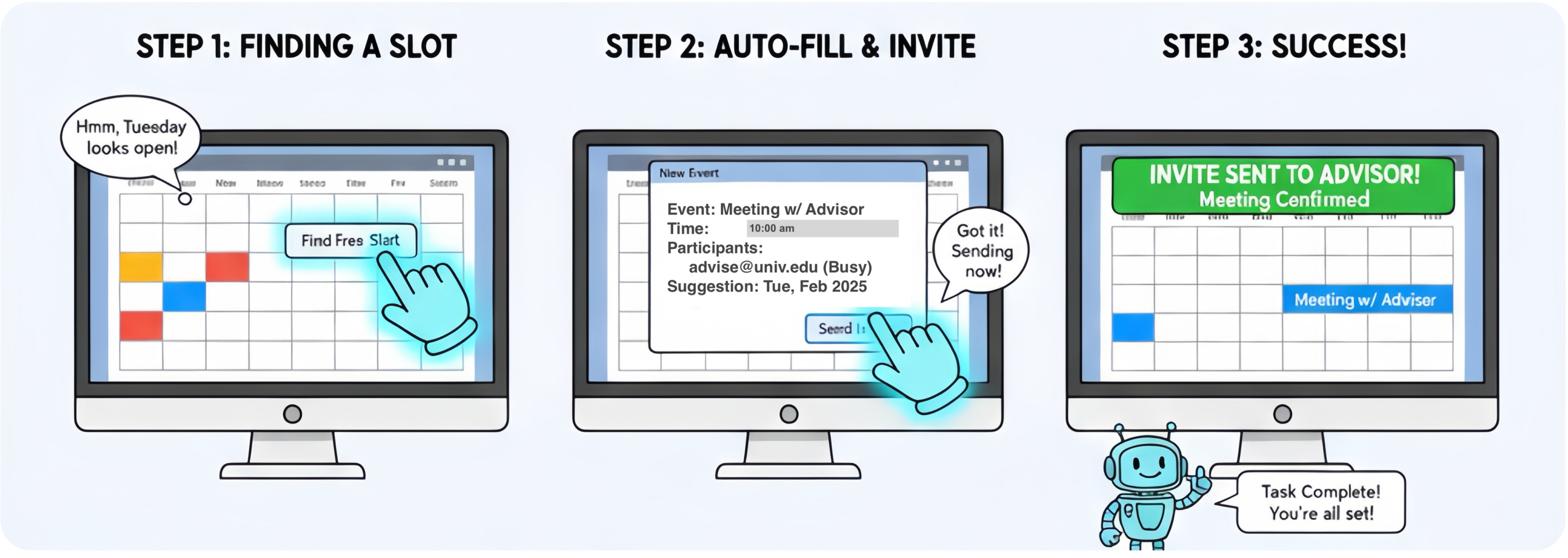
Inference-Time Scaling for GUI Agents with Process Reward Models and Internal World Models

Bella Wang yuw170@ucsd.edu
Rita Yujia Wu yuw172@ucsd.edu
Shuchang Liu shl153@ucsd.edu
Ziyu Huang zih029@ucsd.edu
Mentors: Kun Zhou, Zhiting Hu kuzhou@ucsd.edu zhh019@ucsd.edu

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Background & Introduction

GUI agents are systems that can use real software like a person. They look at the screen, click buttons, type into forms, and navigate multi-step workflows across browsers, operating systems, and mobile apps. Instead of building a custom integration for every app, a capable agent could automate many everyday tasks end-to-end.

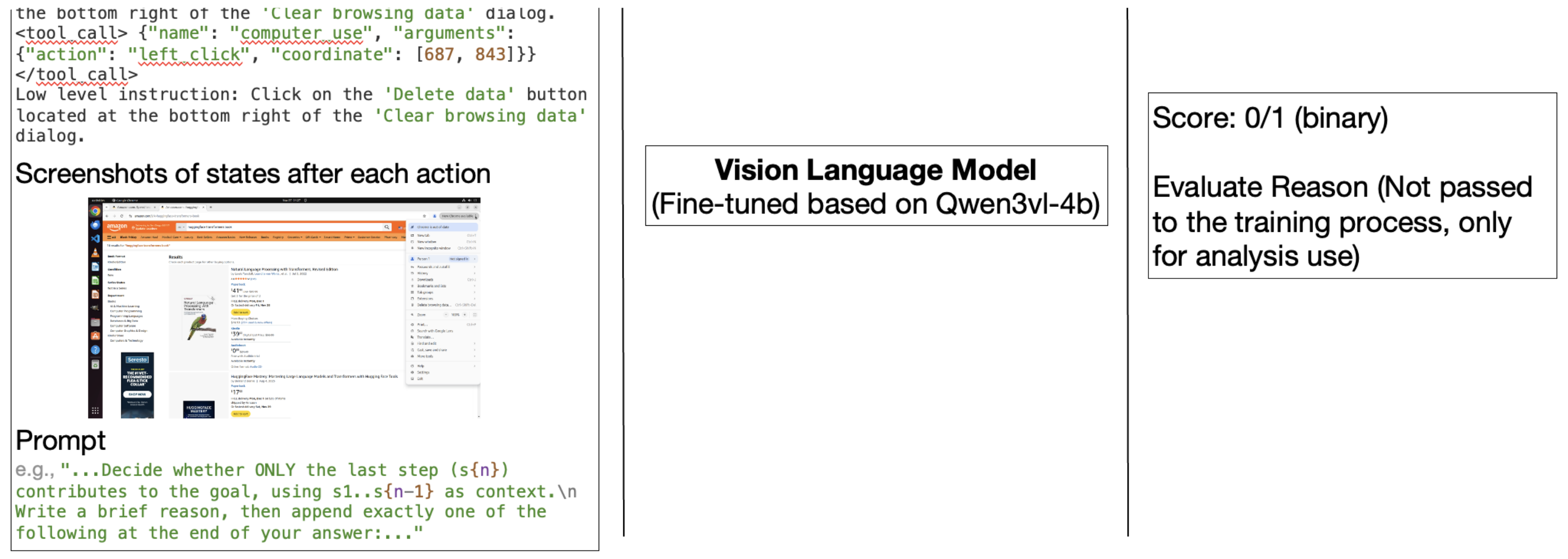


However, the hardest part is making these long sequences reliable. Therefore, in this project, we decided to approach it by letting the model take longer action sequences and consider more possibilities (inference-time scaling):

- **Training with a Process Reward Model (PRM):** teaches the policy what "good progress" looks like at each step, so longer rollouts stay goal-directed. (Like a coach saying "yes, keep going" / "no, that's a detour" during the learning process.)
- **Inference with an internal world model:** dynamically retrieves contrastive past experience to support *think-before-acting* reasoning and guide action selection. (Like recalling a similar situation on the spot.)

Method Part 1: Process Reward Model and Guided Agent Training

Our method is implemented according to the pipeline illustrated in the figure below (!The pipeline will be updated!).



```
the bottom right of the 'Clear browsing data' dialog.
<tool_call> {"name": "computer_use", "arguments":
{"action": "left_click", "coordinate": [687, 843]}}
</tool_call>
Low level instruction: Click on the 'Delete data' button
located at the bottom right of the 'Clear browsing data'
dialog.
```

Screenshots of states after each action

Prompt

e.g., "...Decide whether ONLY the last step (s{n}) contributes to the goal, using s1..s{n-1} as context.\n Write a brief reason, then append exactly one of the following at the end of your answer:..."

Vision Language Model
(Fine-tuned based on Qwen3vl-4b)

Score: 0/1 (binary)

Evaluate Reason (Not passed to the training process, only for analysis use)

1. **Inputs (per step):** task instruction (goal) + recent screenshots + the action the agent just took.
2. **PRM output:** a progress score (0/1 or scaled to [0,1]) and an optional short reason (for debugging only).
3. **How we fine-tune the PRM:** generate and collect OSWorld trajectories → label each step with progress/no-progress with GPT-5-mini → fine-tune a Qwen3-VL-4b model to predict the step score from the inputs above.
4. **How we train agent with PRM:** use Qwen3-VL-4b model as back-bone. During RL, after every action, we query the PRM for a step reward and use that reward to update the agent policy (so longer rollouts stay goal-directed).

PRM Evaluation Result

PRM evaluation on the OSWorld test set (505 three-step windows). The following are the confusion matrix counts and the corresponding derived metrics.

Model	TP	FP	FN	TN	Acc.	Prec.	Rec.	F1
Qwen3-VL-4B (zero-shot PRM)	111	9	304	81	38.02%	92.50%	26.75%	41.50%
Fine-tuned PRM (LLaMA-Factory)	313	41	102	49	71.68%	88.42%	75.42%	81.40%

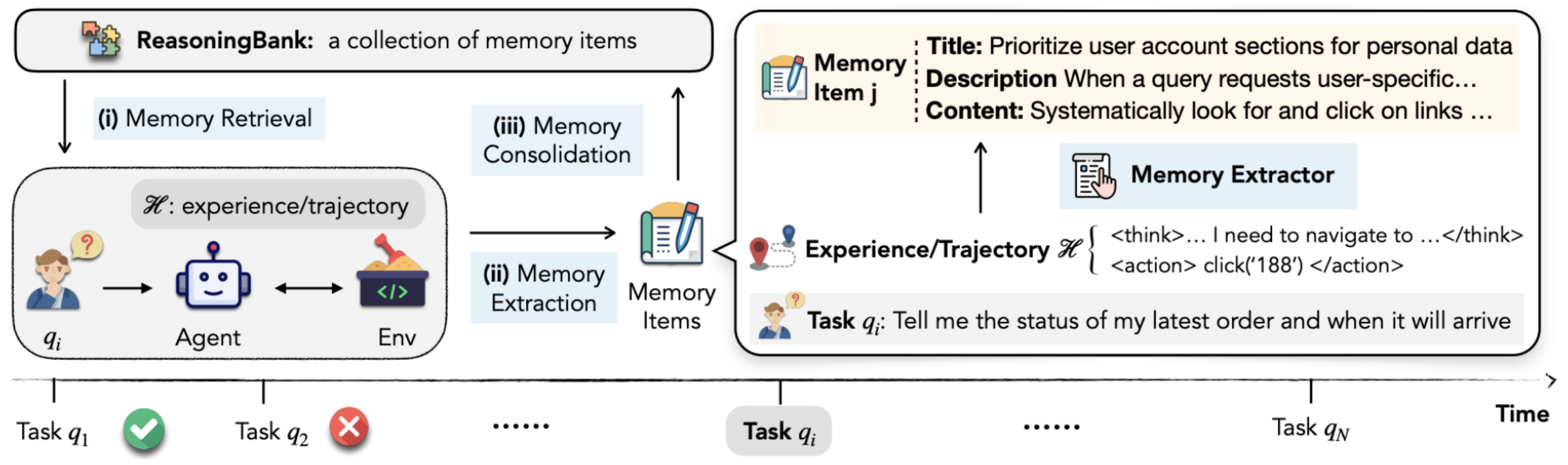
Agent Evaluation Result (Place Holder)

We evaluate our trained agent on the various planning benchmarks, the results are shown in the following table. (!Place Holder for the Table below, result is in progress!)

Model	TP	FP	FN	TN	Acc.	Prec.	Rec.	F1
Qwen3-VL-4B (zero-shot PRM)	111	9	304	81	38.02%	92.50%	26.75%	41.50%
Fine-tuned PRM (LLaMA-Factory)	313	41	102	49	71.68%	88.42%	75.42%	81.40%
Qwen3-VL-4B (zero-shot PRM)	111	9	304	81	38.02%	92.50%	26.75%	41.50%
Fine-tuned PRM (LLaMA-Factory)	313	41	102	49	71.68%	88.42%	75.42%	81.40%
Fine-tuned PRM (LLaMA-Factory)	313	41	102	49	71.68%	88.42%	75.42%	81.40%

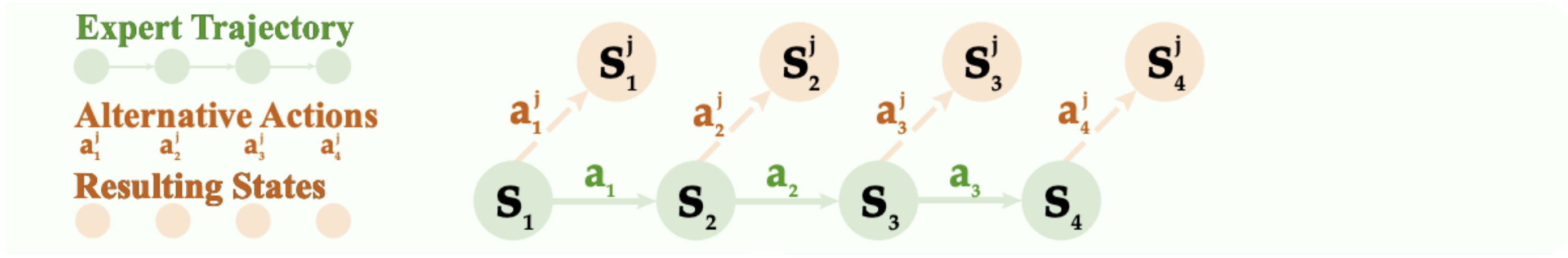
Method Part 2: Internal World Model

Goal: Improve long-horizon GUI reliability by helping the agent *think before acting* via contrastive retrieval of past successes and failures, reducing repeated failure patterns.

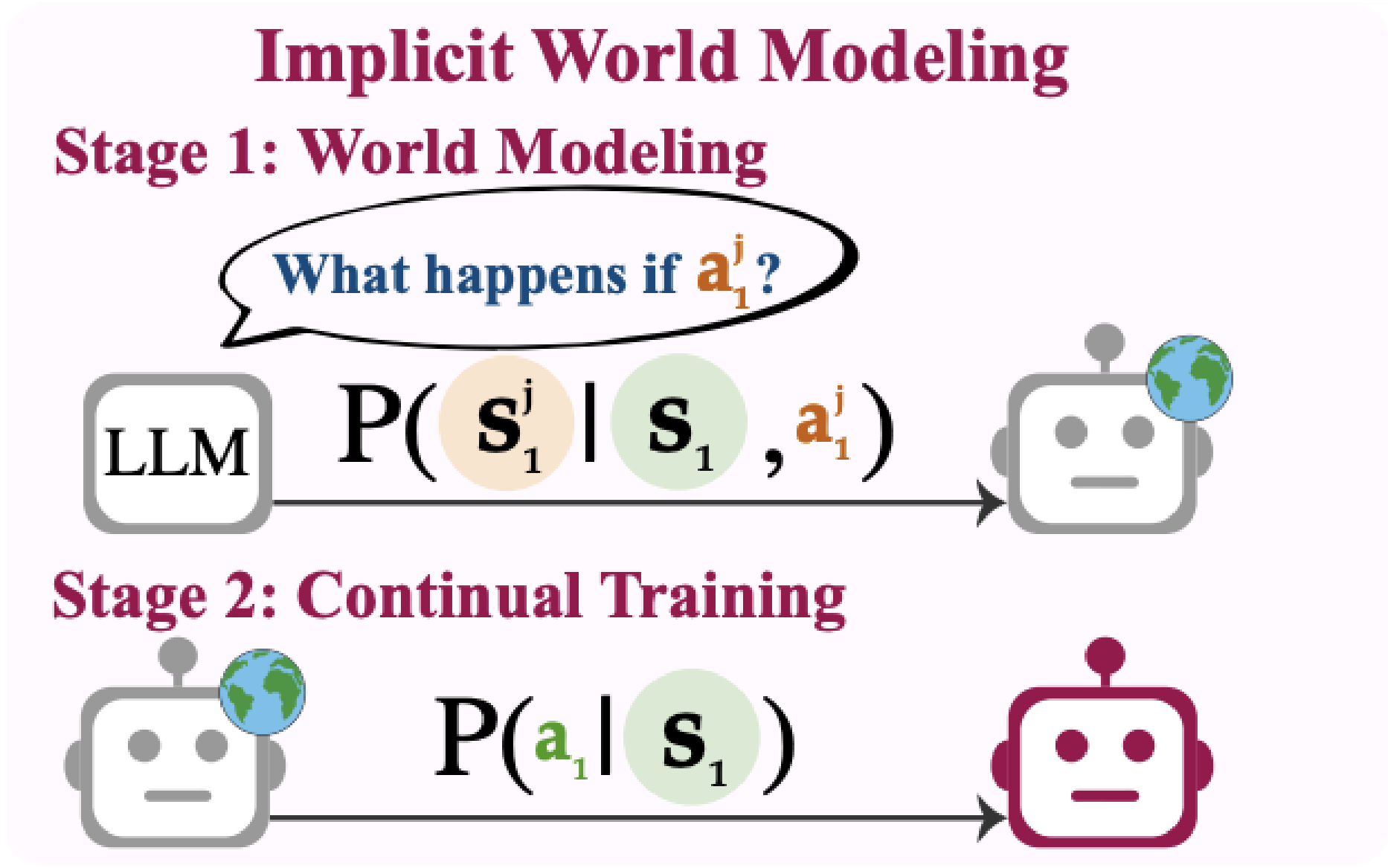


ReasoningBank: a collection of memory items
(i) Memory Retrieval
(iii) Memory Consolidation
$\mathscr{H}$: experience/trajectory
q_i
Agent
Env
(ii) Memory Extraction
Memory Items
Memory Item j
Title: Prioritize user account sections for personal data
Description When a query requests user-specific...
Content: Systematically look for and click on links ...
Memory Extractor
Experience/Trajectory $\mathscr{H}$
<think>... I need to navigate to ...</think>
<action> click('188') </action>
Task q_i: Tell me the status of my latest order and when it will arrive
Task q_1
Task q_2
......
Task q_i
......
Task q_N
Time

Experience abstraction. Agent trajectories are converted into structured memory items and indexed with FAISS. We maintain separate memory banks for successful and failed trajectories to enable contrastive retrieval at inference time.



Candidate-action guidance. At each step, the agent evaluates multiple candidate actions. The world model retrieves similar past trajectories and uses contrastive evidence to prioritize actions aligned with successful behaviors while avoiding known failure patterns.



Contrastive world model pipeline. Given the current screenshot and task, the system:

- **Retrieves** top-k success and failure trajectories via FAISS + CLIP embeddings
- **Analyzes** divergence points between successful and failed action sequences
- **Summarizes** key success patterns and common pitfalls into ∼200-token guidance
- **Injects** guidance into the agent prompt (initial + step-level)

To improve robustness, we incorporate lightweight confidence checks and careful evidence aggregation across retrieved neighbors, which helps stabilize guidance when retrieval signals are weak or ambiguous.

World Model Evaluation Result

We evaluate the internal world model on WebVoyager by comparing a CoMEM-style **success-only retrieval** baseline against our **contrastive (success+failure) retrieval** with **dynamic step-level guidance**. The contrastive world model improves success rate on 3/4 domains, with the largest gain on Amazon (more results in progress).

Domain	Baseline	World Model	Δ (pp)
Google Maps	16.67%	26.83%	+10.16
Amazon	19.51%	39.02%	+19.51
Allrecipes	15.91%	11.11%	-4.80
Coursera	2.38%	9.52%	+7.14

Setup: top-k=3 success + 3 failure trajectories retrieved per step (FAISS + CLIP); dynamic re-retrieval each step.
Dynamic retrieval: on average, 40% of retrieved trajectories change between step 0 and step 5.
Index stats (Amazon example): ∼150 success / ∼80 failure trajectories; avg. length 8.3 steps (success) / 12.1 steps (failure).

Evaluation Results after Combining Methods (On The Way)

We combine **PRM-guided training** with the **contrastive internal world model** to study whether external process supervision and internal retrieval-based reasoning provide complementary benefits for long-horizon GUI tasks.

Preliminary Findings

- The combined agent shows more consistent multi-step behavior and fewer obvious failure loops.
- Contrastive guidance is most helpful when the PRM-trained policy faces ambiguous GUI states.
- Early results suggest the two methods are **complementary rather than redundant**, though full-scale evaluation is ongoing.

Conclusions

For PRM-Guided Agent Training, we built and curated datasets for PRM training, fine-tuned a PRM, and used it to provide dense step-wise rewards for GUI-agent training, with further result analysis and evaluation still ongoing. (Followed by place holders in bullet points)

- **Mauris tempor** risus nulla, sed ornare
- **Libero tincidunt** a duis congue vitae
- **Dui ac pretium** morbi justo neque, ullamcorper

For the Internal World Model, we introduced a contrastive memory mechanism that retrieves both successful and failed trajectories to provide step-level guidance during inference.

- **Contrastive retrieval memory:** Built dual FAISS indices (success vs. failure) with CLIP embeddings to implicitly model action outcomes.
- **Dynamic step-level guidance:** Re-retrieval at each step keeps guidance aligned with the current GUI state and improves long-horizon reliability.
- **Empirical gains:** Improves WebVoyager success rate on 3/4 domains (largest gain on Amazon), while highlighting retrieval sensitivity on heterogeneous sites.

To combine the methods,

- **PRM (training-time):** provides dense step-wise rewards that improve policy learning and action quality.
- **World model (inference-time):** enables *think-before-acting* behavior via contrastive retrieval of past successes and failures.
- **Combined effect:** PRM improves the base policy, while the world model stabilizes decision making at deployment.

References

[1] Wenyi Wu, Kun Zhou, Ruoxin Yuan, Vivian Yu, Stephen Wang, Zhiting Hu, and Biwei Huang. Auto-scaling continuous memory for gui agent. *arXiv preprint arXiv:2510.09038*, 2025.