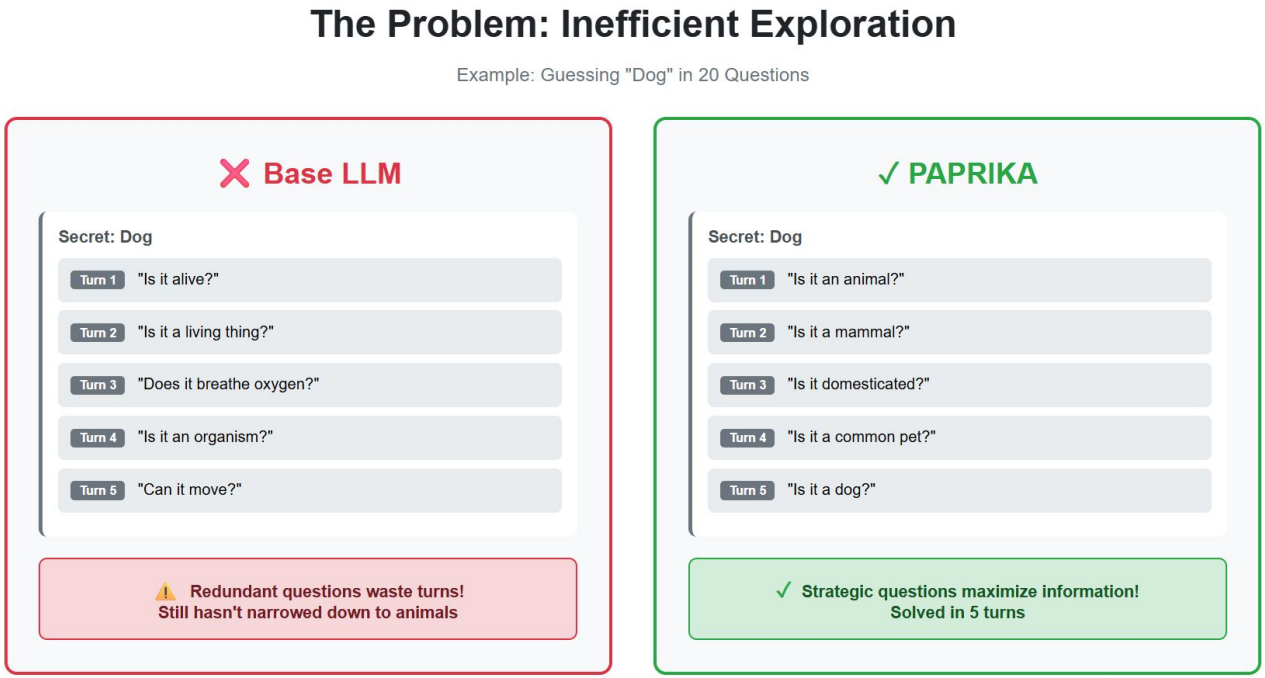
Training a Generally Curious Agent

Fahim Tajwar Yiding Jiang Abitha Thankaraj Sumaita Sadia Rahman J Zico Kolter Jeff Schneider
Russ Salakhutdinov

Motivation

LLMs show great promise as autonomous agents, but they critically lack the ability to strategically explore and gather information when interacting with new environments.

Motivation


The Problem: Inefficient Exploration
Example: Guessing "Dog" in 20 Questions
❌ Base LLM
Secret: Dog
Turn 1 "Is it alive?"
Turn 2 "Is it a living thing?"
Turn 3 "Does it breathe oxygen?"
Turn 4 "Is it an organism?"
Turn 5 "Can it move?"
⚠️ Redundant questions waste turns!
Still hasn't narrowed down to animals
✓ PAPRIKA
Secret: Dog
Turn 1 "Is it an animal?"
Turn 2 "Is it a mammal?"
Turn 3 "Is it domesticated?"
Turn 4 "Is it a common pet?"
Turn 5 "Is it a dog?"
✓ Strategic questions maximize information!
Solved in 5 turns

The core challenge:

- Naturally occurring data lacks the interaction structure needed to teach exploration
- Generating synthetic data for every possible task is simply infeasible at scale
- Deploying models in the real world to collect data is expensive and risky

The key insight:

Instead of training models on specific tasks, we can train them on the general process of solving tasks through in-context RL, so they can adapt to entirely new environments without any additional training.

Task Formulation

Partially Observable Markov Decision Process (POMDP)

Agent cannot see full environment state
Must interact to gather information

Task τ

Single problem instance
Example: Guess "apple" in 20 Questions

Group G

$G = \{\tau_1, \tau_2, \ldots, \tau_{|G|}\}$

Collection of similar tasks
Example: All 20 Questions games

Episode h

$h = (o_0, a_0, \ldots, o_k, a_k)$

Full interaction sequence in one task

Action a_t & Observation o_t

Both are text strings
Agent sends text, receives text

Reward r(h)

Single score at episode end
No intermediate feedback

Performance

$Perf(G) = 1/|G| \Sigma \mathbb{E}[r(h)]$

Average reward over all tasks in group

Critical Setup

Train on G_{train} → Test on completely different G_{test} to measure generalization

PAPRIKA: Task Design

The authors design a suite of tasks specifically to teach strategic information gathering that generalizes across environments.

- **Text-based interactions** — Both actions and observations are natural language strings, making LLMs a natural fit as the agent
- **Multi-turn sequential decision making** — The agent must understand prior history in its context and choose actions that maximize future success, not just respond to a single prompt
- **Partially observable** — Observations do not reveal the full state or hidden information, forcing the agent to actively explore while also exploiting what it knows
- **Diverse strategies required** — Different tasks should require genuinely different exploration approaches, preventing the model from overfitting to a single strategy

Task Group

Table 1. Summary of the task groups used by PAPRIKA.

Task Group	# Train Tasks	# Test Tasks	Maximum Turns	Env Feedback	Uses COT
Twenty questions	1499	367	20	LLM generated	✗
Guess my city	500	185	20	LLM generated	✗
Wordle	1515	800	6	Hardcoded program	✓
Cellular automata	1000	500	6	Hardcoded program	✓
Customer service	628	200	20	LLM generated	✗
Murder mystery	203	50	20	LLM generated	✗
Mastermind	1000	500	12	Hardcoded program	✓
Battleship	1000	200	20	Hardcoded program	✓
Minesweeper	1000	200	20	Hardcoded program	✓
Bandit best arm selection	81	1	21	Hardcoded program	✓

Dataset Construction

Goal: Generate diverse trajectories covering both successful and unsuccessful exploration strategies

Step 1
Generate Diversity

Sample 20 trajectories per task using high-temperature Min-p sampling

→ Produces diverse exploration behaviors with both good and bad strategies

Step 2
Create Pairs

Pair best trajectory (succeeds in fewest turns) with a **randomly chosen worse trajectory** (fails or takes more turns)

→ Random selection maintains diversity

Step 3
Filter Bad Data

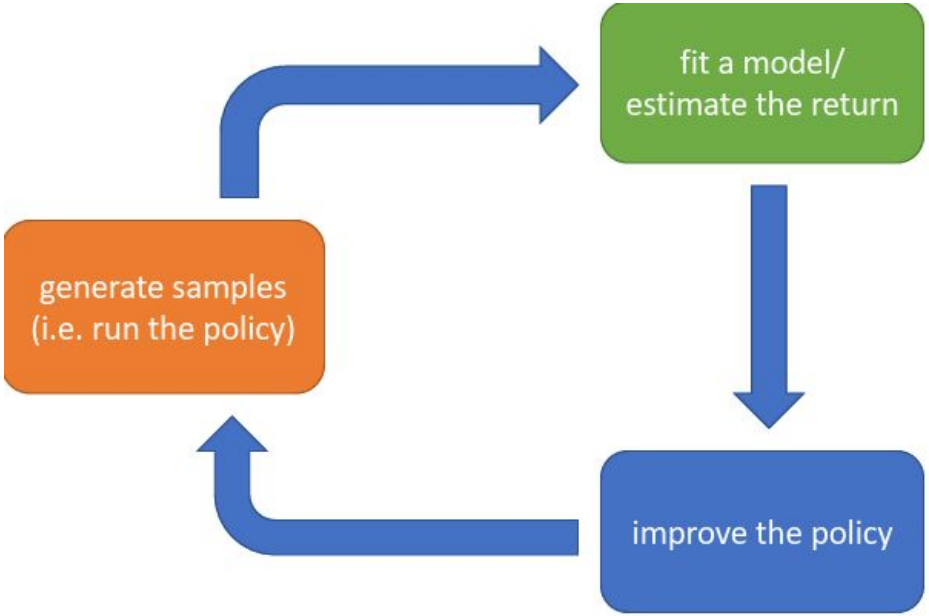
Use LLM judges and validators to remove trajectories where the environment gave incorrect feedback or was exploited

→ Ensures high-quality training signal

📊 **Final Dataset**

17,181	5,260
Trajectories for SFT	Preference Pairs for RPO

RL Premier


fit a model/
estimate the return
generate samples
(i.e. run the policy)
improve the policy

Optimization

- Reasoning Preference Optimization
- SFT + DPO

Supervised Fine Tuning

- Maximize Likelihood of Winning Trajectory
- “Copy the Winner”

SFT - “Copy the Winner”

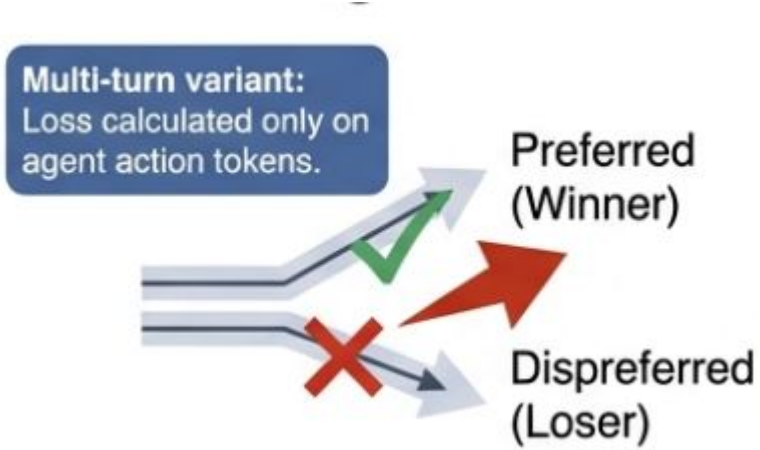
- "Look at these specific winning games and try to **repeat the exact same types** of guesses."

SFT - “Copy the Winner”

- Goal: It sets a baseline of "decent" behavior so the model doesn't just output random text.

Direct Preference Optimization

- Train the model to explicitly prefer the winning trajectory over the losing one.
- “Understand Why you Won”



DPO – “Understand Why you Won”

- “**Focus on the difference** between the smart questions in the winning game and the redundant ones in the losing game".

DPO – “Understand Why you Won”

- Goal: It teaches the model **Strategic Nuance**.
  - Not just copying
  - Learning to actively avoid mistakes it previously made

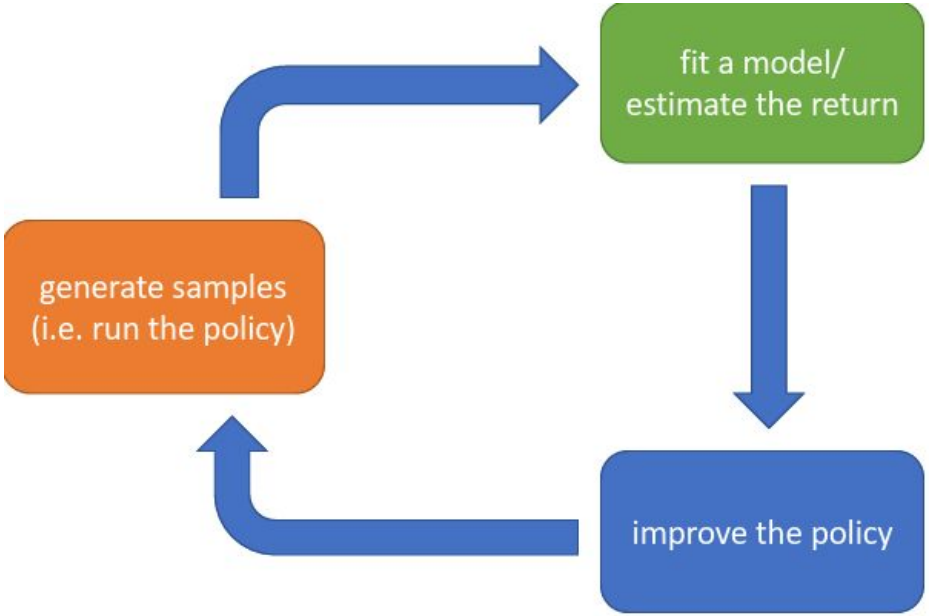
Reasoning Preference Optimization

- DPO Reduces the Probability of Prefered Trajectories
- Combines SFT + DPO to prevent unlearning

$$L_{\text{RPO}} = L_{\text{DPO}} + \alpha \cdot L_{\text{SFT}}$$

α: Weighting hyperparameter

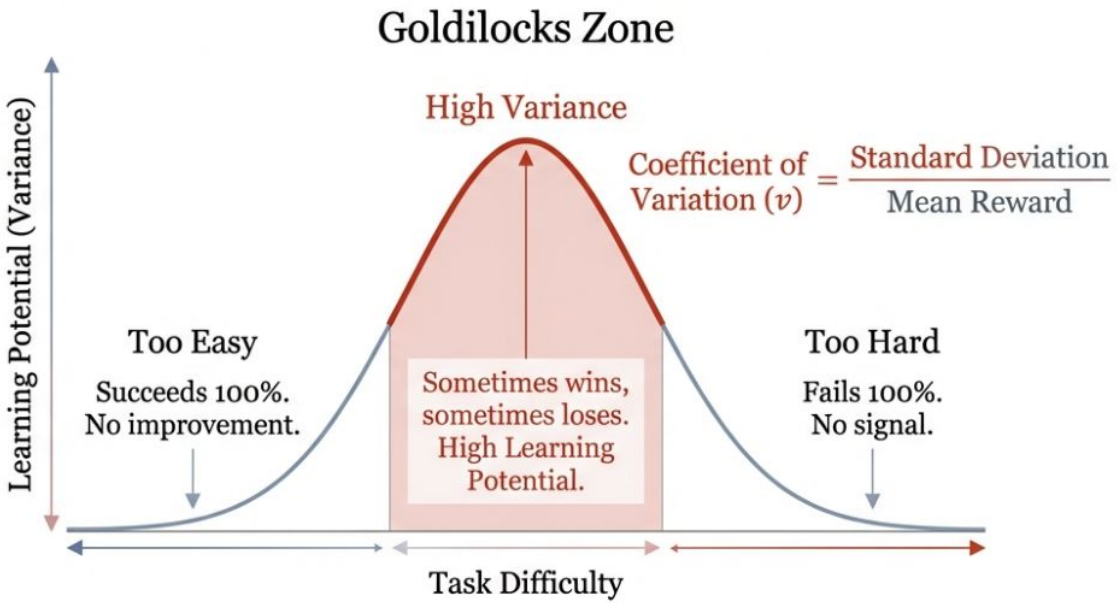
RL Premier


fit a model/
estimate the return
generate samples
(i.e. run the policy)
improve the policy

The Bottleneck: Sample Efficiency

- Thousand Trajectories -> Expensive
- Not All Trajectories are Equal
- Uniform Sampling -> Bad Idea

The Godilocks Zone



Sample Efficiency Goal

- Prioritize Task with Highest Learning Potential

Scalable Online Curriculum Learning

- The model acts like a “Student” focusing on the subjects where they are making the fastest progress that are selected by the “Teacher” UCB.

Learning Potential - Coefficient of Variation

$$\nu_\pi(\tau) = \frac{\sqrt{\sigma_\pi^2(\tau)}}{R_\pi(\tau)}$$

- High Variability (σ)
  - Indicates the agent is on the verge of a breakthrough.
- Mean Reward (R)
  - Normalizes the score across different task types.

Upper Confidence Bound (UCB)

- “Teacher” to Prioritize Tasks in the Godilock Zone

Upper Confidence Bound (UCB) - “Teacher”

- Selects Task Groups by Balancing:
  - Exploitation (Practicing Known High-potential Tasks)
  - Exploration (Trying Unpracticed Tasks)

Upper Confidence Bound (UCB) - “Teacher”

- For each task group k, a priority score g_k is calculated:

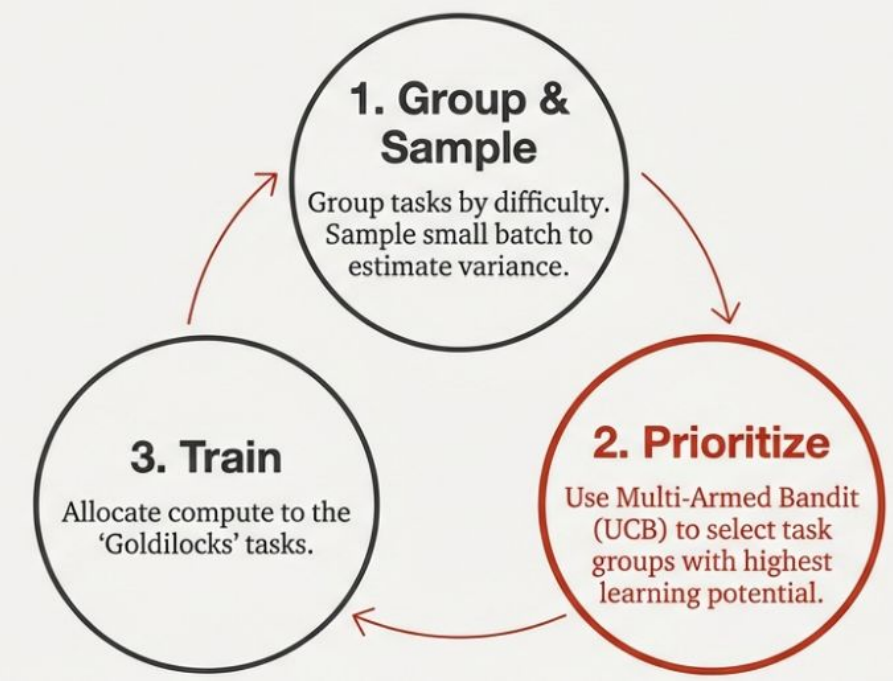
$$g_k = \underbrace{\frac{s_k}{n_k}}_{\text{Exploitation}} + \underbrace{\sqrt{\frac{2 \log \sum n_k}{n_k}}}_{\text{Exploration}}$$

Upper Confidence Bound (UCB) - “Teacher”

$$g_k = \underbrace{\frac{s_k}{n_k}}_{\text{Exploitation}} + \underbrace{\sqrt{\frac{2 \log \sum n_k}{n_k}}}_{\text{Exploration}}$$

- s_k(Sum of Potential):
  - Cumulative learning potential observed for group .
- n_k (Visit Count)
  - Number of times this task group has been sampled.

Scalable Online Curriculum Learning



Scalable Online Curriculum Learning

- The model acts like a “Student” focusing on the subjects where they are making the fastest progress that are selected by the “Teacher” UCB.

Experiment Setup

- **Models:** Llama-3.1-8B-Instruct and Gemma-3-12B-IT
- **Evaluation:** 4 trajectories per test task, temperature 0.7, Min-p 0.3
- Report average success rate across the 4 rollouts

PAPRIKA Improves Performance

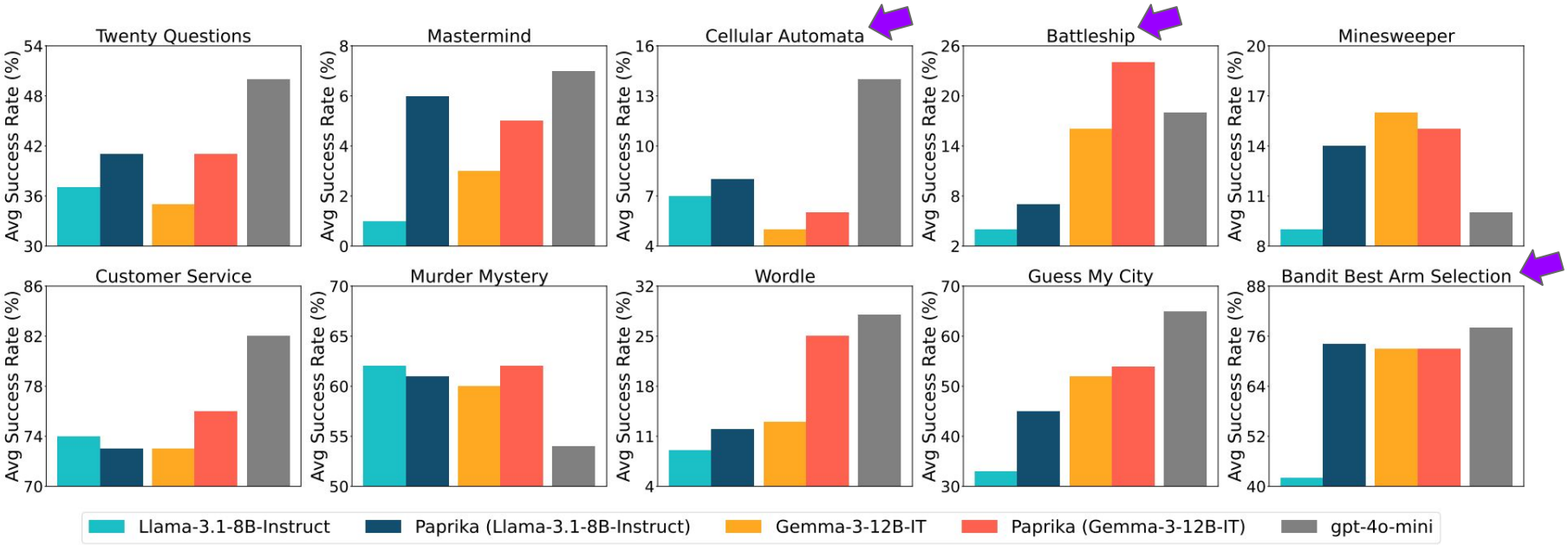


Figure 2. **(PAPRIKA improves success rate on a diverse range of task groups)** Average success rate on all 10 task groups at temperature 0.7. PAPRIKA generally improves performance of both Llama-3.1-8B-Instruct and Gemma-3-12B-IT models.

Zero-Shot Generalization: Leave-One-Out (LOO) Experiments

🎯 Example: Testing on Bandit

✓ TRAIN on 9 Groups:

20 Questions	Wordle	Battleship
Minesweeper	Mastermind	Cellular Automata
Customer Service	Murder Mystery	Guess My City

↓

✗ TEST on Held-Out Group:

Bandit ✗

❓ Can exploration strategies transfer?

Can the model use strategies learned from **Battleship** and **Wordle** to improve on **Bandit** without ever seeing any bandit examples?

🔄 We Repeat for ALL 10 Groups

LOO Experiment 1: Train on Groups 2-10 → Test on Group 1

LOO Experiment 2: Train on Groups 1,3-10 → Test on Group 2

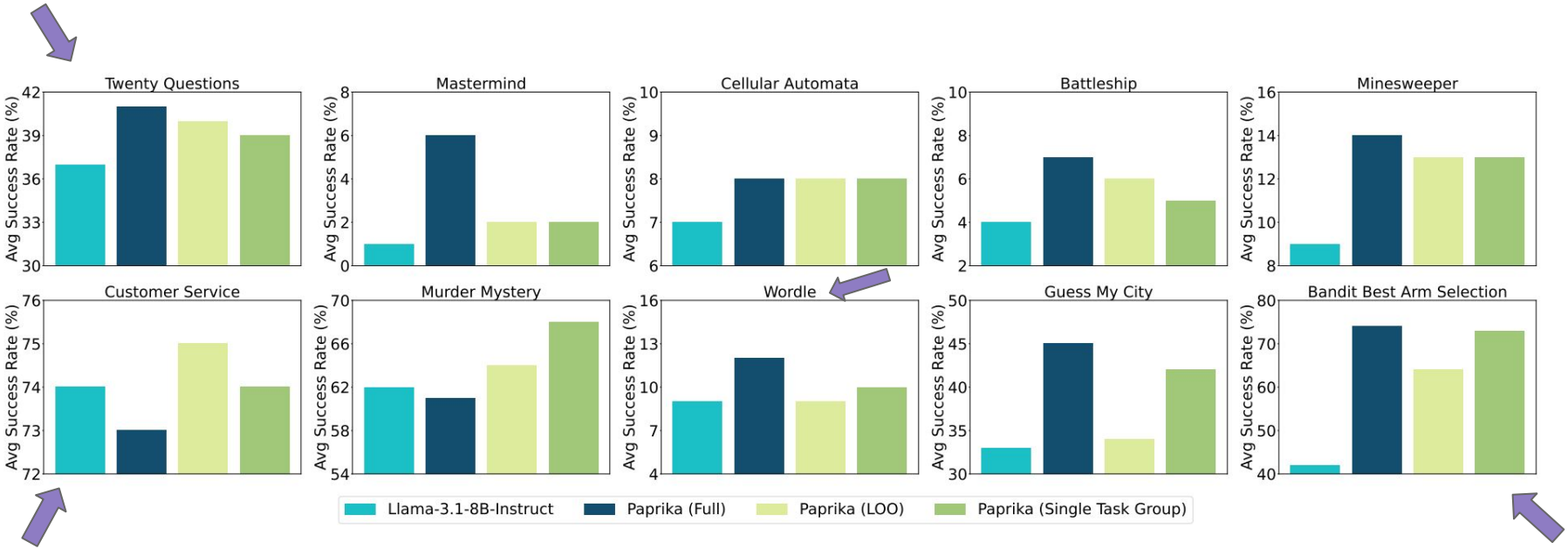
...

LOO Experiment 10: Train on Groups 1-9 → Test on Group 10

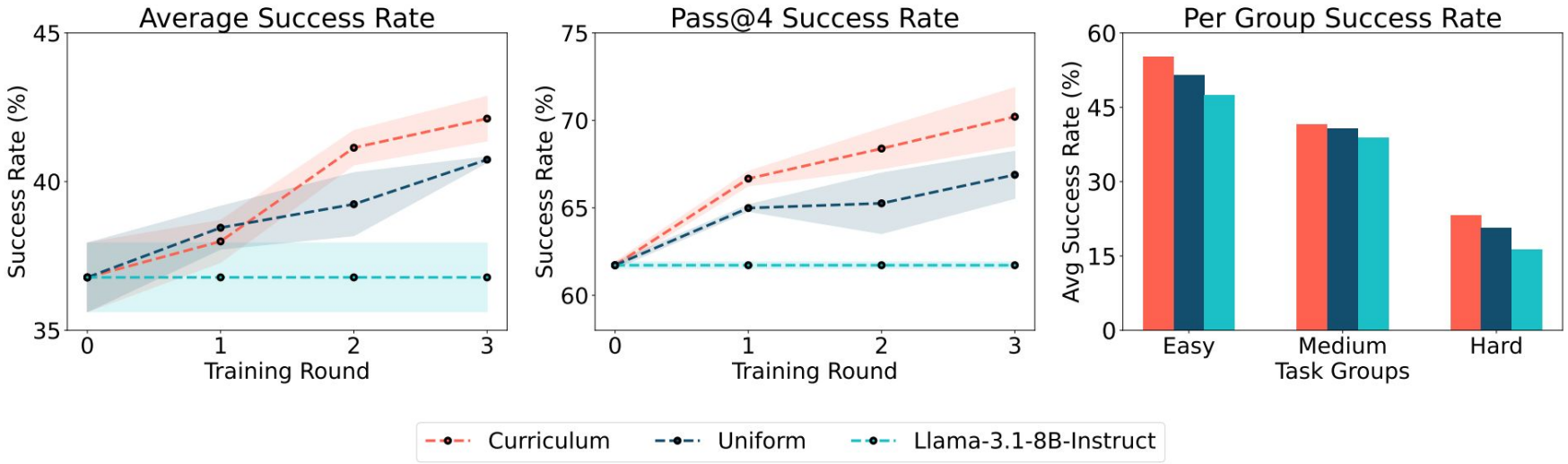
✓ Result: Improves on 9 out of 10 Groups!

This proves PAPRIKA learns **general exploration strategies**, not task-specific memorization.

Generalization Result: Leave-One-Out



Curriculum Learning



Related Work

- LLM Alignment
- In-Context Reinforcement Learning
- Curriculum Learning in RL

Limitations

- **Base Model Dependency:**
  - If base model is "too weak," it cannot generate the curriculum needed to make itself stronger
- **Offline Training**
  - Computational Constraints.

Limitations

- **Human-In-The-Loop**
  - Environment design still requires significant human effort despite GPT-4 assistance.
- **Curriculum Sensitivity:**
  - Effectiveness depends heavily on the quality of task group clustering (Very easy + Very Hard **X**) .

Future Work

- Online RL
- Automated Task Generation using LLM
- Improved Curriculum Learning

Conclusion

- **PAPRIKA Framework:**
  - In-Context RL via Self Play.
- **Zero-Shot Generalization**
  - Solves unseen tasks without training.
- **Curriculum Efficiency**
  - UCB maximizes high-potential learning.