RAGEN with A*-PO: Optimizing Multi-Turn Reasoning and Self-Evolving LLM Agents

INFO 7375 - FALL 2025

Presented by:
Anvitha Hiriadka 002472965
Nikhil Pandey 002775062
Vrinda Shinde 002290028
Ahsan Zafar Syed 002801441
Praneeth Reddy 002089375

Background: What is RAGEN?

Problem: LLMs as agents face challenges in long-horizon decision-making and stochastic feedback.

RAGEN Framework:

- A modular RL system for LLMs to simulate multi-turn reasoning and self-evolution.
- Based on StarPO — a trajectory-level optimization algorithm.
- Supports diverse simulated environments (e.g., WebShop, ToolBench, TextWorld).

Key Concepts:

- Echo Trap: gradient spikes due to reward cliffs → instability.
- StarPO-S: adds filtering, critic networks, and gradient stabilization.

Transition line: "Our work replaces StarPO with A*-PO to see if we can further stabilize learning and improve policy convergence."

A*-PO: The New Addition

What is A*-PO?

- A two-stage policy optimization technique enabling the efficient training of LLM for reasoning tasks.
- **Stage 1:** Estimate the optimal value function ***V**** using a reference model.
- **Stage 2:** Perform on-policy update using a simple least-squares regression loss with only a single generation per prompt.
- Improves **efficiency, training time**, **stability** and memory consumption compared to PPO/GRPO.

Key Formula:

- **Stage 1: Offline Optimal Value Estimation**

$$\hat{V}^*(x) = \beta \ln \left(\frac{1}{N} \sum_{i=1}^{N} \exp(r(x, y_i)/\beta) \right)$$

- **Stage 2: Online Policy Update**

$$\ell_t(\pi) := \mathbb{E}_{x, y \sim \pi_t(\cdot|x)} \left[\left(\beta \ln \frac{\pi(y|x)}{\pi_{ref}(y|x)} - (r(x, y) - \hat{V}^*(x)) \right)^2 \right]$$

About the WebShop Benchmark

Realistic **language-based RL environment** simulating online shopping.

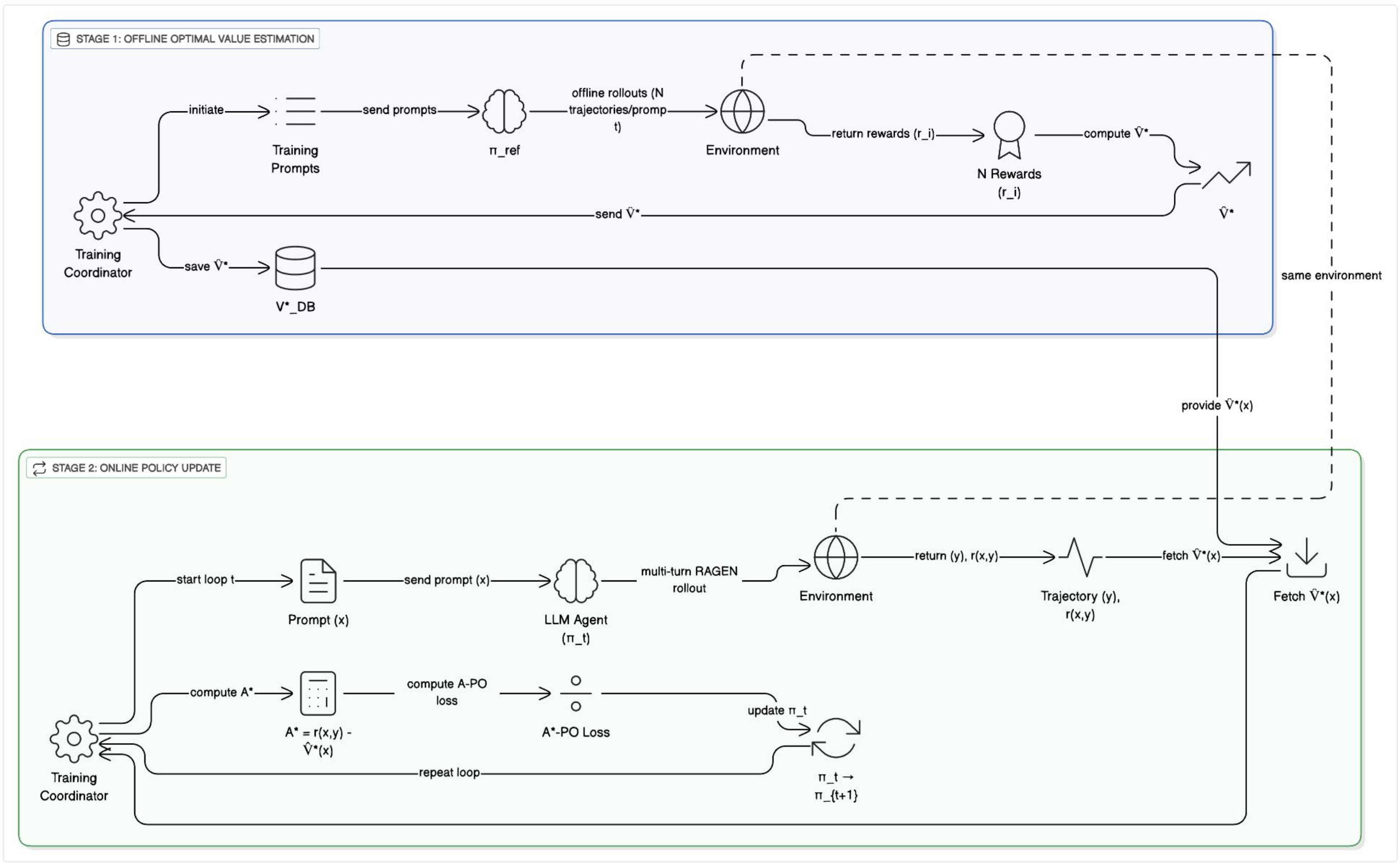
Each episode: an instruction like *"Find a blue running shoe under $50".*

Agent actions: search, click, filter, buy.

Rewards:

- 1.0: Successfully bought correct item
- 0.5-0.8: Bought item but wrong attributes
- 0.3-0.5: Made progress (searched, clicked valid products)
- 0.1-0.3: Valid action format but no progress
- 0.0: Invalid actions or no

System Architecture: RAGEN + A*-PO



Implementation Details

Tools and Set up:

Environment : WebShop
Reference Model : Qwen/Qwen2.5-3B

Policy Model : Qwen/Qwen2.5-3B
Optimizer : A*-PO

Training Parameters :
$\beta = 0.3$
KL coefficient = 0.03
Learning rate = 3e-7 → stability for reasoning updates

These parameters were chosen for efficiency and stability.

System Performance on WebShop Benchmark

Metric	RAGEN (A*-PO, ours)
Success Rate (%)	**0**
Loss	**1.77 - (-0.49)**
Average Reward	**0.40**
Training Steps to Converge	**100**
Evaluation Step	**50**

```
[2025-11-03 17:03:05] Step 70: Loss=-0.0099, Reward=0.444, Success=0.00%
  Computing V* for 1 prompts (1 samples each)...
Epoch 0:  71%|███████   | 71/100 [18:59<07:26, 15.39s/it, loss=-0.0099, reward=0.444, success=0.00%, step=70]Epoch 0:  71%|███████   | 71/100 [19:13<07:26, 15.39s/it, loss=-1.2003, reward=0.420, success=0.00%, step=71]
  Computing V* for 1 prompts (1 samples each)...
Epoch 0:  72%|███████▏  | 72/100 [19:13<06:59, 14.98s/it, loss=-1.2003, reward=0.420, success=0.00%, step=71]Epoch 0:  72%|███████▏  | 72/100 [19:25<06:59, 14.98s/it, loss=-0.7539, reward=0.425, success=0.00%, step=72]
  Computing V* for 1 prompts (1 samples each)...
Epoch 0:  73%|███████▎  | 73/100 [19:25<06:19, 14.06s/it, loss=-0.7539, reward=0.425, success=0.00%, step=72]Epoch 0:  73%|███████▎  | 73/100 [19:38<06:19, 14.06s/it, loss=-0.6260, reward=0.433, success=0.00%, step=73]
Epoch 0:  74%|███████▍  | 74/100 [19:38<05:54, 13.63s/it, loss=-0.6260, reward=0.433, success=0.00%, step=73]⠿ Running (1/1 containers active)... View app at ht
Step 75: Loss=-0.2704, Reward=0.450, Success=0.00%
Epoch 0:  74%|███████▍  | 74/100 [19:58<05:54, 13.63s/it, loss=-0.5371, reward=0.437, success=0.00%, step=74]
  Computing V* for 1 prompts (1 samples each)...
  ✓ V* cache: 1/1 hits (100.0%)
Epoch 0:  75%|███████▌  | 75/100 [19:58<06:27, 15.48s/it, loss=-0.5371, reward=0.437, success=0.00%, step=74]Epoch 0:  75%|███████▌  | 75/100 [20:10<06:27, 15.48s/it, loss=-0.4101, reward=0.442, success=0.00%, step=75]
[2025-11-03 17:04:16] Step 75: Loss=-0.4101, Reward=0.442, Success=0.00%
⠿ Running (1/1 containers active)... View app at https://modal.com/apps/nikhilp0799/main/ap-PYIy2ER5un3AfcnJfqdrYi
```

Failure Cases

Example	What Happened	Likely Cause
Unable to arrive at the Success Rate	The agent is able to fetch the valid reward, but fails to identify the appropriate the final step.	Evaluate Policy unable to find the right action

Sample Example from training

Prompt Format Violation & Reasoning Hallucination:
The agent is behaving like a chatbot, not an autonomous agent.

Task: Find me machine wash women's tops, tees & blouses with short sleev…

Extracted Action: search[casual shirts]

Agent's Full Output (The "Reasoning"):

"To complete this task using the given guidelines and provided information about webshops A to D:

1. Search for "casual shirts":
    * Navigate through each website step by step.
    * For Website A [Amazon], visit their homepage...

2. ...wait patiently while package gets shipped out via express courier service... check email notification..."
Result: Reward: 0.0 (Trajectory eventually failed)

Conclusion

1. We Successfully Integrated RAGEN + A*-PO
We built a novel agent training system from scratch, combining RAGEN's multi-turn, reasoning-based rollouts with A*-PO's two-stage, critic-free optimizer.

2. Performance is Promising, but Gated by V* Quality
Our results on WebShop show the agent successfully learned complex, multi-step tasks. However, performance is critically dependent on the quality of the offline V* (optimal value) estimate from Stage 1.

Future Work & Next Steps:

1. **Iterative V* Refinement:** Instead of a single offline step, we would re-calculate the V* values halfway through training using the new, smarter policy, giving the agent a more accurate target to aim for.

2. **Adaptive Sampling (StarPO-S Idea):** We would integrate the "trajectory filtering" idea from the RAGEN paper into Stage 1. By focusing our N offline samples on high-variance, uncertain prompts, we could build a more robust V* database from the start.

References

- RAGEN: Understanding Self-Evolution in LLM Agents via Multi-Turn Reinforcement Learning
- WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents
- Accelerating RL for LLM Reasoning with Optimal Advantage Regression

Thank you