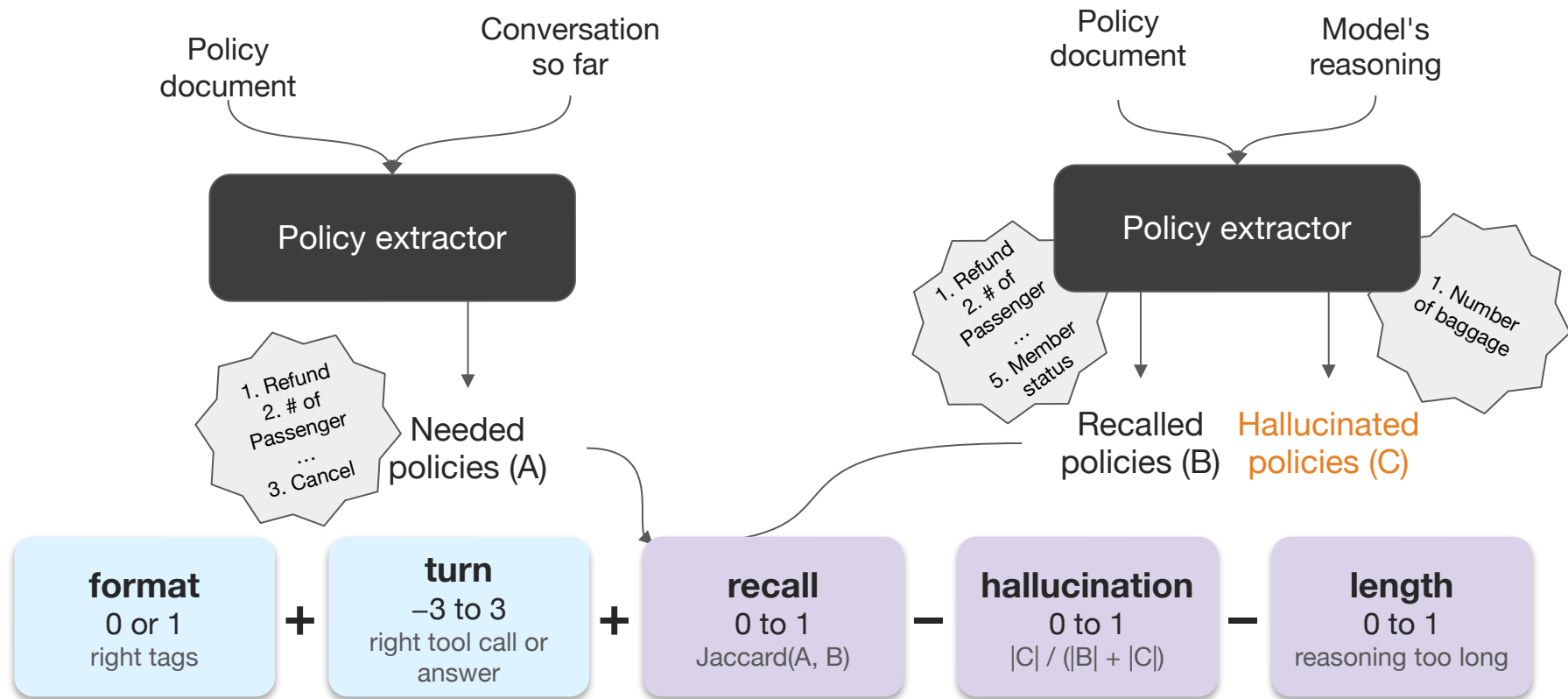
# Stage 3 runs **GRPO**; the new reward term pays for **recalling the right policies**



PolicyRecall reward = **Jaccard(A, B)**: every extra rule lowers it

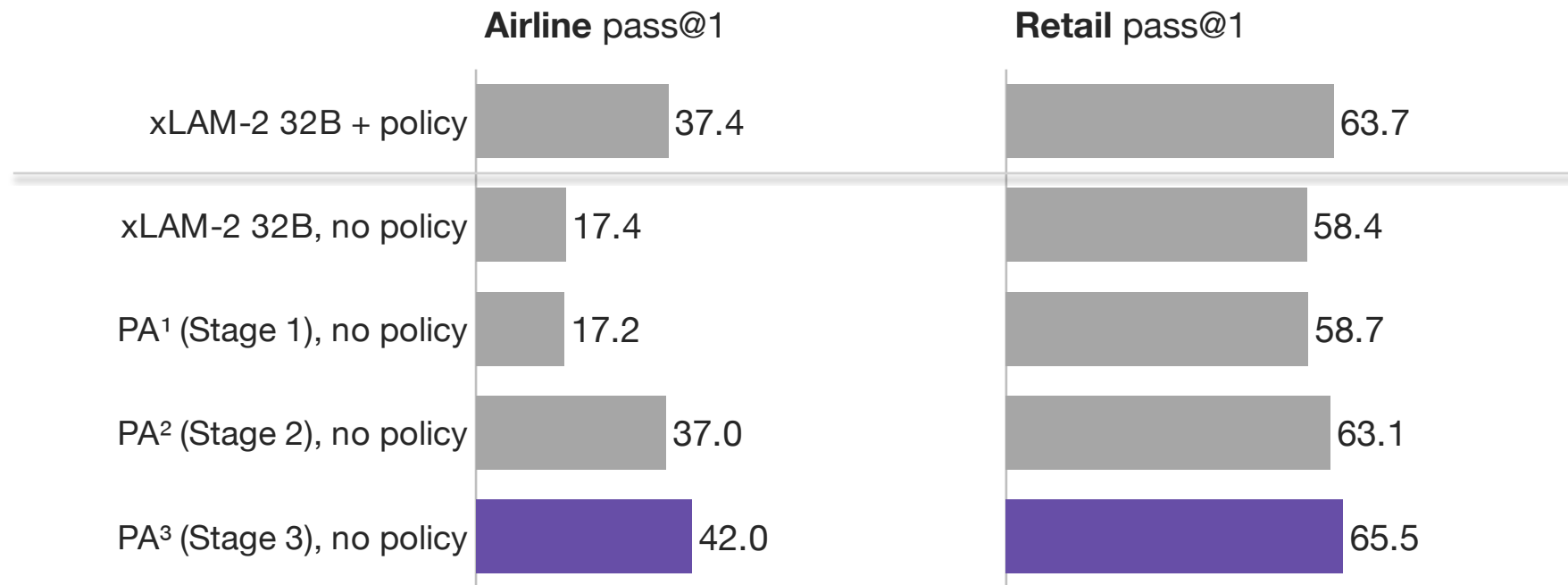


Final reward = **format** + **turn** + **recall** – **hallucination** – **length**



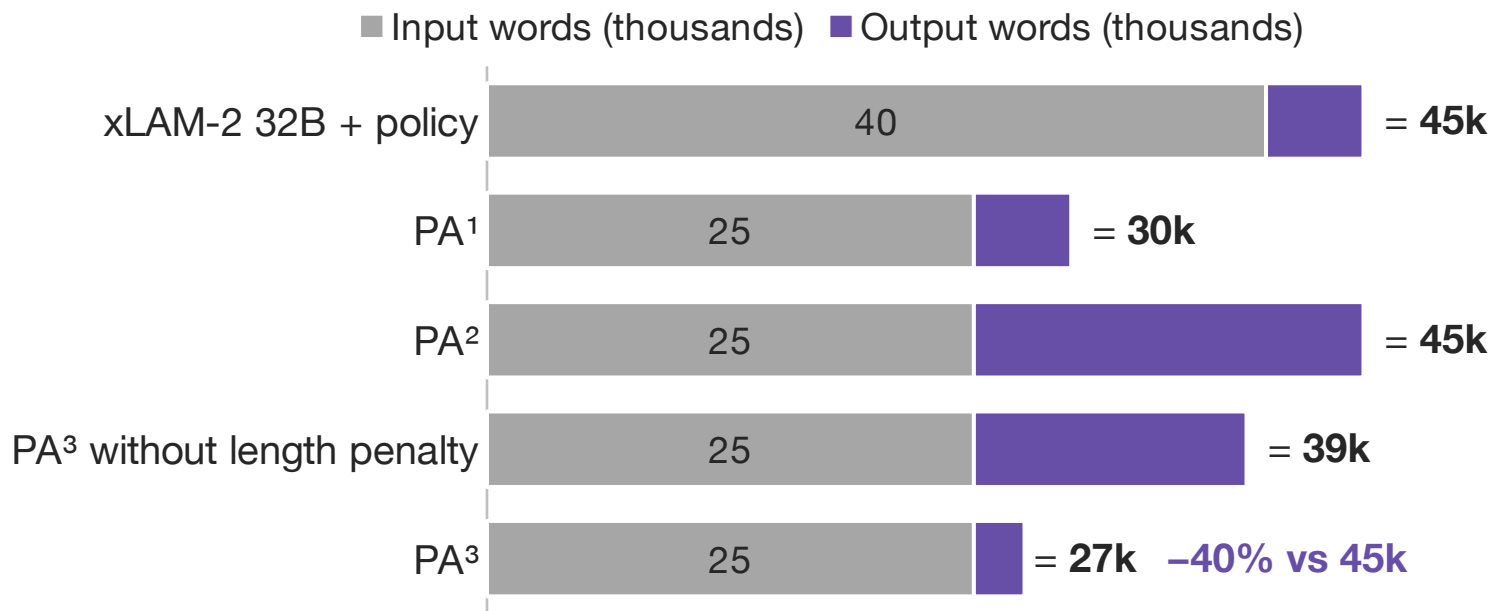
Total range: -5 to 5. Tool-call score follows ToolRL (Qian et al., 2025); the answer judge is Qwen3-32B.

With GRPO, PA<sup>3</sup> **beats xLAM-2 reading the policy**: average 53.8 vs 50.5



★ Airline 17.4 → 42.0 (+141%) over xLAM-2 without policy; retail +12%

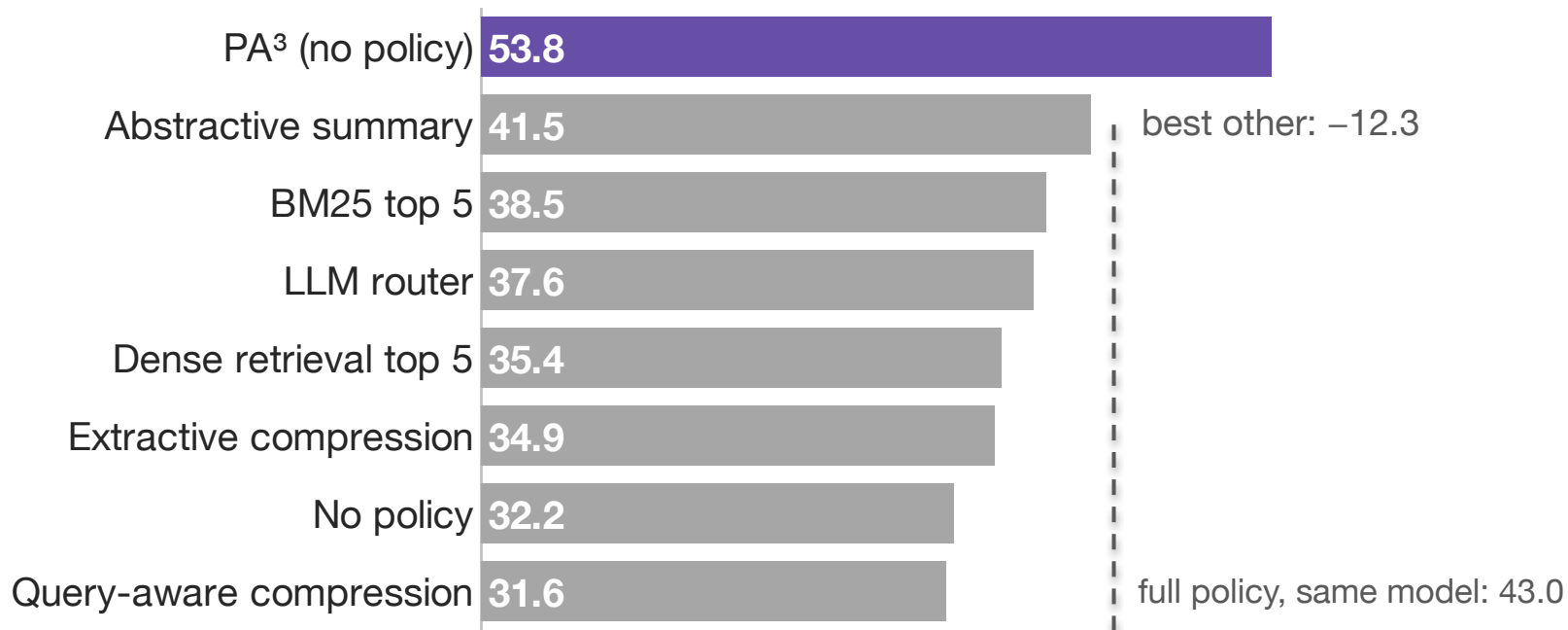
# PA<sup>3</sup> uses **40% fewer words** per conversation than xLAM-2 reading the policy



★ No policy in the input; the length penalty cuts the reasoning from 14k to 2.6k words

Mean words per conversation (words, not tokens, to be fair across tokenizers). Input = prompt (with the policy for xLAM-2), user turns, tool outputs.

# Can retrieval or compression do as well? **No: the best of 26 setups is 12 points behind**

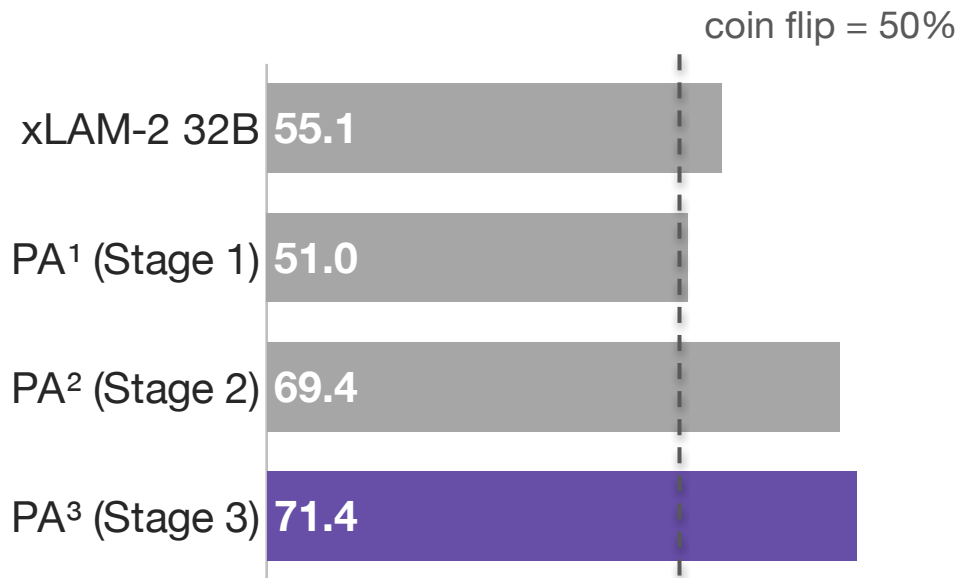


All on the same base model (Qwen2.5 32B, untrained). Full policy on it: 43.0

Average pass@1, airline + retail, 5 trials. Best setting per family. 26 configurations, 47 runs (most on both domains).

# Is the policy really in the weights? Yes/no accuracy goes **from coin flip to 71%**

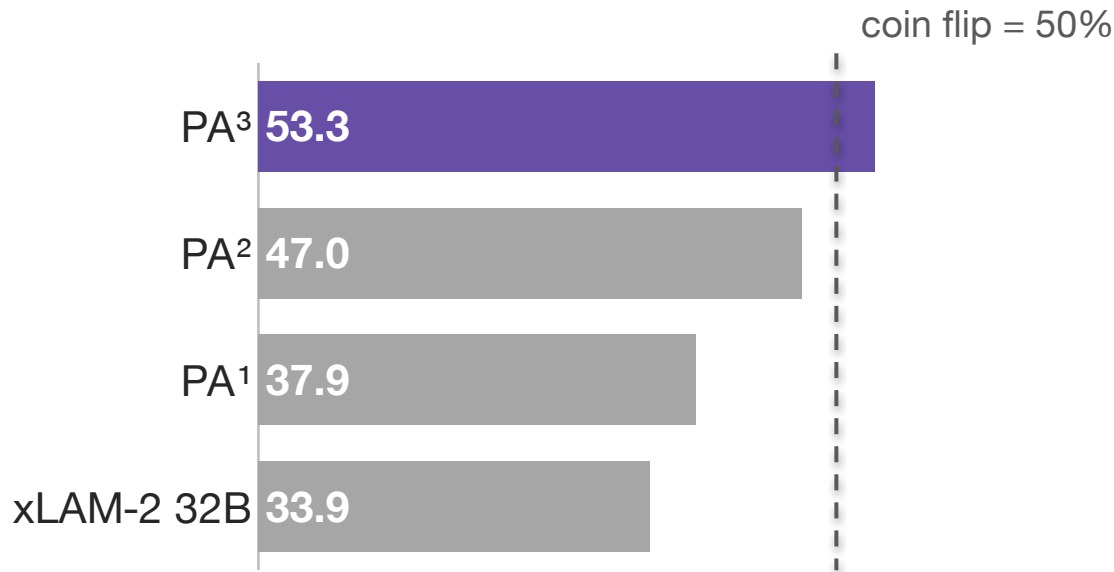
49 yes/no questions on the airline policy, written by the authors. No policy in the prompt.



★ The jump comes with Stage 2, where the model learns to recall the policy

# PA<sup>3</sup> follows a changed policy **only 53% of the time**, near a coin flip

Airline tasks where the one policy the task hinges on is replaced; the new policy is in the prompt. Judge: Claude 3.7 Sonnet.



⚠ Learning the policy does not yet make the model follow a new one

## Takeaway: **recall the policy, don't read it**

- With **no policy in the prompt**, a 32B model beats the same-size model that reads it: **53.8 vs 50.5**, with **40% fewer words**.
- The key is recalling the policy **in the reasoning**. Yes/no policy knowledge goes from coin flip to **71%**.
- The reward pays for recalling **the right policies** and penalizes invented ones.
- Open problem: following a **changed** policy (53%).