ICML

International Conference
On Machine Learning

Enhancing Decision-Making of Large Language Models via Actor-Critic

Heng Dong*[1], Kefei Duan*[2], Chongjie Zhang[2]

[1]Tsinghua University, [2]Washington University in St. Louis

Motivation

- **LLMs** are powerful in NLP but struggle in **complex decision-making** tasks.

- **Challenges:**
  - Limited long-term reasoning (auto-regressive bias).
  - Fragile rollout-based planning.

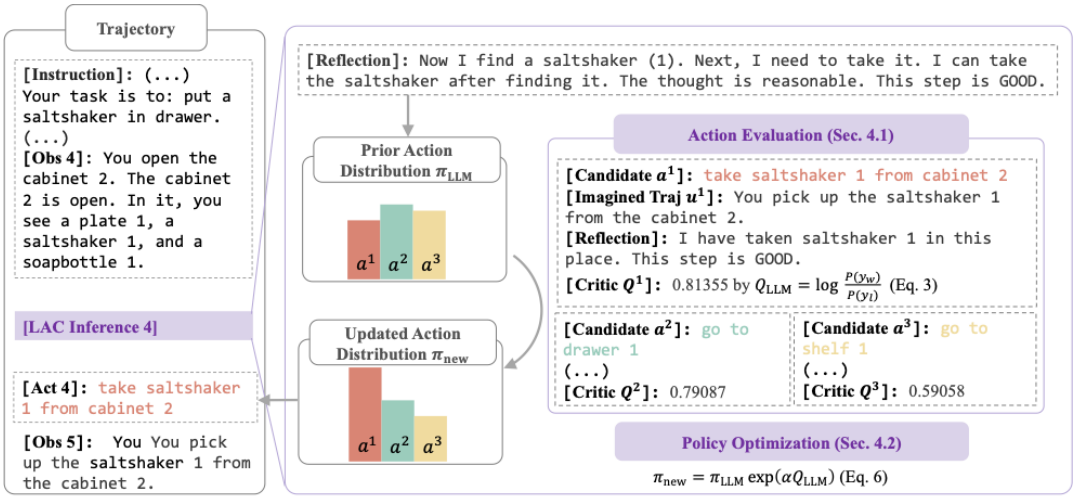
- **Goal:** Enable robust and scalable decision-making with LLMs using reinforcement learning insights.

Key Idea — LAC Framework

- **LAC: LLM-based Actor-Critic**
  - Combines **LLM prior policy** (actor) with **action evaluation** (critic).
  - Critic uses **logits of positive/negative tokens** to compute **Q-values.**
  - Optimizes policy via **gradient-free KL-constrained update.**

Algorithm 1 LAC: LLM-based Actor-Critic algorithm.

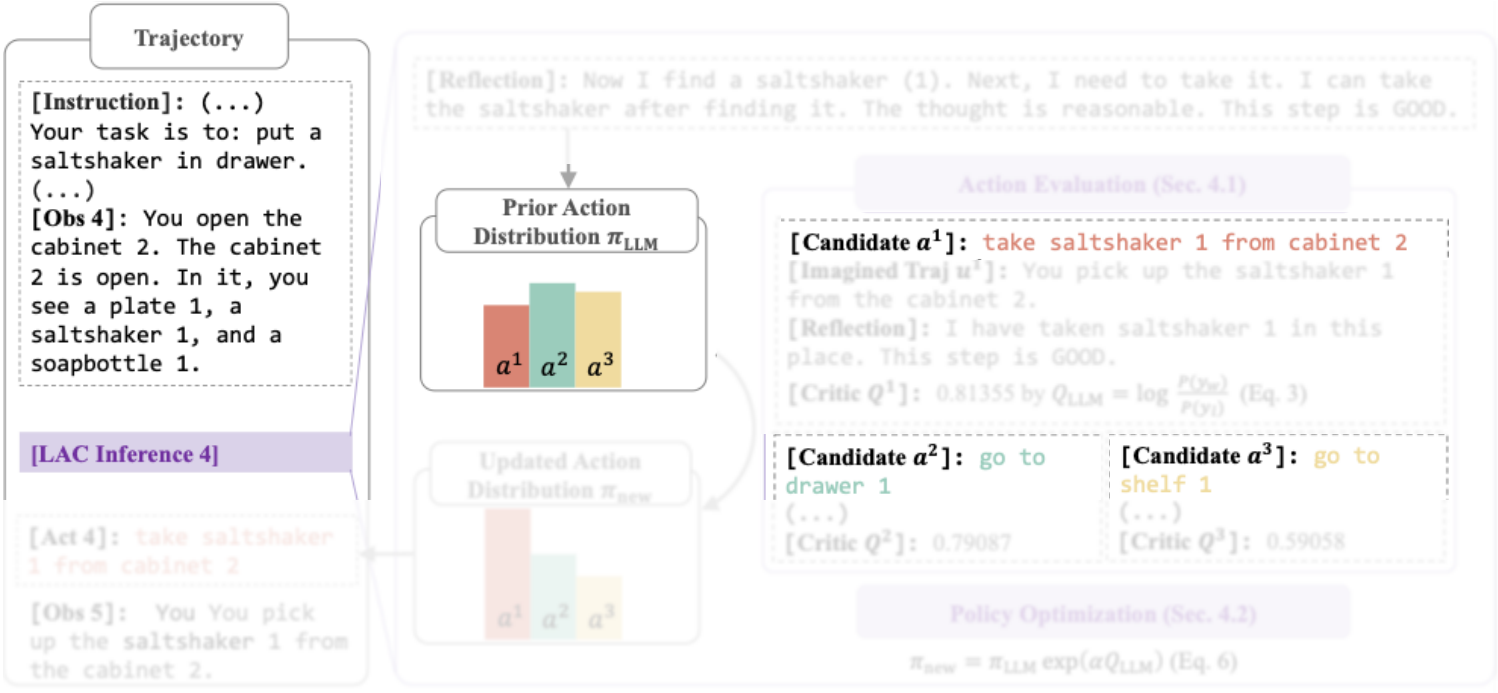
Input: current task goal g, history h_t, actor π_{LLM}, forward model f_{LLM}, value-based critic $\mathcal{Q}_{\text{LLM}}$, hyperparameter α, candidate action size n.
Output: selected action a_t^*
$\{a_t^i\}_{i=1}^n \sim \pi_{\text{LLM}}(\cdot|g, h_t)$; ▷ candidate actions
for $i = 1$ **to** n **do**
  $u_t^i \leftarrow f_{\text{LLM}}(g, h_t, a_t^i)$; ▷ predict future trajectory
  $\mathcal{Q}_{\text{LLM}}(g, h_t, a_t^i, u_t^i) \leftarrow \log \frac{P(y_w|g,h_t,a_t^i,u_t^i)}{P(y_l|g,h_t,a_t^i,u_t^i)}$; ▷ action evaluation (Sec. 4.1)
end for
$\pi_{\text{new}}(a_t^i|g, h_t) \leftarrow \pi_{\text{LLM}}(a_t^i|g, h_t) \exp(\alpha \mathcal{Q}_{\text{LLM}}(g, h_t, a_t^i, u_t^i))$; ▷ policy optimization (Sec. 4.2)
$a_t^* \leftarrow \arg\max_{a_t^i} \pi_{\text{new}}(a_t^i|g, h_t)$



How LAC Works

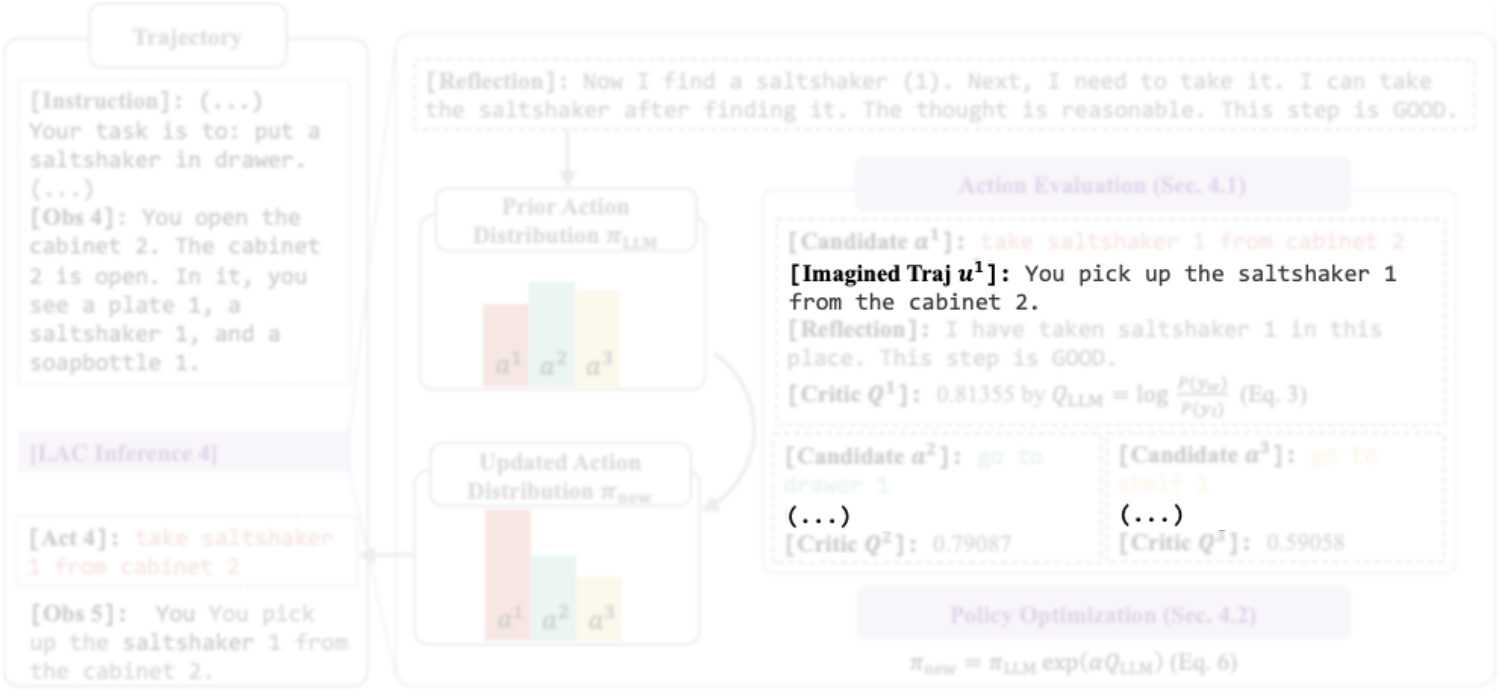
How LAC Works

1. **Sample actions** from LLM (π_{LLM}).



How LAC Works

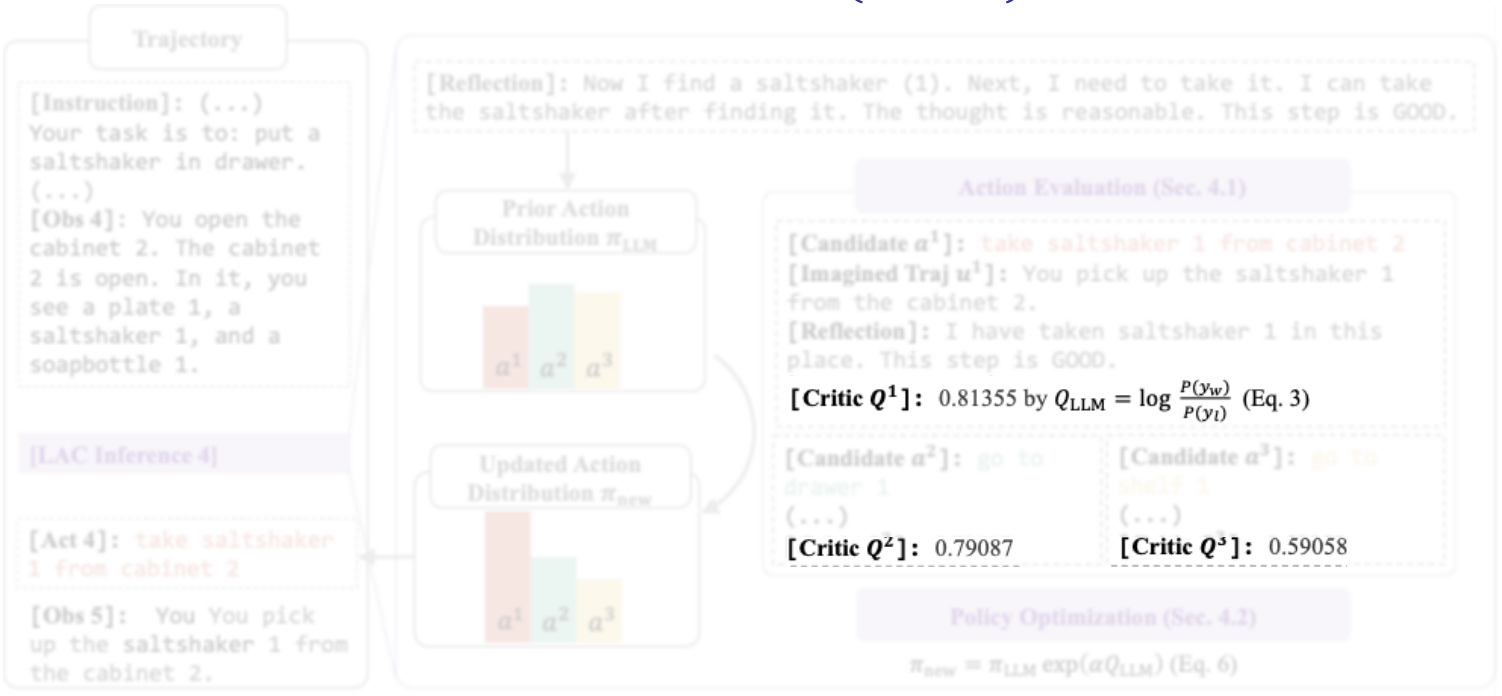
2. Predict future trajectories (via f_{LLM}).



How LAC Works

3. Evaluate each action:

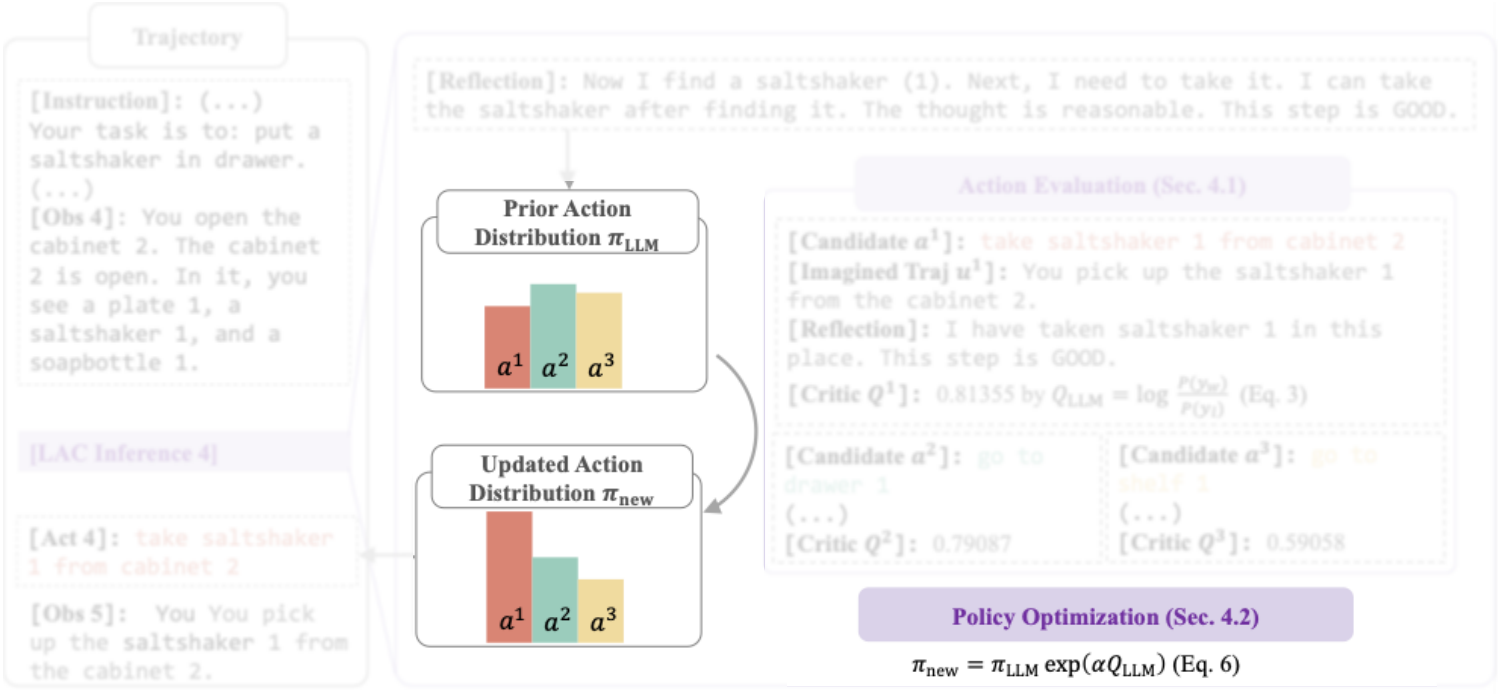
$$Q(a) = \log \frac{P(\text{“GOOD”})}{P(\text{“BAD”})}$$



How LAC Works

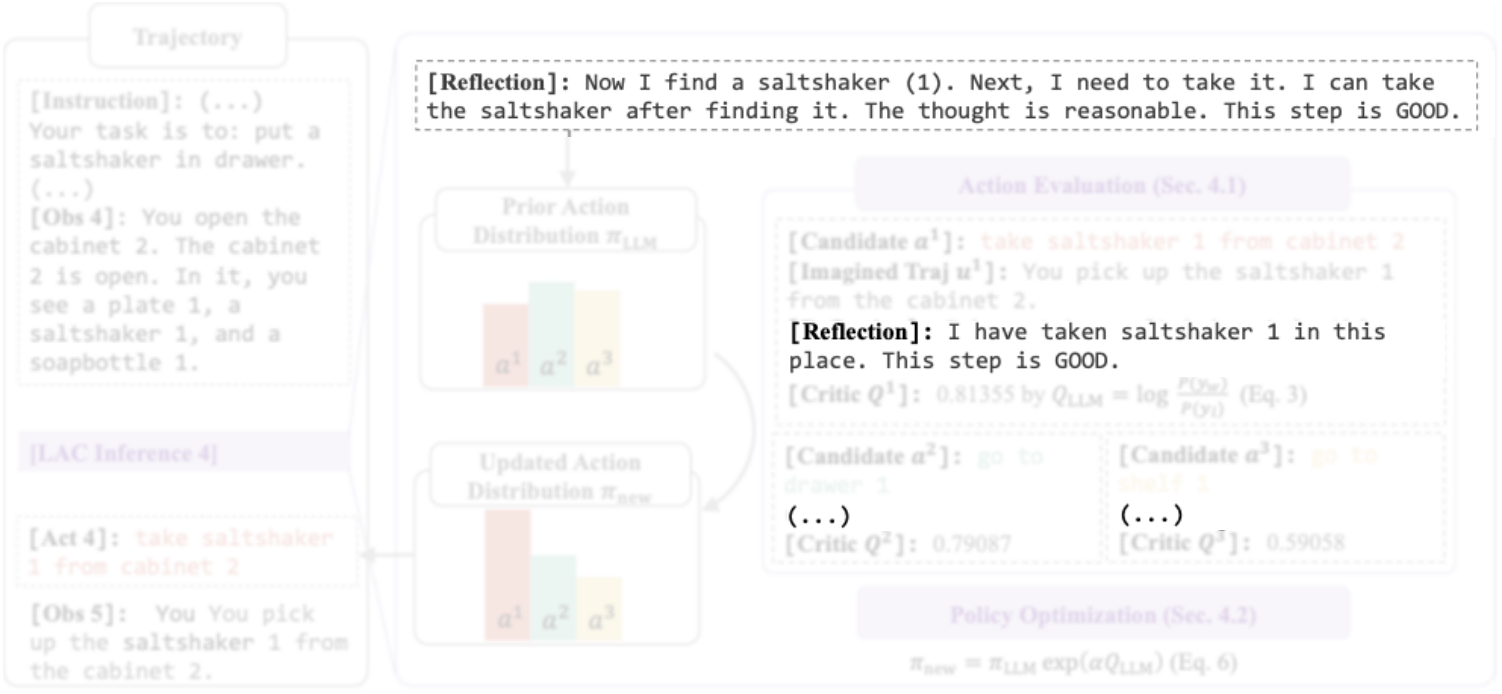
4. Update policy:

$$\pi_{new} \propto \pi_{LLM} \cdot \exp(\alpha\, Q)$$



How LAC Works

🧠 **Reflections:** LLM also generates self-judgments ("This step is GOOD") to aid evaluation.

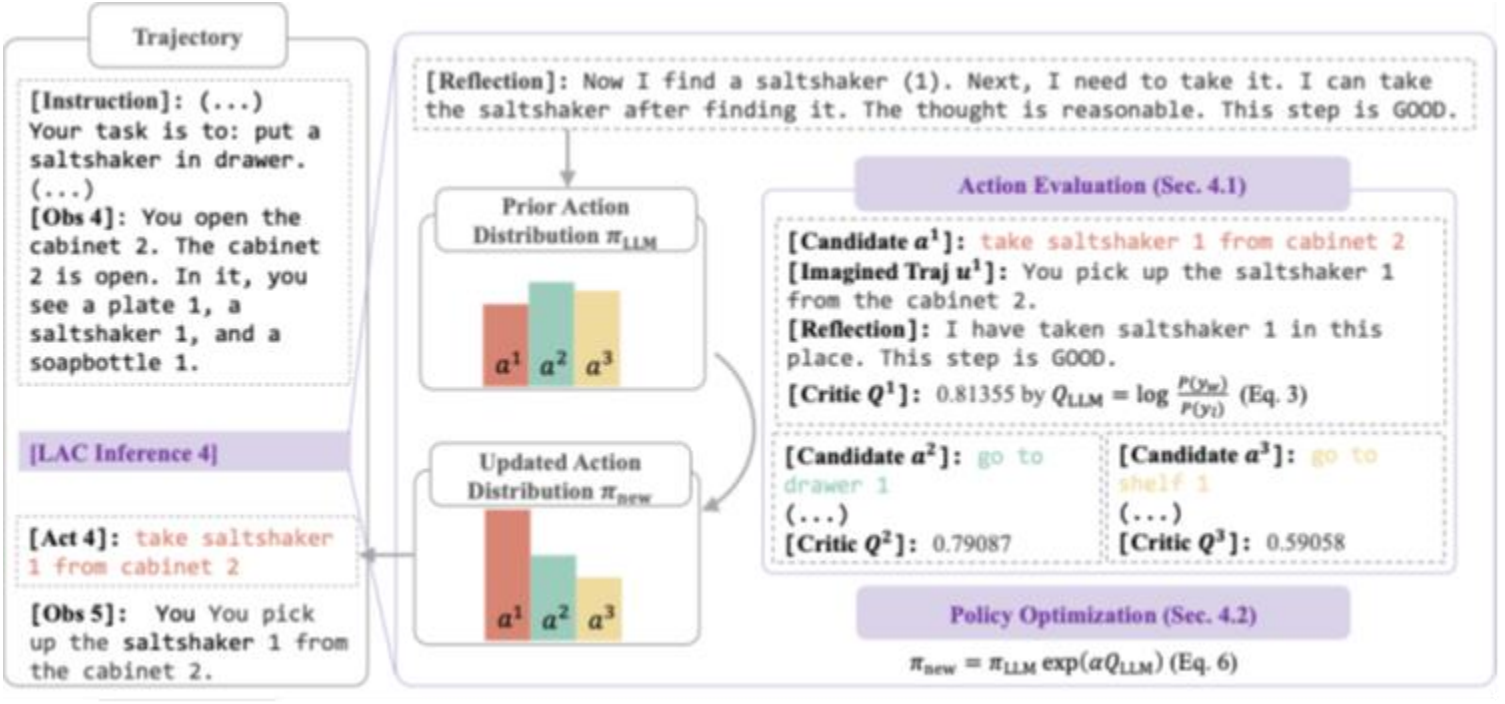


How LAC Works

- Sample an action from π_{new} and then execute it in the environment

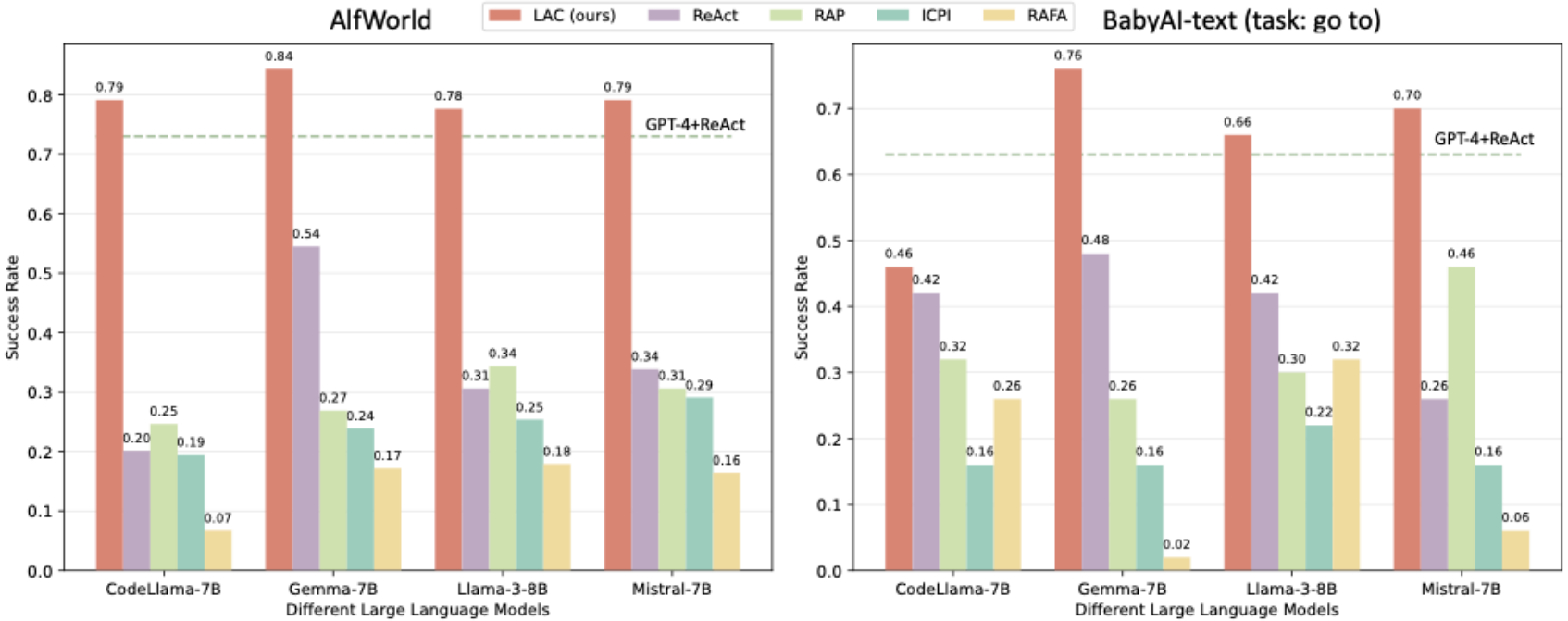
How LAC Works

- This is the full algorithm process.
-



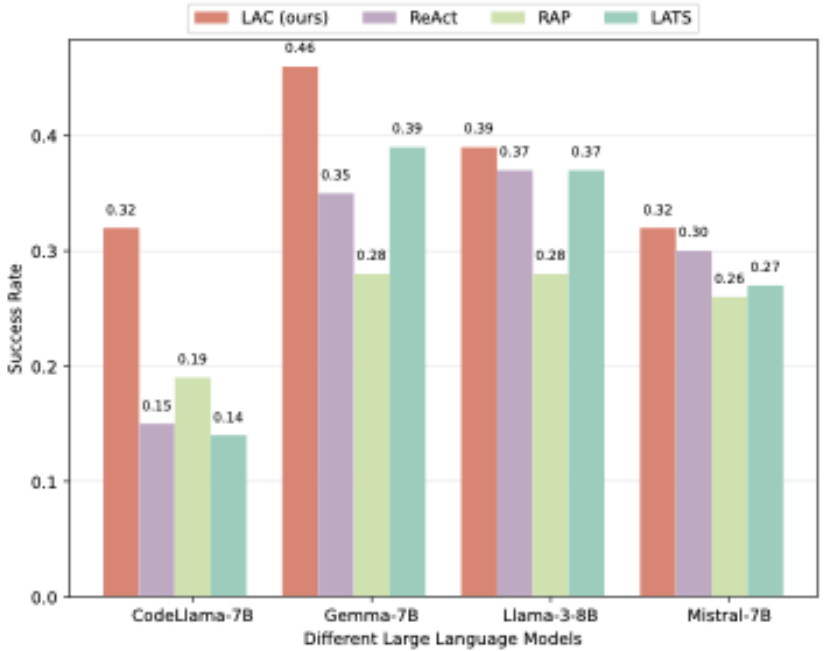
Results & Benchmarks

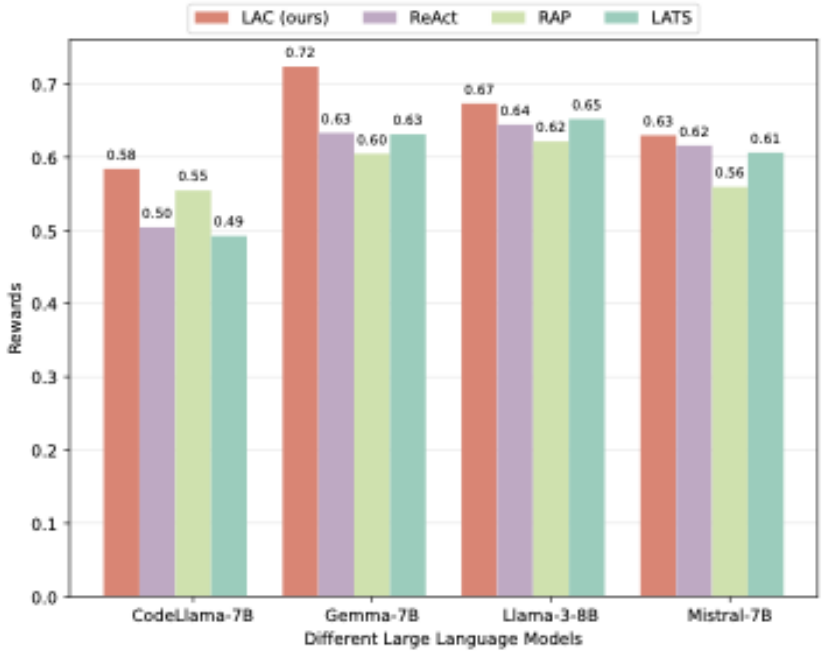
- **Benchmarks**
  - ALFWorld (high-level), BabyAI-Text (low-level), WebShop (infinite actions)



Results & Benchmarks

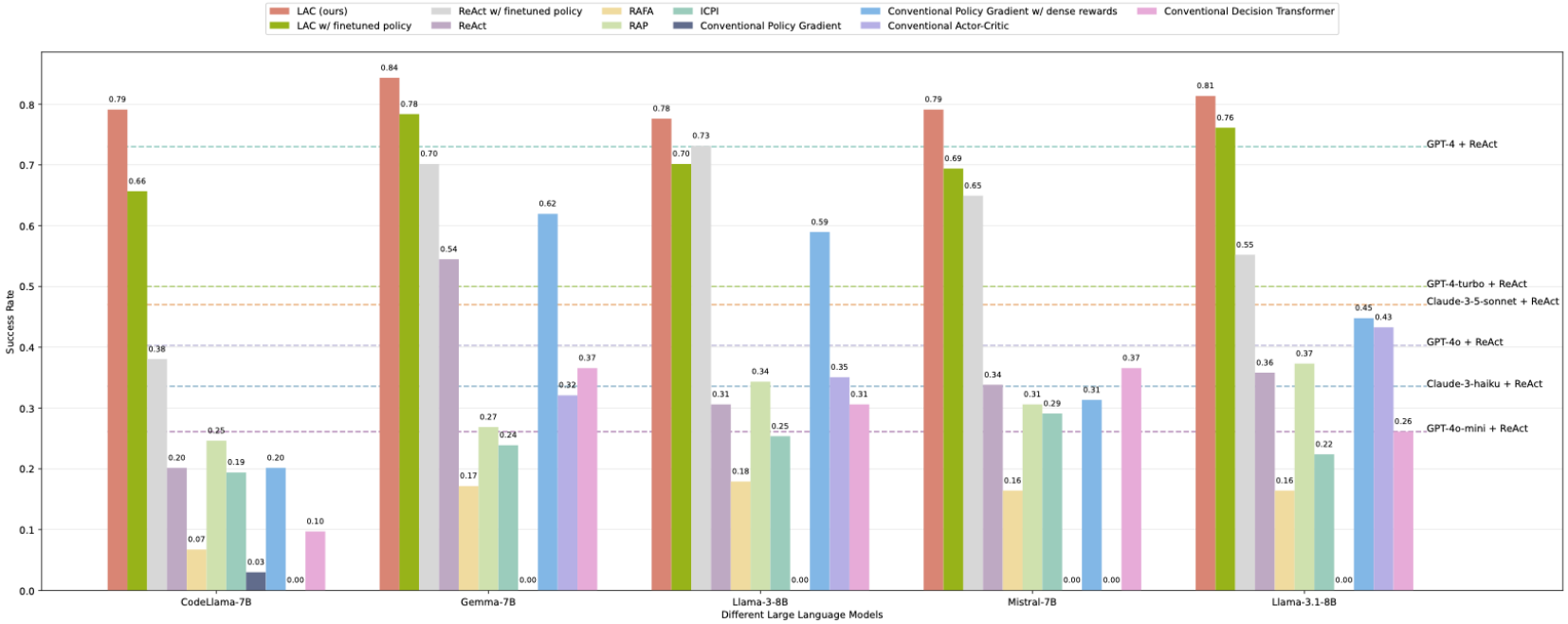
- **Benchmarks**
  - ALFWorld (high-level), BabyAI-Text (low-level), WebShop (infinite actions)





Results & Benchmarks

- **Highlights**
  - Outperforms ReAct, RAP, ICPI, even GPT-4+ReAct in many tasks
  - Works well across **multiple open-source LLMs** (e.g., CodeLlama, Mistral)



Summary & Takeaways

- **Summary**
  - New way to combine LLM priors with evaluated planning
  - Works with lightweight LLMs
  - Gradient-free, interpretable, and generalizable

- 💡 **Future**
  - Deeper planning with trees, adaptation to continuous rewards.

Our code: https://github.com/drdh/LAC

ICML

International Conference
On Machine Learning

Thanks for Your Listening