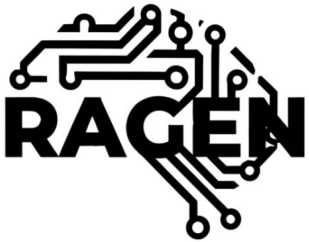


RAGEN:
Training Agents by Reinforcing Reasoning

Zihan Wang*, Kangrui Wang*, Qineng Wang*, Pingyue Zhang*, Linjie Li*,
Zhengyuan Yang, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Monica Lam,
Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, Manling Li

Introduction: LLM Agents & Multi-Turn RL

Challenge: Training Large Language Models (LLMs) as autonomous agents

- Requires sequential decision-making, memory across turns, adaptation to stochastic feedback.
- Applications: Planning assistants, robotics, tutoring agents.
- Goal: Self-improvement through experience.

Problem: Existing RL methods for LLMs often struggle in multi-turn, stochastic settings.

- Instability, complex reward signals, limited generalization.

Core Question: How to design stable and effective self-evolving LLM agents?

Framework: StarPO

State-Thinking-Actions-Reward Policy Optimization (StarPO)

- **Goal:** General framework for trajectory-level agent RL.
- **Key Idea:** Treat the entire multi-turn interaction trajectory (observations, reasoning, actions, rewards) as a single coherent unit for optimization.

Key Innovation: Focuses on optimizing complete interaction sequences rather than individual actions.

Objective Function:

$$J_{\text{StarPO}}(\theta) = E_{M,\tau \sim \pi_\theta}[R(\tau)]$$

where τ is a full trajectory and $R(\tau)$ is the cumulative reward over the trajectory.

StarPO: Optimization Procedure

1 **Rollout Generation**

Agent generates multiple trajectories (N) starting from initial states (P).

2 **Structured Output**

Each action a_t^T includes reasoning and executable action a_t:

$$a_t^T = \langle \text{think} \rangle \ \ldots \ \langle \text{answer} \rangle a_t \langle \text{answer} \rangle$$

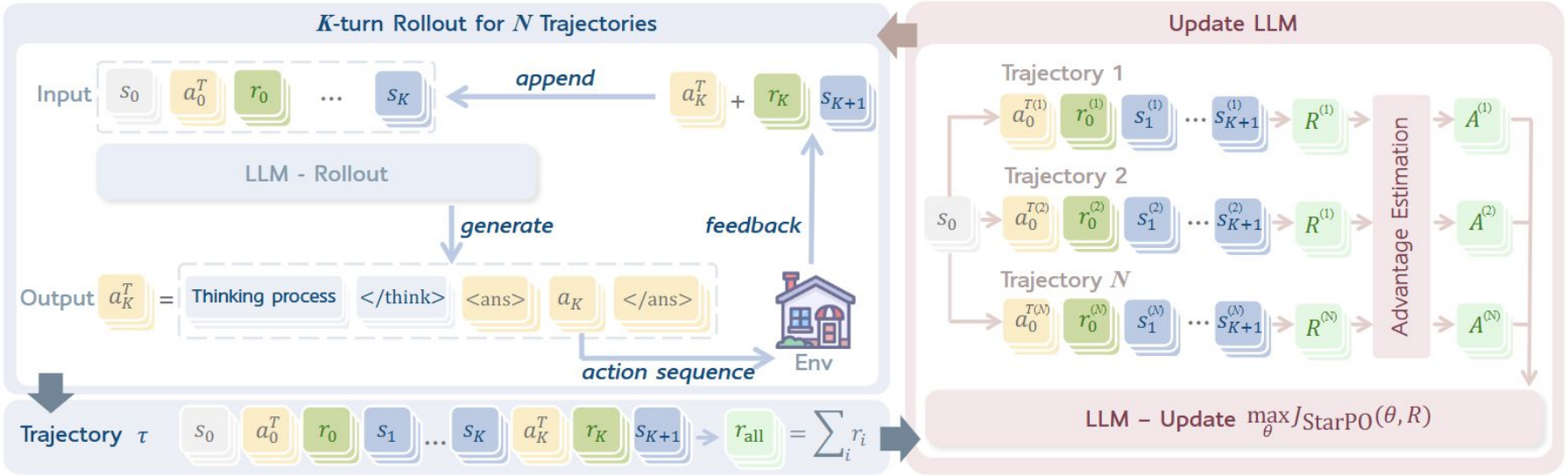
3 **Trajectory**

Full sequence of interactions $\tau = \{s_0, a_0^T, r_0, s_1, \ \ldots, s_K\}$.

4 **Interleaving**

Rollouts and updates are performed iteratively.

StarPO: Optimization Procedure



System: RAGEN

Structured Rollouts

Supports structured output sequences with reasoning and actions

Customizable Rewards

Flexible reward function design for different task types

Multi-turn Environments

Integration with stochastic and deterministic environments

Research Focus

Designed for stability, generalization, and learning dynamics analysis

Functionality

Execution backend for StarPO, platform for research and analysis

Experiment Setup: Environments & Tasks

Bandit

Single-turn, Stochastic

Risk-sensitive symbolic reasoning

Sokoban

Multi-turn, Deterministic

Irreversible multi-step planning

Frozen Lake

Multi-turn, Stochastic

Planning combined with uncertainty

Characteristics

Minimal, controllable, stripped of real-world priors.

Purpose

Clean analysis of reasoning emergence and learning dynamics.

Experiment Setup: Training & Evaluation

🧠 Model & Framework

- **Model:** Qwen-2.5 (0.5B) LLM
- **Framework:** StarPO (PPO & GRPO variants)

💻 Hardware & Iterations

- **Hardware:** H100 GPUs
- **Iterations:** 200 rollout-update cycles

🔄 Batch & Trajectory

- **Batch:** $P = 8$ prompts, $N = 16$ rollouts/prompt
- **Trajectory:** Max 5 turns, 10 actions/turn

Finding 1: Instability in Multi-Turn RL

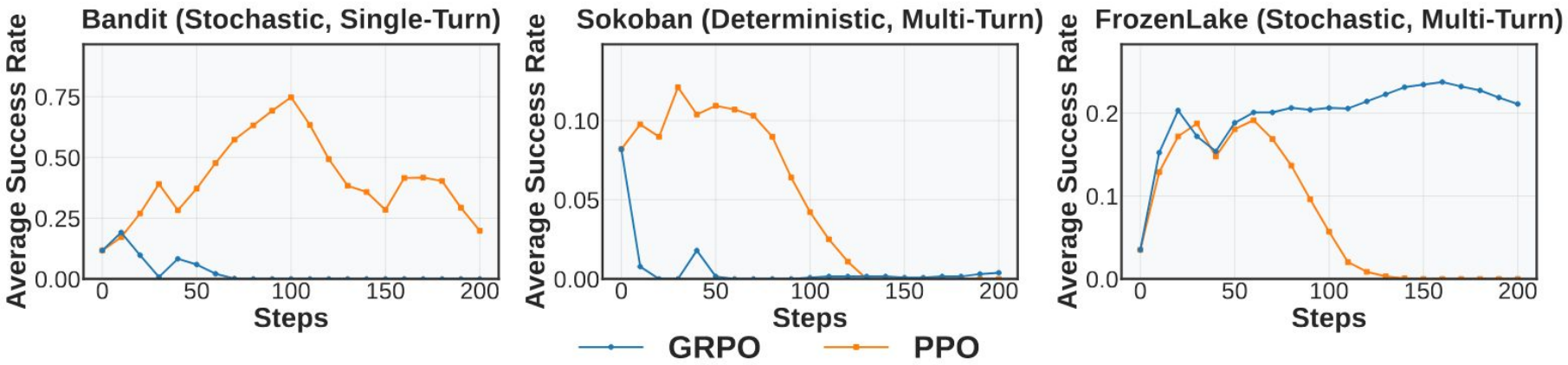


Figure 5 | **Baseline StarPO performance across environments.** All StarPO runs initially improve but eventually collapse in multi-turn agent settings. PPO variant shows better training stability compared to the GRPO variant, especially in Bandit and Sokoban, indicating that critic-based methods better resist early-stage degradation under long-horizon dynamics.

Finding 1: Instability in Multi-Turn RL

📊 Performance Pattern

- Different from single-turn RL tasks.

⚖️ Comparison

- PPO variant generally shows better stability than GRPO in Bandit and Sokoban.
- Critic (PPO) seems to help resist early degradation.
- GRPO on Frozen Lake surprisingly stable.

💡 Conclusion

Direct adaptation of single-turn RL methods (PPO/GRPO) to multi-turn agent settings is insufficient.

Finding 2: The "Echo Trap" Failure Mode

Step	Model Output
0	`Dragon is often associated with patience, strength, and reliability, while Phoenix is associated with speed, agility, and creativity.`
0	`The symbolic meaning of Dragon and Phoenix's names suggests that they may have different reward distributions... Therefore, I think Phoenix is more likely to give higher rewards on average.`
150	`My initial intention is to pick arm Dragon. My first action is to go for arm Dragon.`
150	`My initial intention is to pick arm Dragon, so my first action is to go for arm Dragon.`
150	`My initial intention is to choose Arm Dragon. After analyzing both arms, my intention was to choose Arm Dragon.`

Table 1 | Examples of reasoning patterns in the Bandit task. Top rows show diverse reasoning from model before training, while bottom rows show repetitive and collapsed reasoning after RL training.

Finding 2: The "Echo Trap" Failure Mode

Problem: Model collapse manifests as "Echo Trap".

Model Behavior Evolution

Early-stage:

Diverse reasoning (symbolic, exploratory).

Late-stage:

Repetitive, deterministic templates (e.g., "choosing Dragon" without justification).

- RL reinforces superficial patterns, suppressing exploration.
- Leads to trajectory diversity collapse and long-term performance degradation.

Finding 3: Collapse Indicators

Early Warning Signs: Metrics that signal impending performance degradation

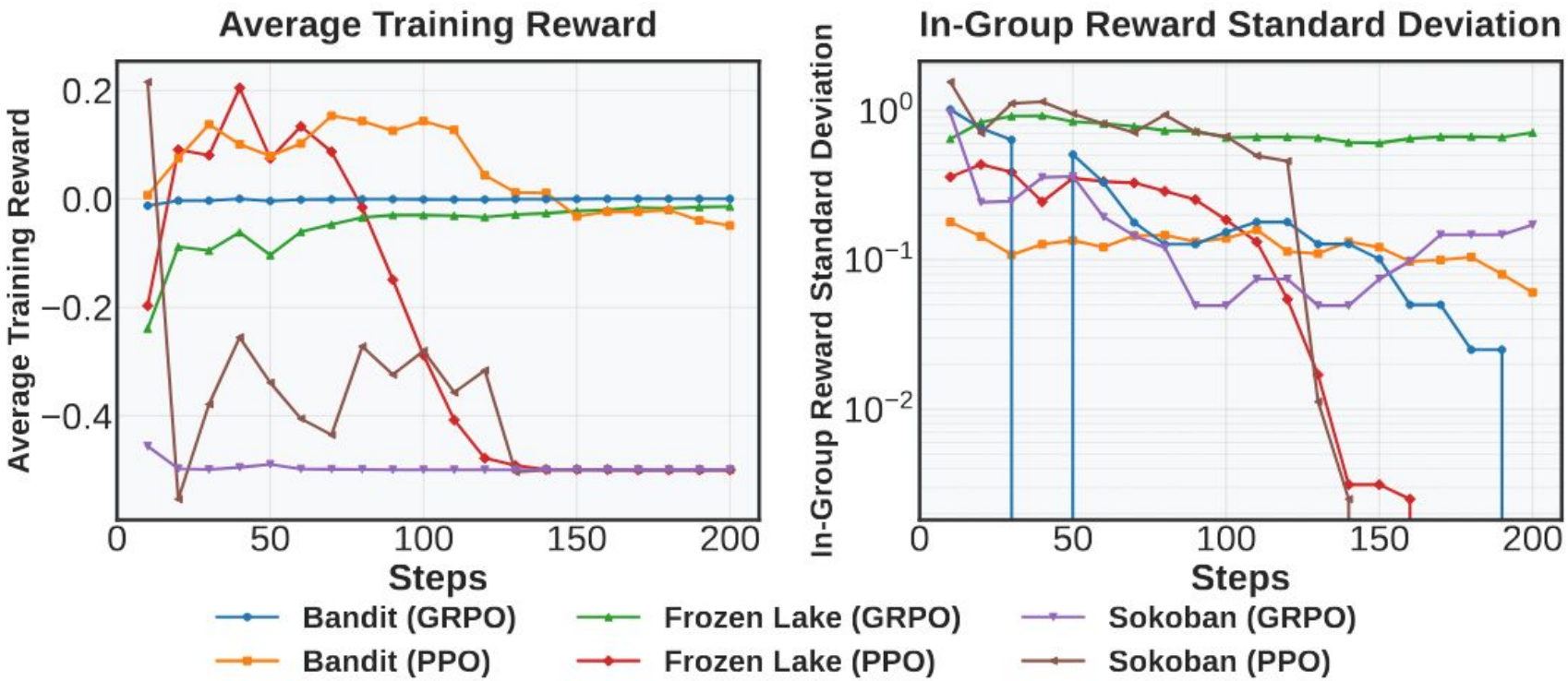
⚠ Early Warning Signs

- **Reward Standard Deviation:** Drops sharply before performance degrades (Fig 6).
- **Output Entropy:** Sharp drop indicates overconfidence, narrow reasoning paths.

❌ Confirmation Signs

- **Average Reward:** Plateaus or drops.
- **Gradient Norm:** Spikes indicate instability and irreversible collapse.

Finding 3: Collapse Indicators



Finding 3: Collapse Indicators

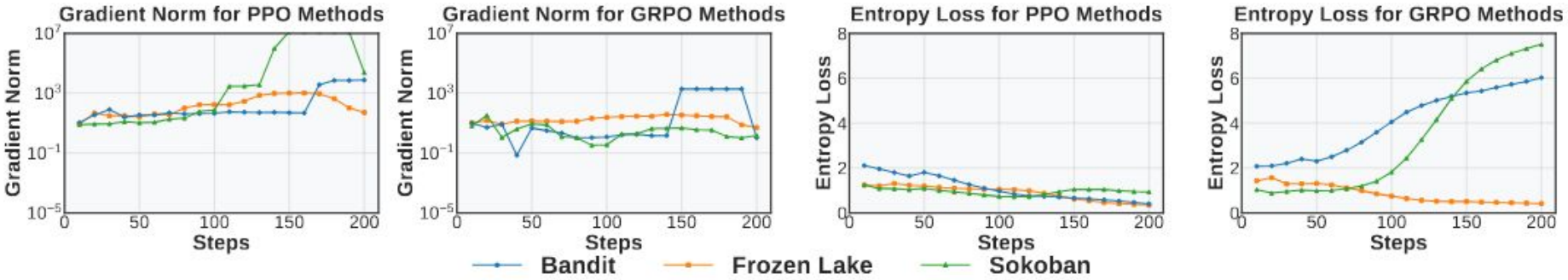


Figure 6 | **Collapse indicators and early warning signals in multi-turn RL.** Reward standard deviation and entropy (right-side plots) drop early, often before reward degrades, serving as early warning signals. Reward mean and gradient norm (left-side plots) reflect collapse directly—plateaus and spikes confirm performance and training instability.

StarPO-S: Stabilized StarPO

💡 Core Idea

Focus on training on more informative samples.

1 Trajectory Filtering

Retain only highly uncertain prompts (high reward variance).

2 KL Term Removal

Eliminate KL divergence penalty in PPO.

3 Decoupled Clipping (Asymmetric)

Use higher upper bound for PPO clipping.

Finding 4: Uncertainty Filtering Improves Stability

Method: Sort prompts by reward variance and retain only the most uncertain ones.

Uncertainty Filtering

Sort prompts by reward variance (Std) from rollouts. Retain top $p\%$ most uncertain prompts.

$$U(\pi_\theta, M, s_0) = \mathrm{Std}_{\tau \sim \pi_\theta(\cdot \mid s_0)}[R(\tau)]$$

Effect (Fig 7)

- Filtering low-variance rollouts significantly delays or eliminates collapse (esp. PPO).
- Improves training efficiency (fewer updates needed).
- Aggressive filtering (e.g., 25% retention) is effective.

StarPO-S: Stabilized StarPO

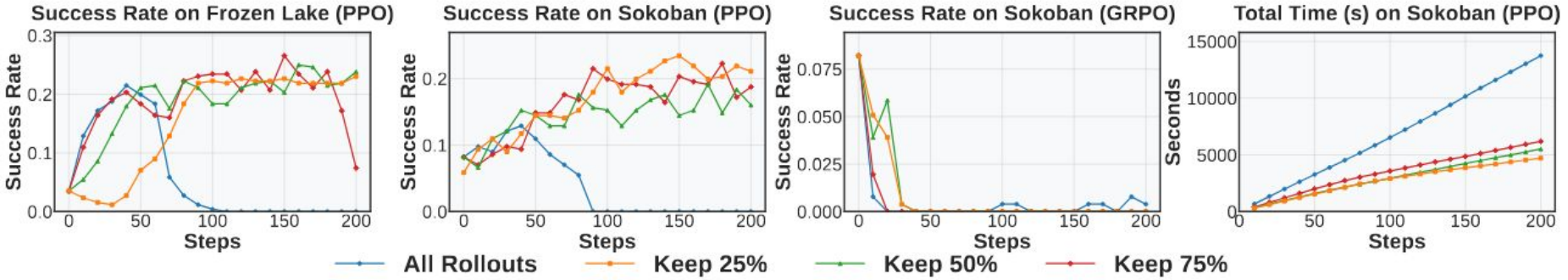


Figure 7 | **Effect of uncertainty-based filtering on multi-turn RL stability.** Filtering out low-variance trajectories reduces collapse risk and improves success rate. On PPO variants, collapse is largely mitigated when more than half of the trajectories are filtered.

Finding 5: KL Removal & Asymmetric Clipping

Methods (from DAPO, adapted for multi-turn):

KL Removal

Eliminate KL divergence penalty in PPO objective.

Clip-Higher (Asymmetric)

Decouple PPO clipping bounds.

$$\epsilon_{high} = 0.28, \epsilon_{low} = 0.2$$

Note: Higher upper bound for better exploration

Effect (Fig 8)

- Both methods improve peak success rate and delay collapse.
- Allows more aggressive learning from high-reward trajectories.
- More flexible gradient shaping.

StarPO-S: Stabilized StarPO

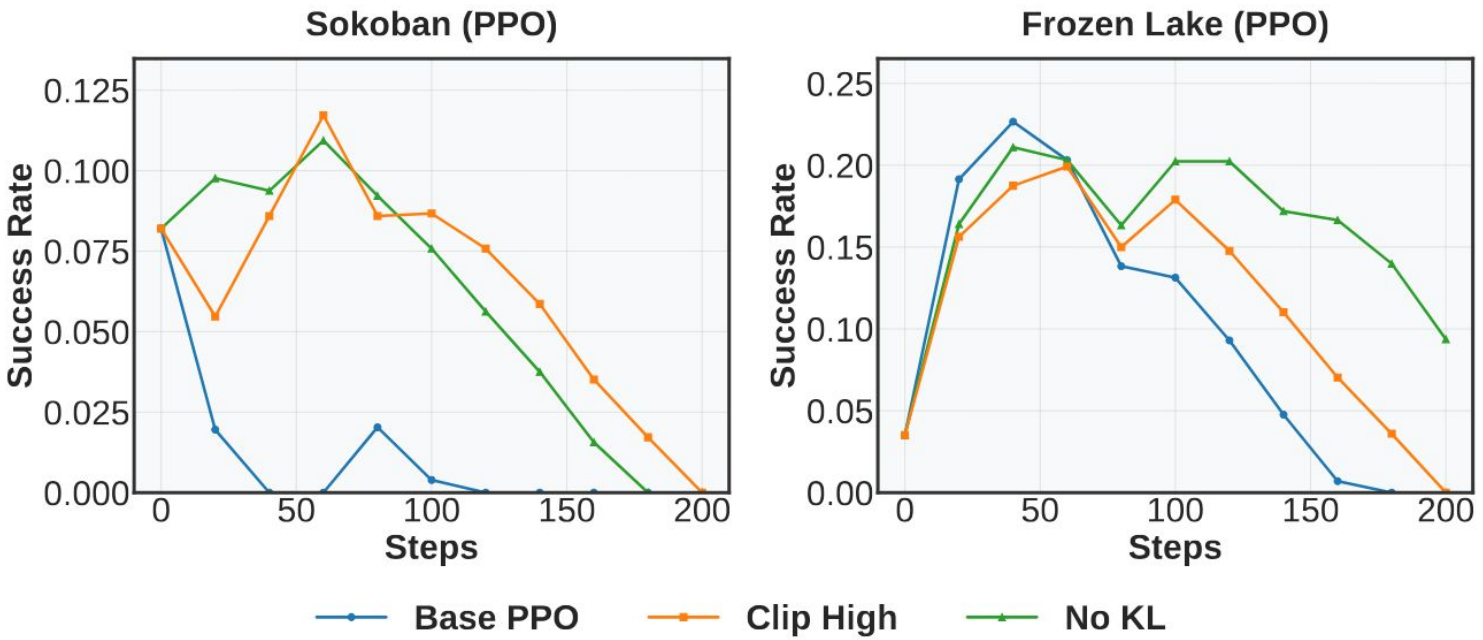


Figure 8 | **Effect of KL removal and asymmetric clipping on PPO stability.** Removing KL constraints and enabling stronger positive gradient flow both improve peak performance and delay collapse in multi-turn RL.

StarPO-S: Stabilized StarPO

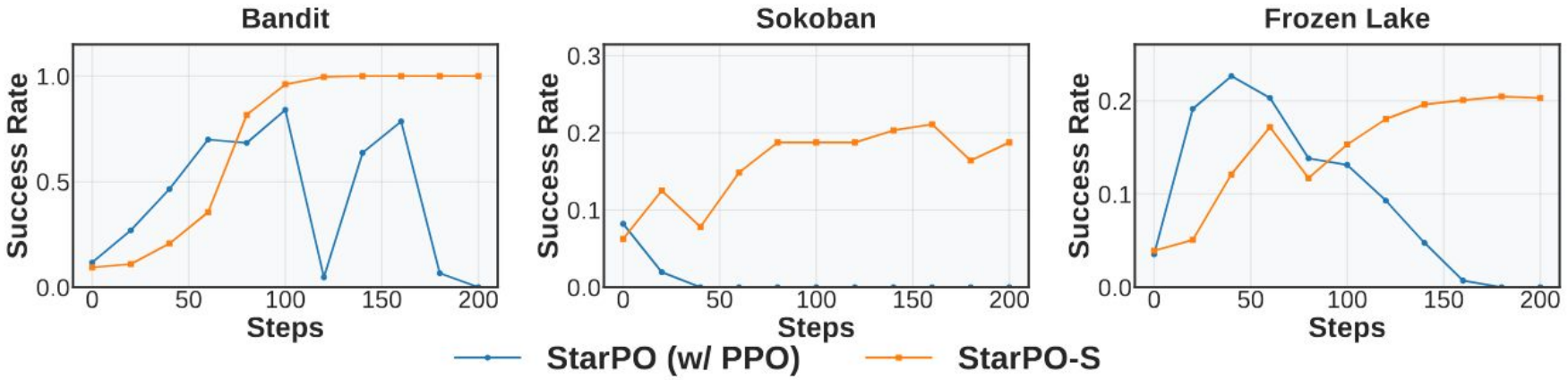


Figure 9 | **StarPO-S improves stability and final performance across tasks.** Compared to vanilla StarPO, StarPO-S achieves higher success rates and reliefs collapse in all three tasks.

Finding 6: Rollout Quality Matters

Key Dimensions: Task diversity, interaction granularity, rollout frequency

Key Dimensions of Rollout Quality: Task Diversity | Interaction Granularity | Rollout Frequency

Task Diversity (Table 2)

Higher diversity:
More distinct prompts improves generalization

Optimal configuration:
Fewer responses per prompt (e.g., 4 per prompt)

Minimum requirement:
Multiple responses per prompt to enable contrast

Interaction Granularity (Table 3)

Optimal actions per turn:
5-6 actions enables better planning

Too many actions:
7+ actions degrades performance

Reason:
Avoids noise from overly long rollouts

Finding 7: Reasoning Needs Fine-Grained Rewards

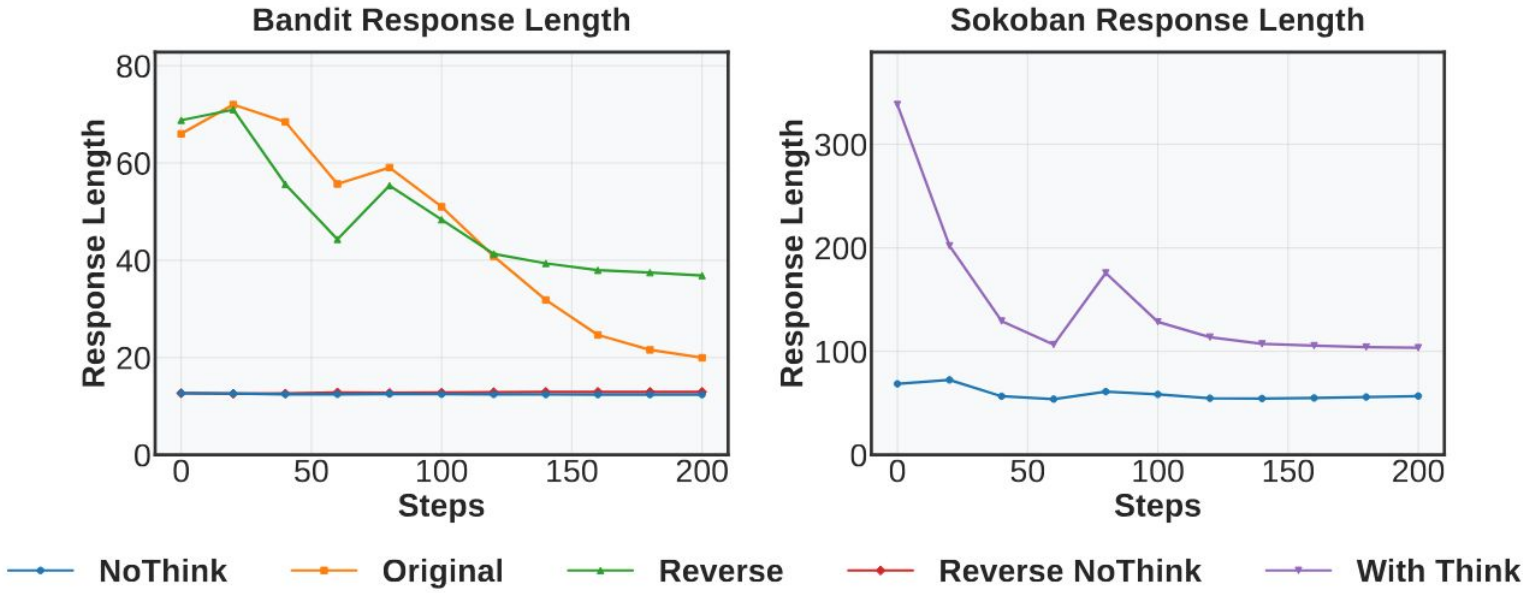


Figure 11 | **Reasoning length over training iterations across different tasks.** We track the average token count of reasoning segments (`<think>` blocks) during RL training. Across all environments, reasoning length declines as training progresses, with `BanditRev` maintaining longer traces—possibly due to greater semantic-reward conflict requiring more deliberation.

Finding 7: Reasoning Needs Fine-Grained Rewards

🤔 **Problem:** Simply encouraging reasoning doesn't guarantee actual reasoning

⚠️ Current Reward Limitations

- Models often regress to direct action selection if no reward advantage exists.
- Shallow strategies suffice in simple MDPs.

🎭 Hallucination

When rewards only reflect success, models produce fake reasoning ("hallucinated thoughts").

Conclusion

StarPO Framework

Proposed trajectory-level RL framework for LLM agents.

RAGEN System

Developed modular system for training and evaluation.

Key Challenges

Identified Echo Trap, instability, and rollout quality issues.

StarPO-S

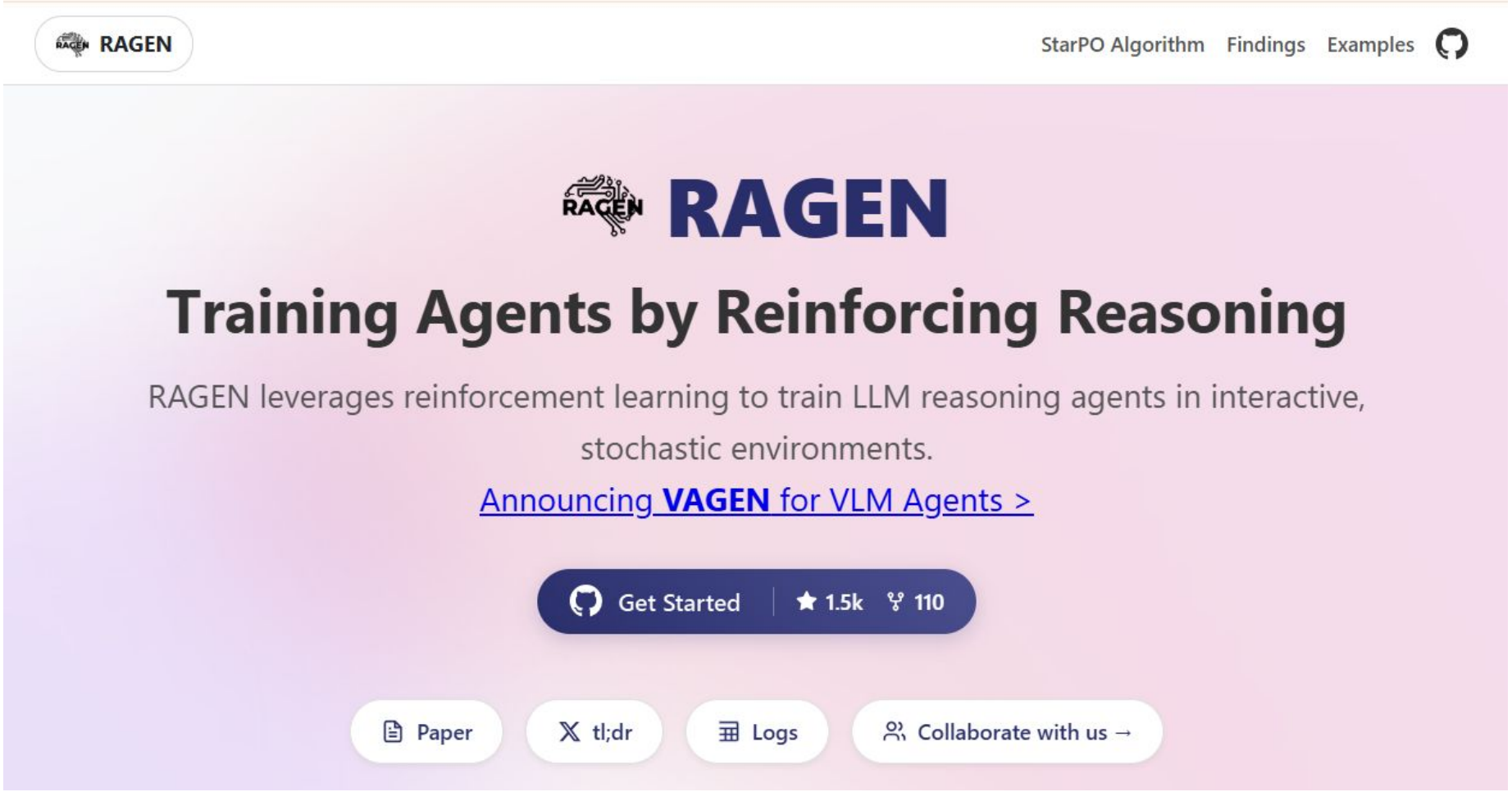
Stabilized StarPO with filtering, KL removal, and asymmetric clipping.

Rollout Quality

Highlighted importance of diversity, granularity, and freshness.

Reasoning Rewards

Emphasized need for fine-grained reward signals for reasoning.



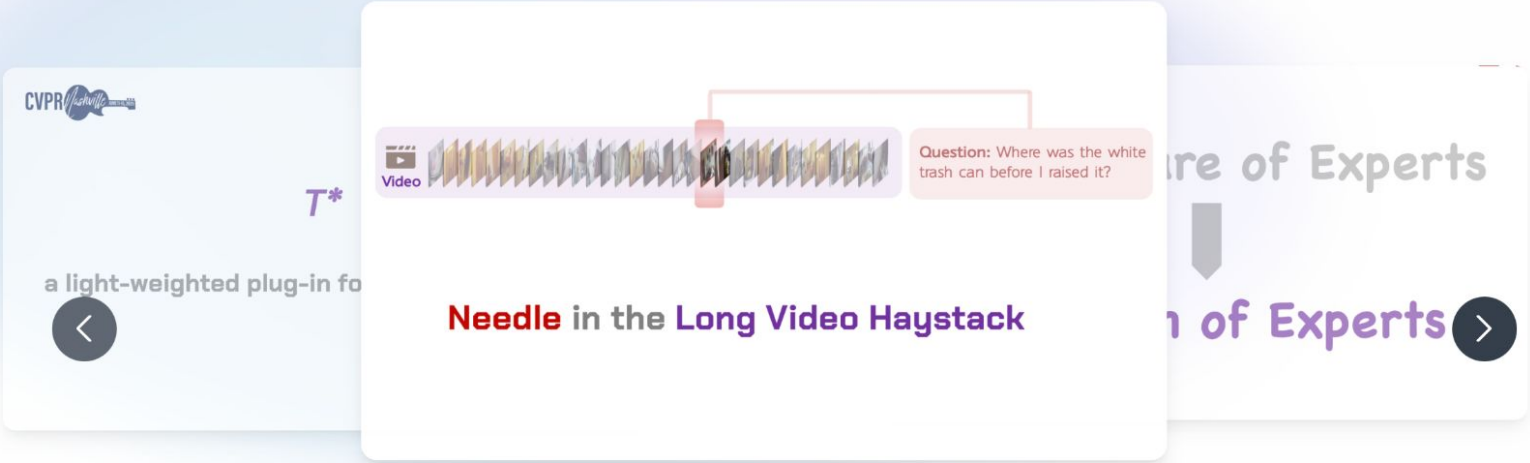
Check out more at https://ragen-ai.github.io!

MLL Lab
Machine Learning and Language

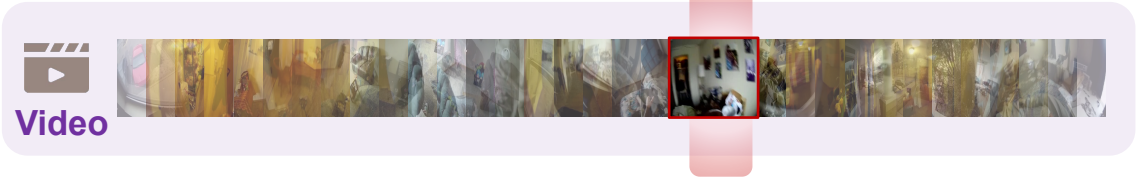
We develop intelligent language + X (vision, robotics, etc) models that reason, plan, and interact with the physical world.

Join Us

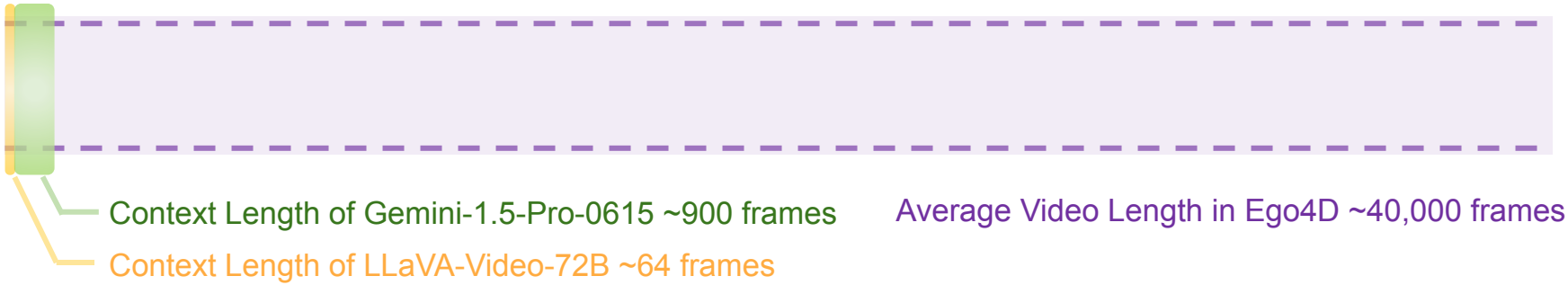
Announcing **RAGEN**: Training RL Agents - Github ⭐ 1.5k >



Long Video Haystack



Needle in the Long Video Haystack



Needle in the Long Video Haystack

Hugging Face

Datasets: LVHaystack / **LongVideoHaystack**

Haystack-Ego4D

988 videos

432 hours

15,092 QA pairs

23,800 frames

Haystack-LVBench

246 videos

57.7 hours

602 QA pairs

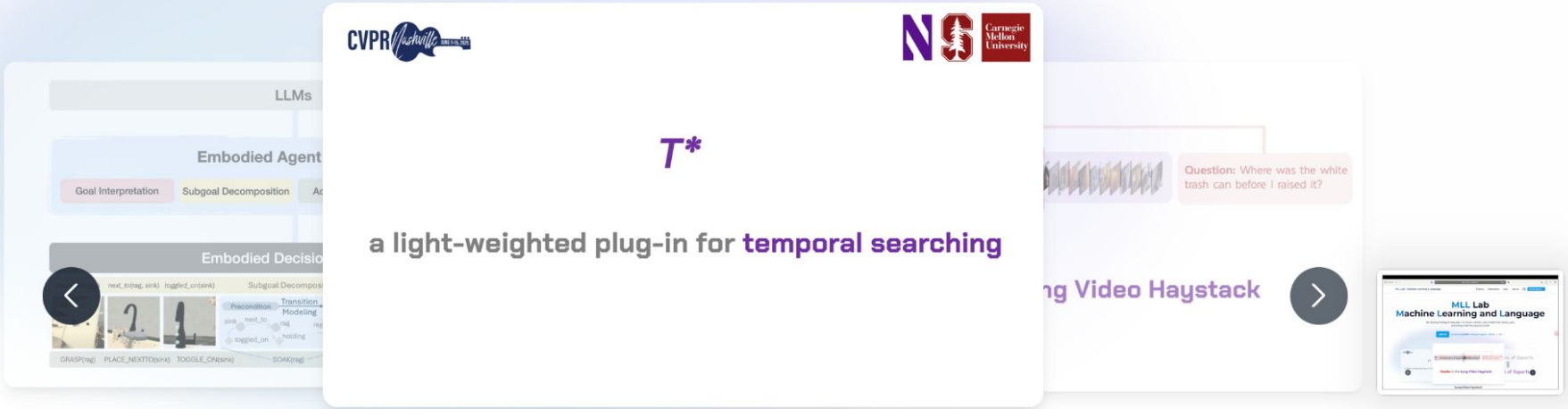
1,070 frames

MLL Lab
Machine Learning and Language

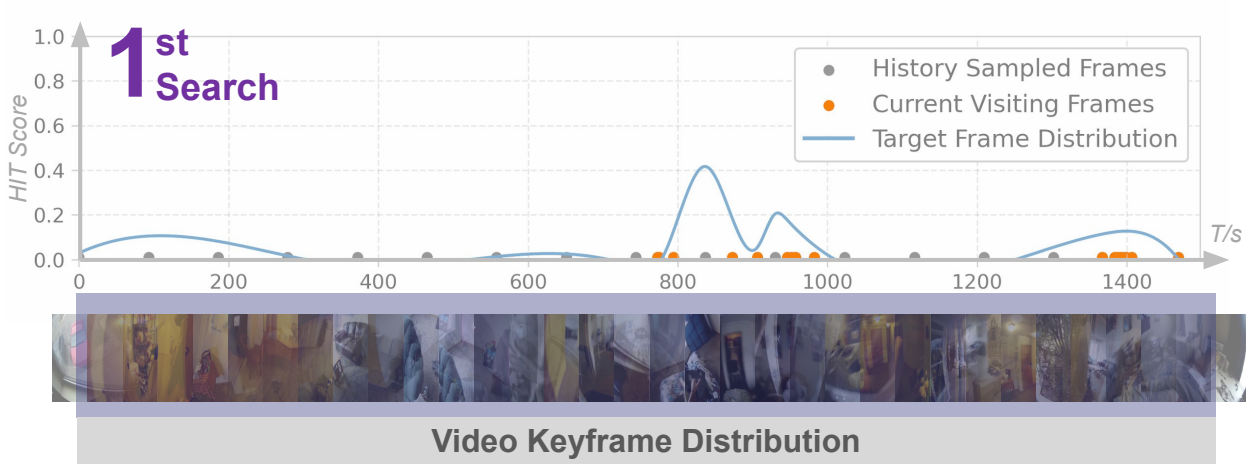
We develop intelligent language + X (vision, robotics, etc) models that reason, plan, and interact with the physical world.

Join Us

Announcing **RAGEN**: Training RL Agents - Github ⭐ 1.5k >



Temporal Search Plug-in

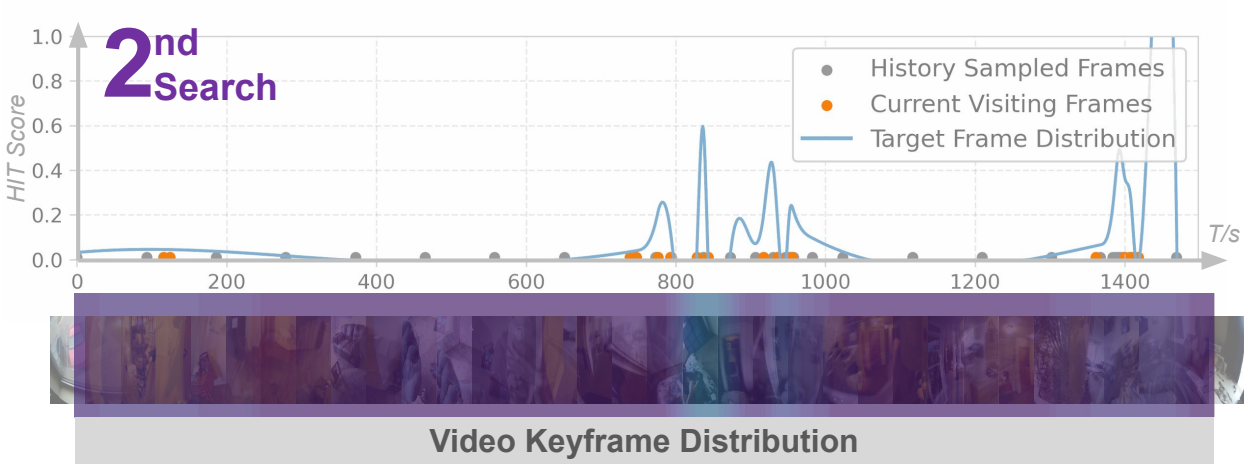


Video Keyframe Distribution

Frame Sampling

Distribution Updating

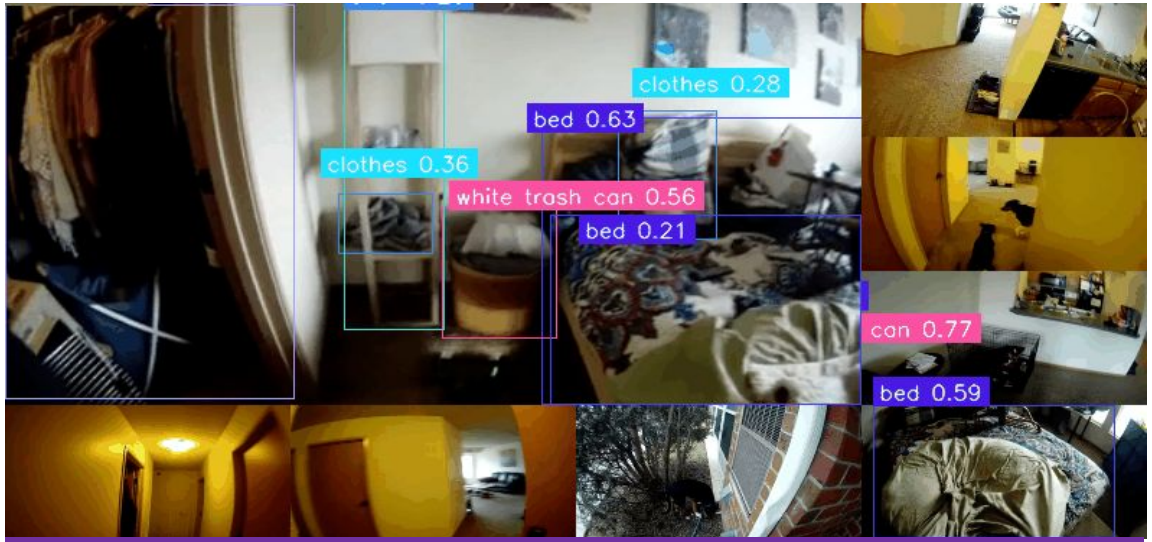
Sampled Frames



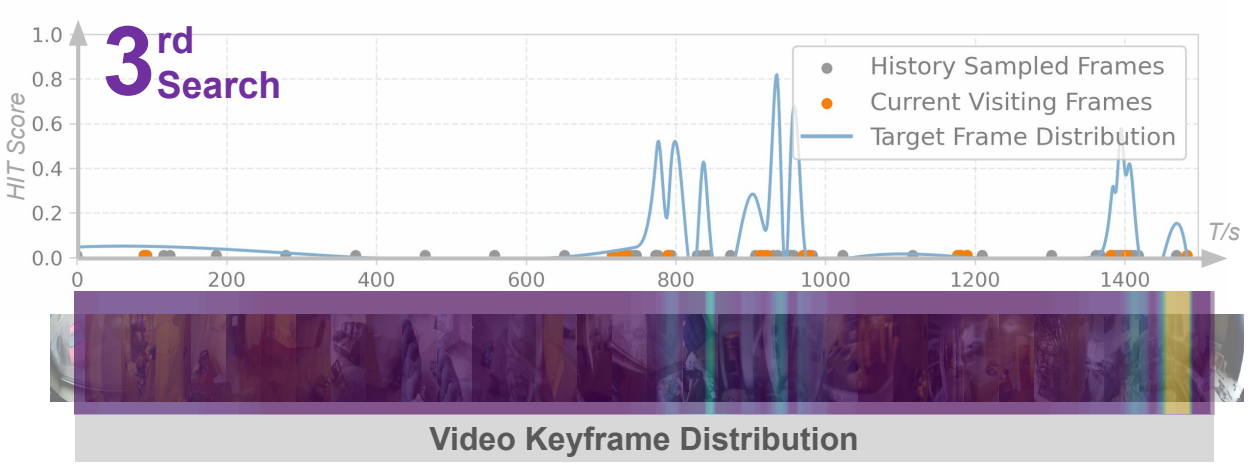
Video Keyframe Distribution

Frame Sampling

Distribution Updating



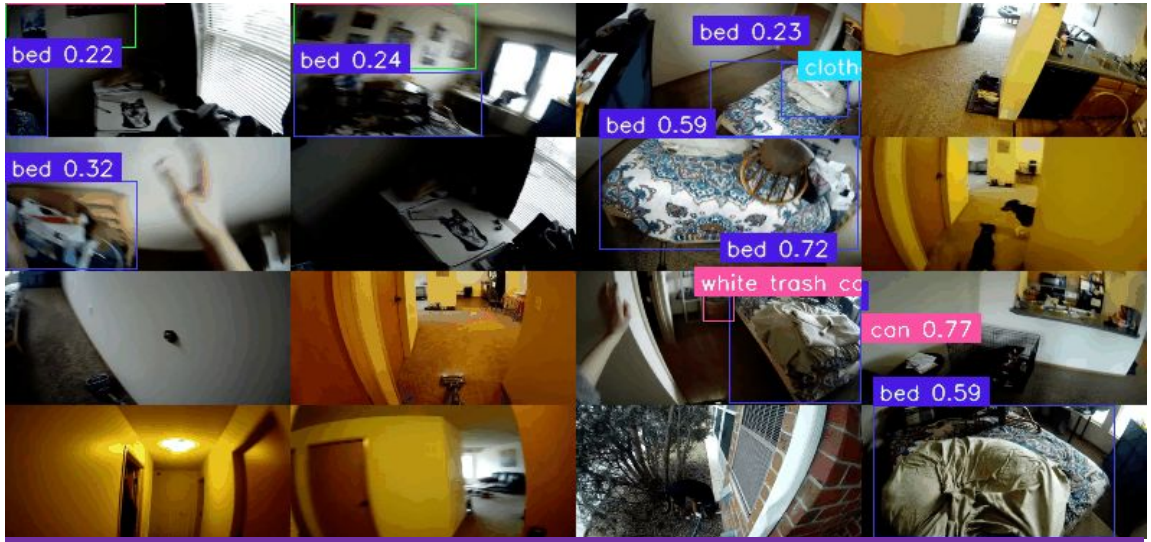
Sampled Frames



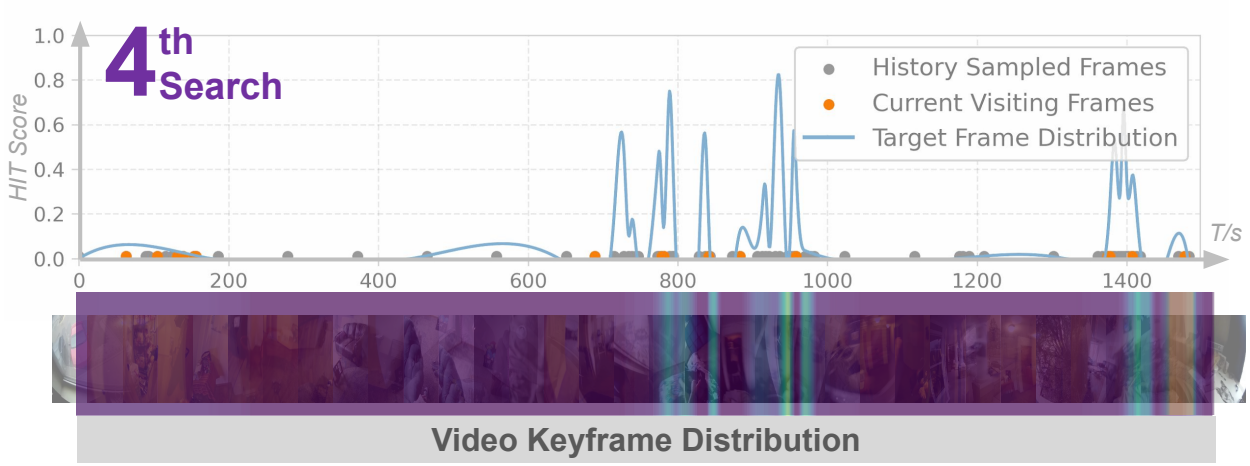
Video Keyframe Distribution

Frame Sampling

Distribution Updating



Sampled Frames

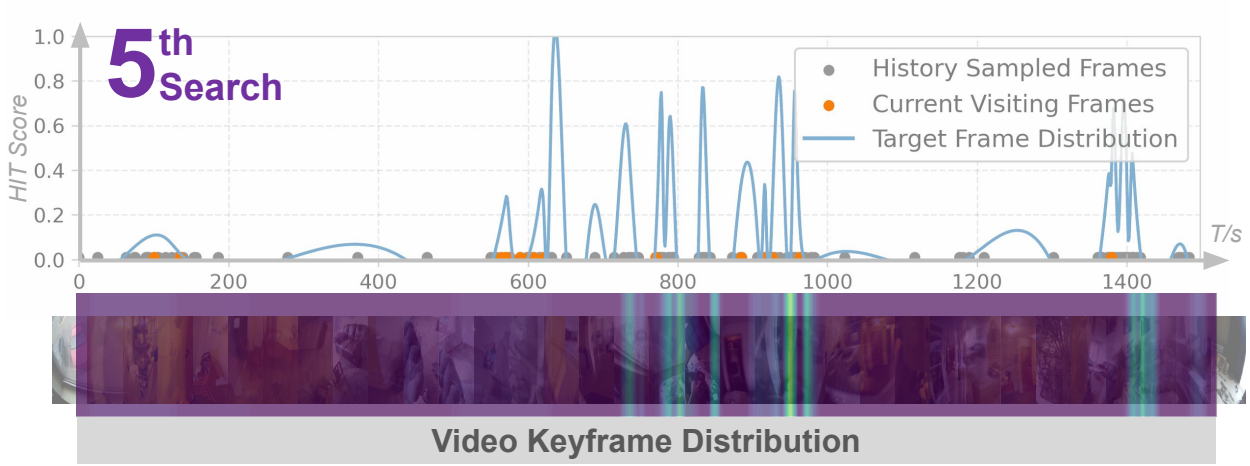


Video Keyframe Distribution

Frame Sampling

Distribution Updating

Sampled Frames

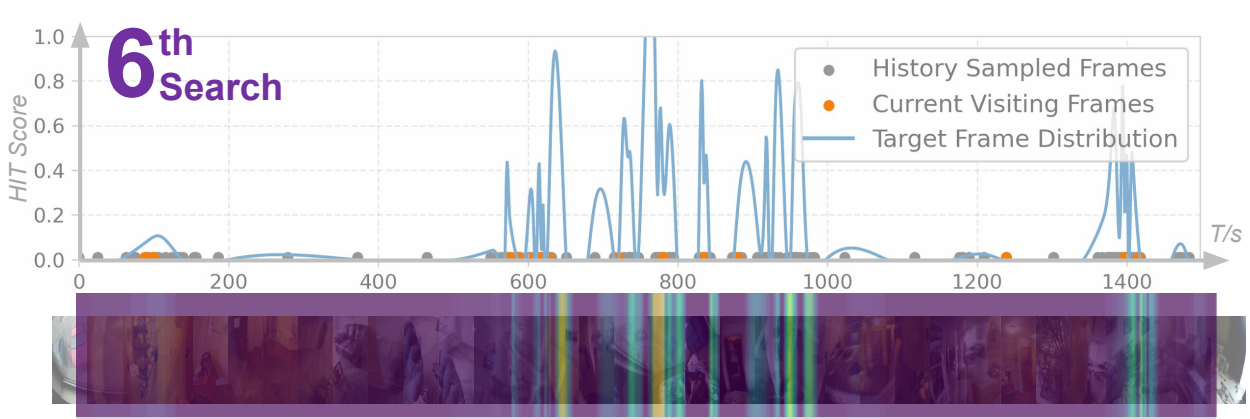


Video Keyframe Distribution

Frame Sampling

Distribution Updating

Sampled Frames



Video Keyframe Distribution

Frame Sampling

Distribution Updating

Sampled Frames

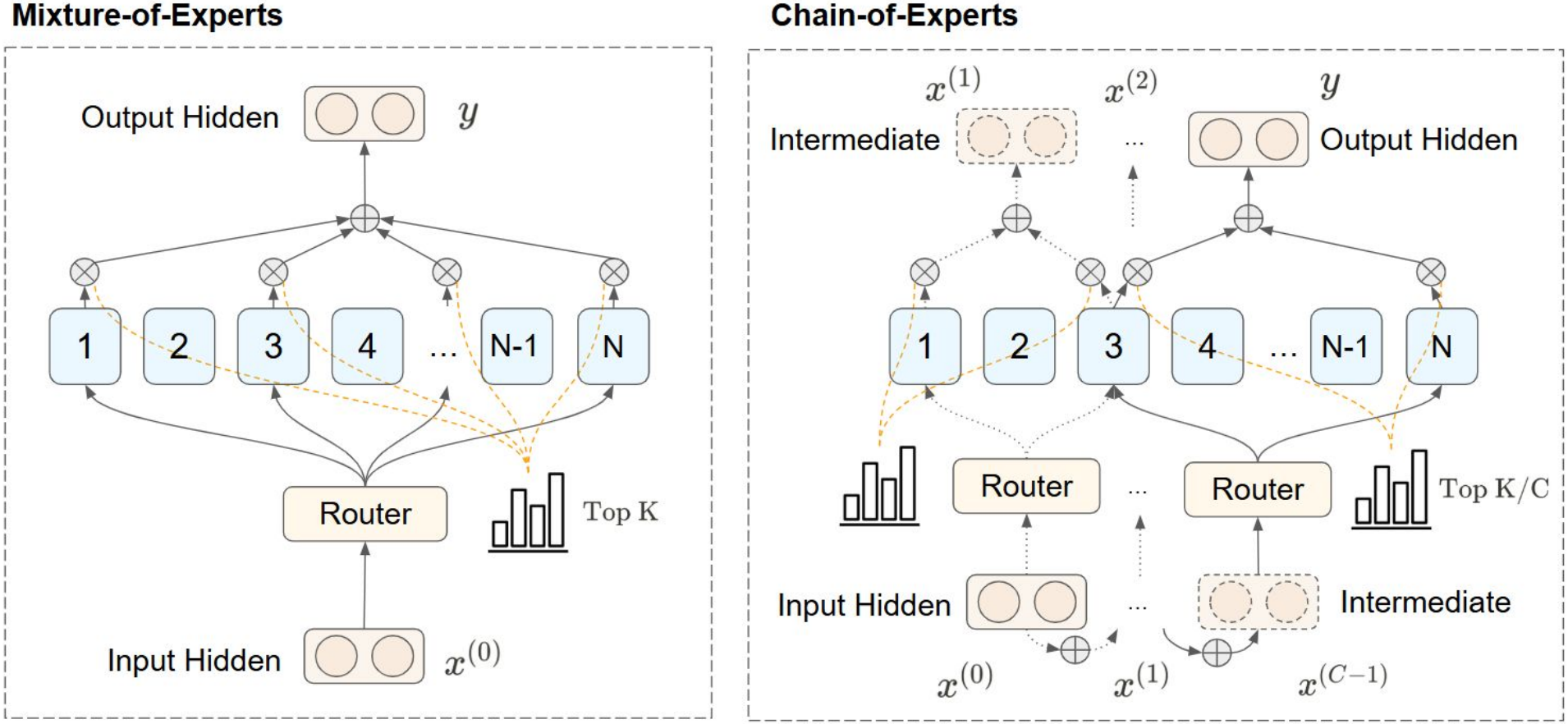
MLL Lab
Machine Learning and Language

We develop intelligent language + X (vision, robotics, etc) models that reason, plan, and interact with the physical world.

Join Us

Announcing **RAGEN**: Training RL Agents - Github ⭐ 1.5k >

Chain of Experts


Mixture-of-Experts
Output Hidden
y
1
2
3
4
...
N-1
N
Router
Top K
Input Hidden
$x^{(0)}$
Chain-of-Experts
$x^{(1)}$
$x^{(2)}$
y
Intermediate
...
Output Hidden
1
2
3
4
...
N-1
N
Router
...
Router
Top K/C
Input Hidden
...
Intermediate
$x^{(0)}$
$x^{(1)}$
$x^{(C-1)}$