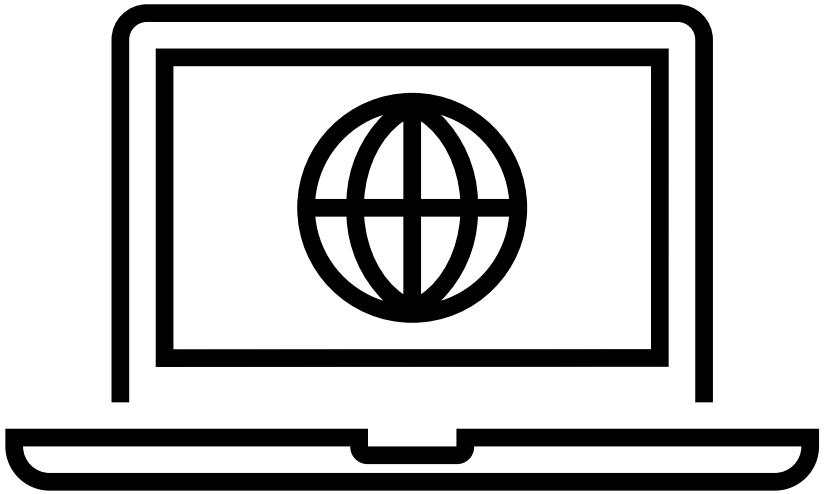
WARC-Bench

Web Archive Based Benchmark for GUI Subtask Executions



Sanjari Srivastava, Gang Li, Cheng Chang, Rishu Garg, Manpreet Kaur, Charlene Y. Lee, Yuezhang Li, Yining Mao, Ignacio Cases, Yanan Xie, Peng Qi

Orby AI / Uniphore

ArXiv: https://arxiv.org/pdf/2510.09872, ICLR 2026

Key Takeaways

- Realistic, lightweight, reproducible **438 GUI subtask** challenges based on web archives (WARCs)
- Claude-Sonnet 4.0 achieves the highest success rate of 64.8%, showing that WARC-Bench is challenging for large foundation models
- Demonstrate SFT + RLVR on open-source models as an effective training technique to bridge the gap with frontier models
- Analyze usefulness of benchmarking subtasks, as well as RLVR training

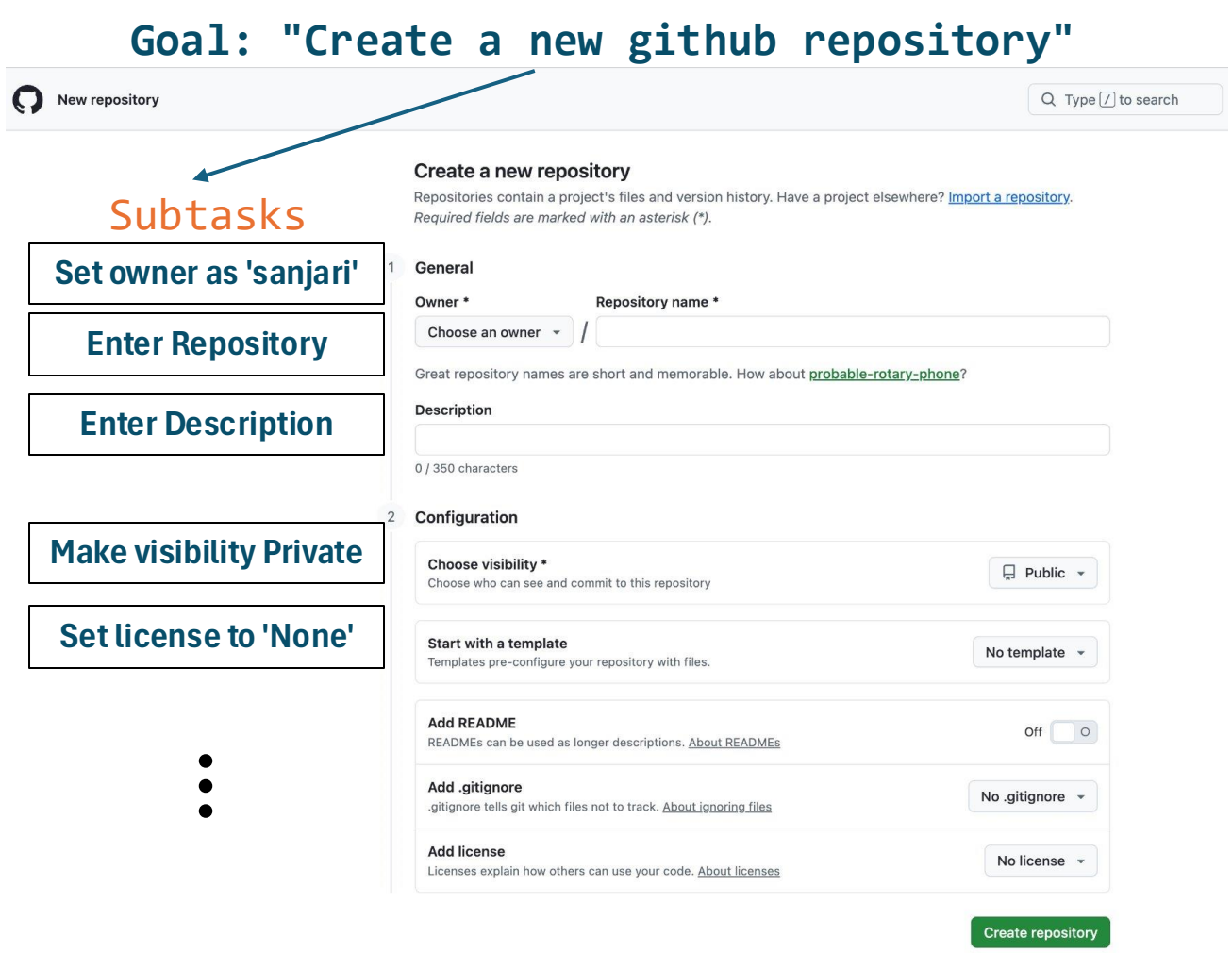
ArXiv: https://arxiv.org/pdf/2510.09872, ICLR 2026

Web agents need to master GUI subtasks

Subtasks — short-horizon interactions w/ UI components within larger browser-based workflows

Includes Scrolling, Rich Text Editing, Data Extraction, Interactions w/ dropdowns, menus, date pickers

Critical building blocks — but no existing benchmark measures them in isolation



How WARC-Bench Works

1. Capture

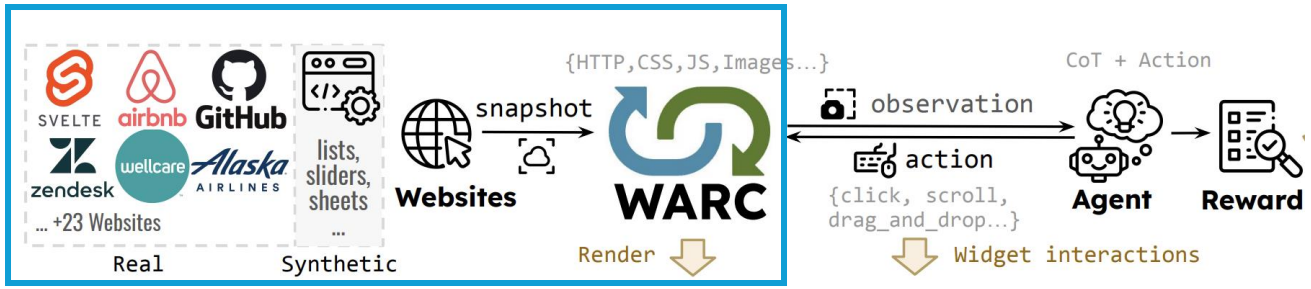
Record real/synthetic websites as Web Archive (WARC) files

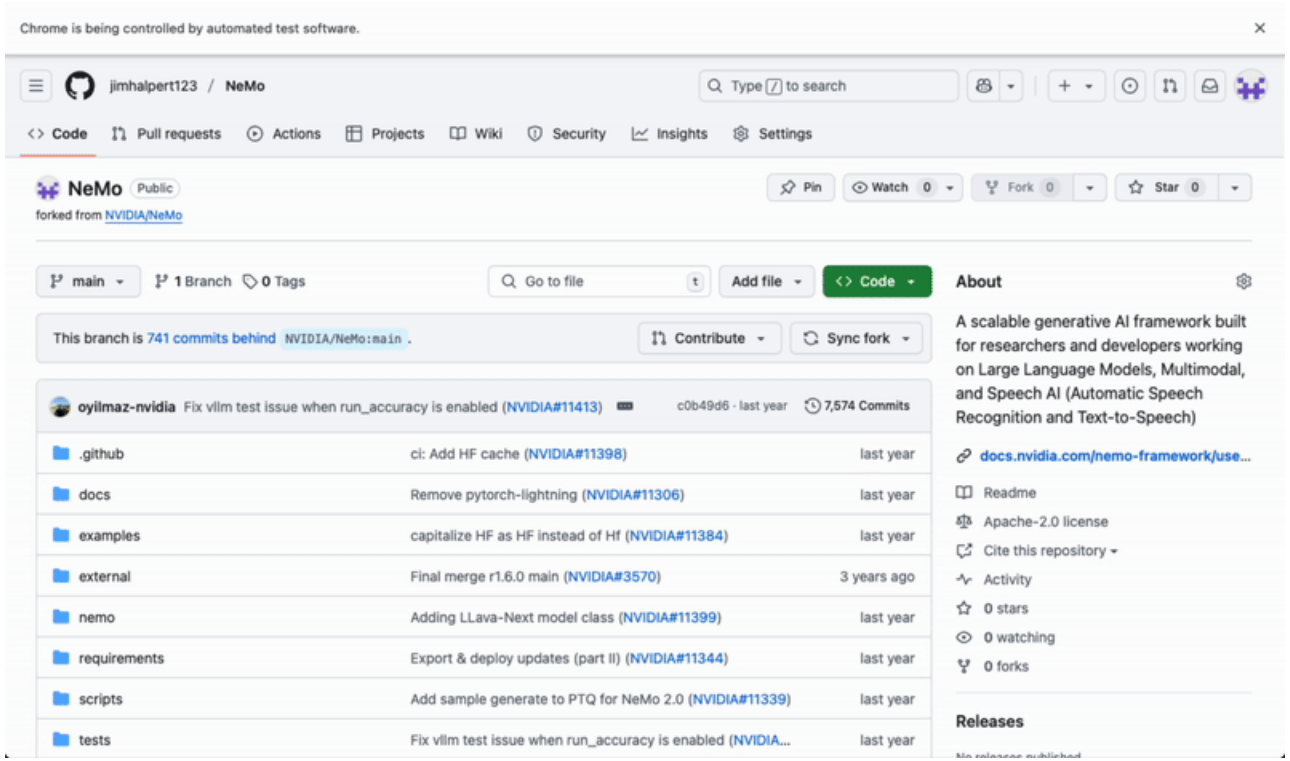
2. Replay

Custom tool for high-fidelity replay in isolated browser

3. Evaluate

Agent performs subtask; verify with per-task verifiable reward





How WARC-Bench Works

1. Capture

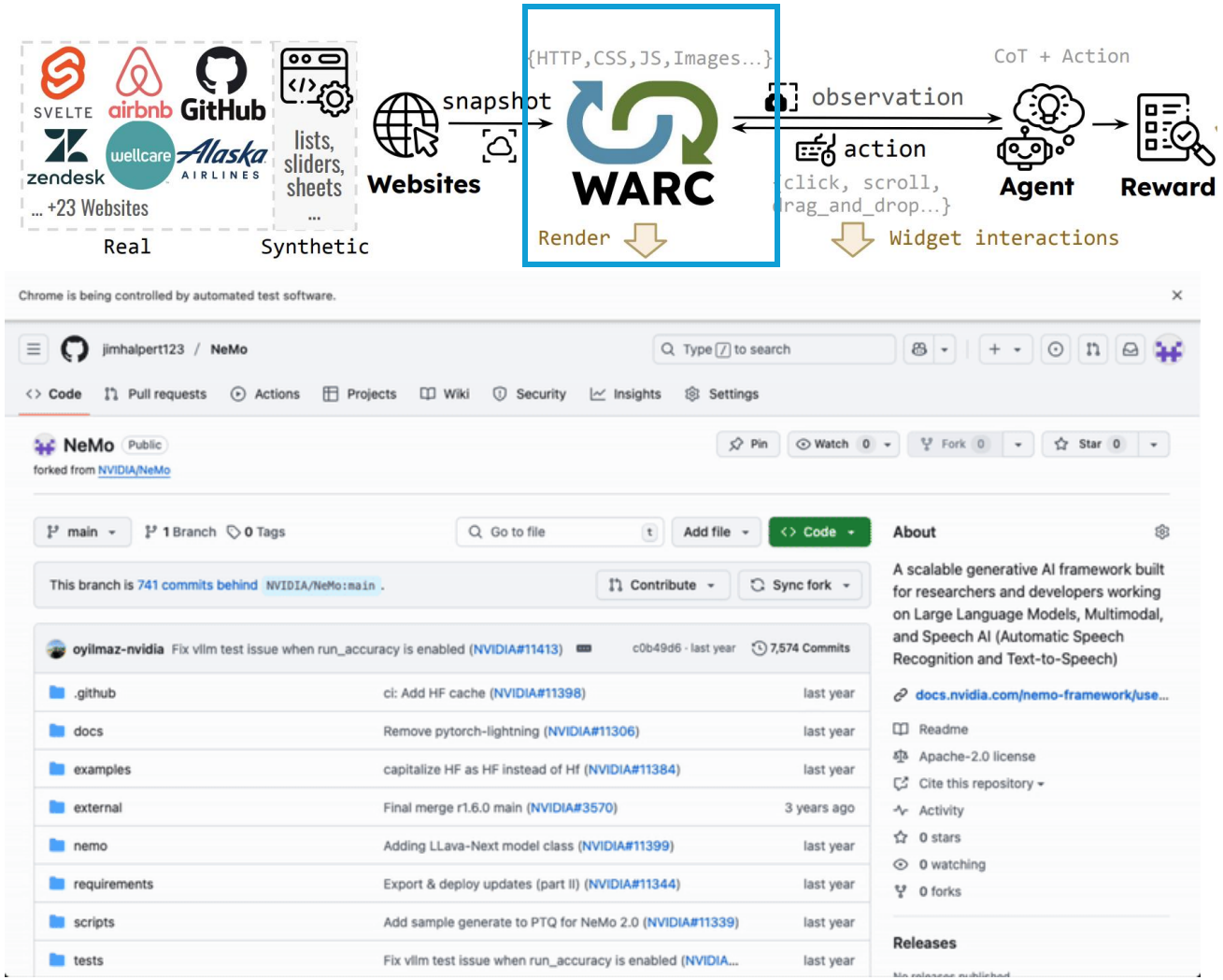
Record real/synthetic websites as Web Archive (WARC) files

2. Replay

Custom tool for high-fidelity replay in isolated browser

3. Evaluate

Agent performs subtask; verify with per-task verifiable reward



How WARC-Bench Works

1. Capture

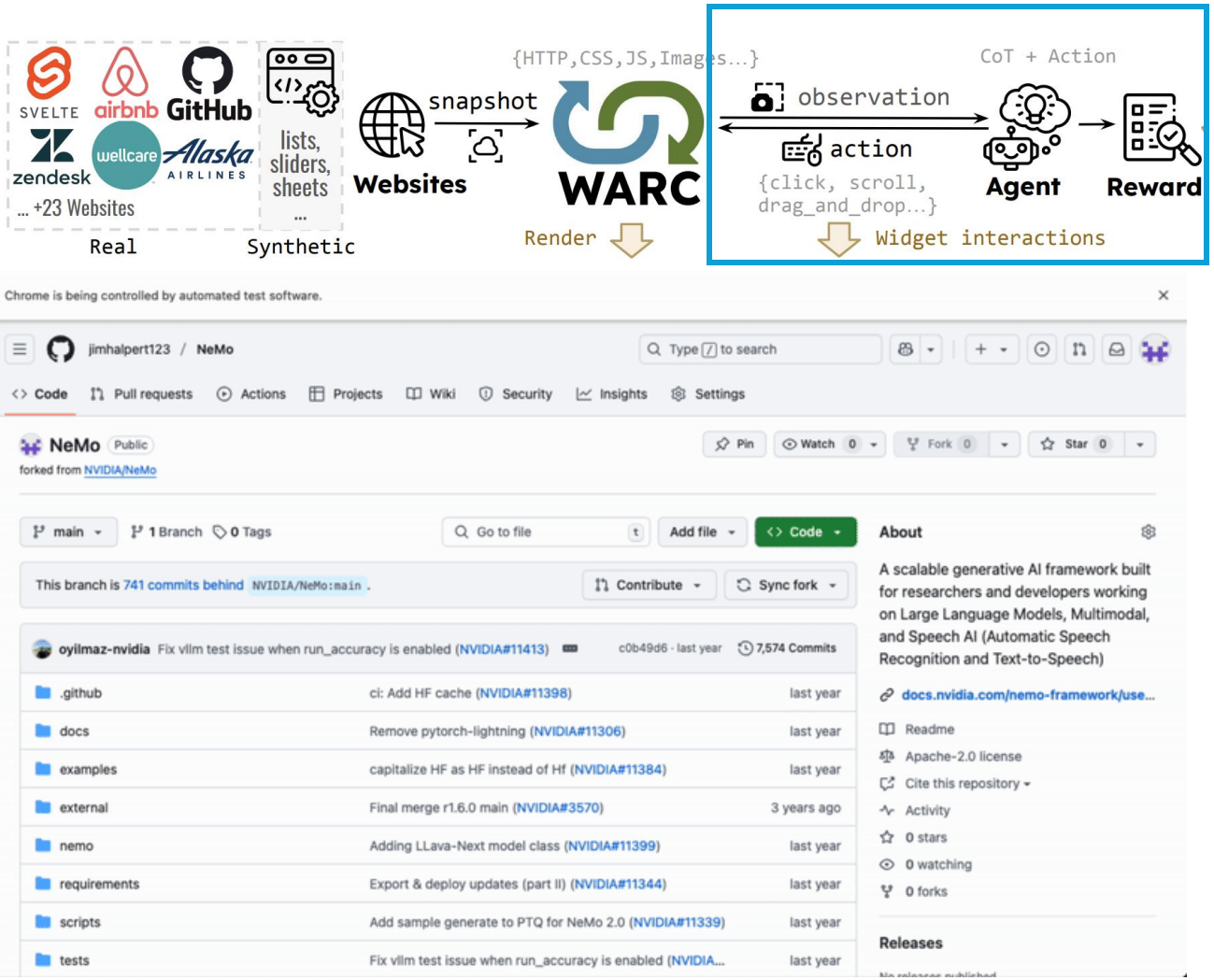
Record real/synthetic websites as Web Archive (WARC) files

2. Replay

Custom tool for high-fidelity replay in isolated browser

3. Evaluate

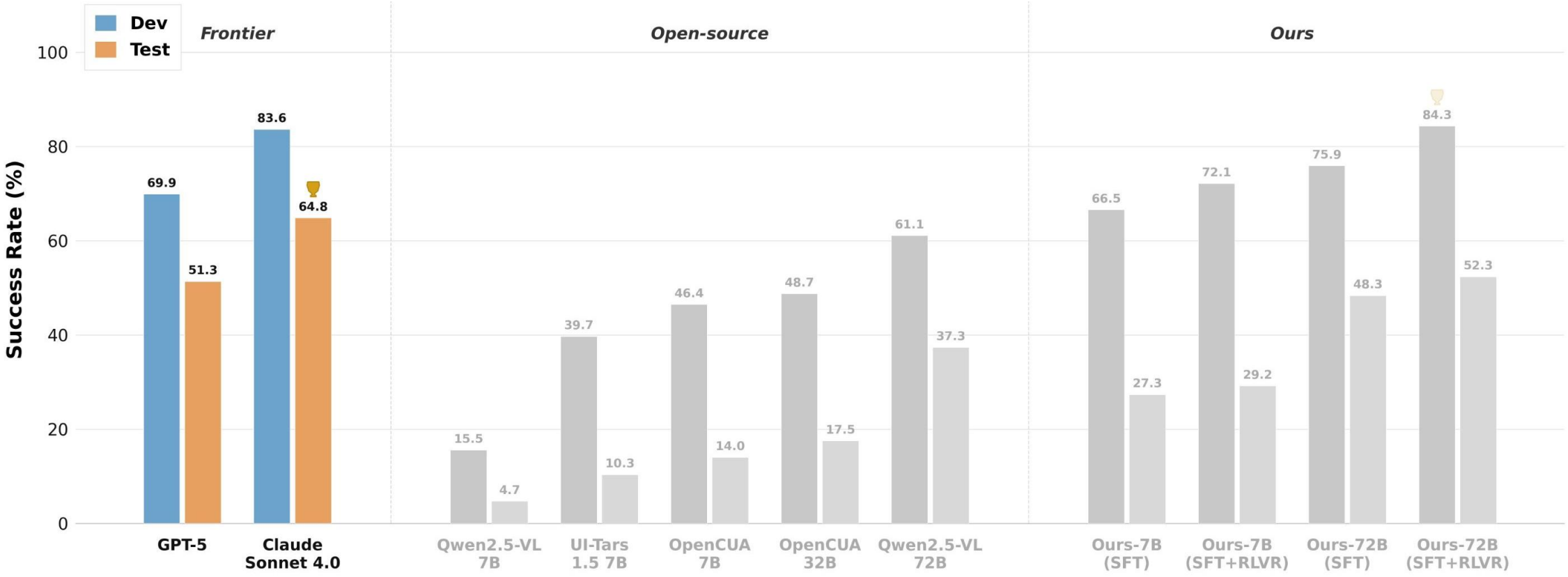
Agent performs subtask; verify with per-task verifiable reward



Contains 1497(1059 train, 200 test, 238 dev) tasks over 29 real and 62 synthetic websites

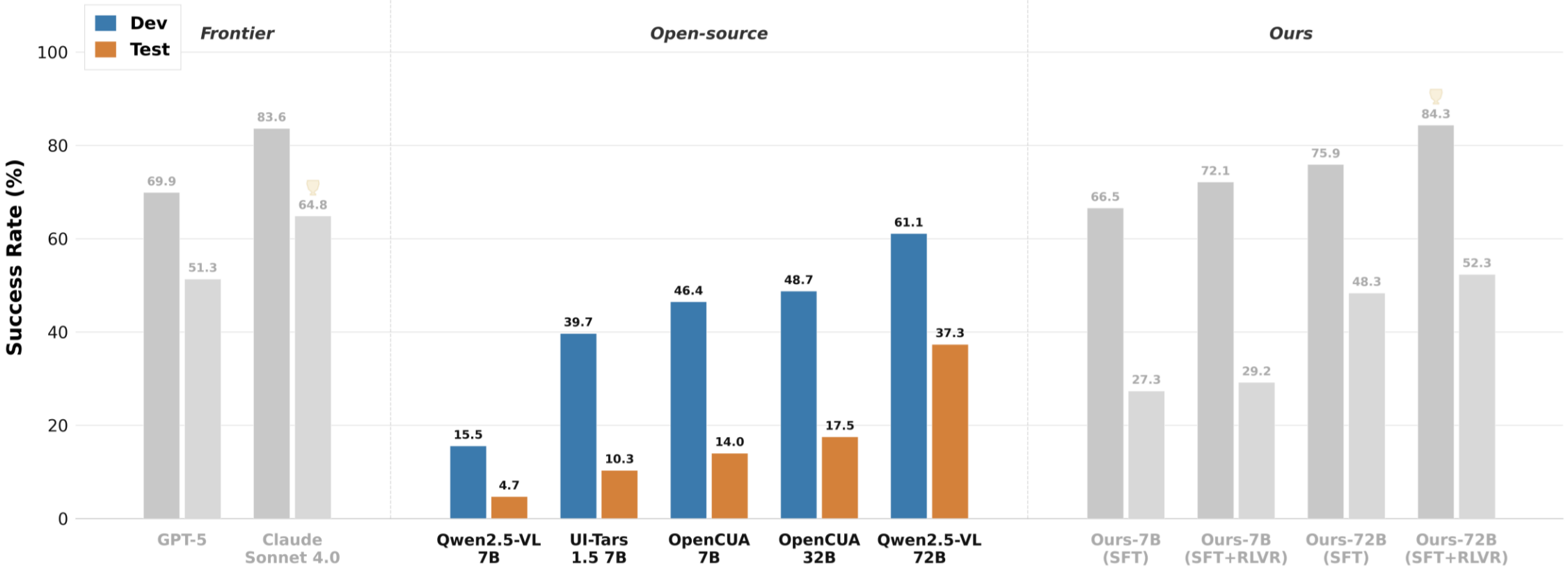
WARC-Bench is challenging for frontier VLMs

Highest test success rate is 64.8% - showing significant room for improvement.



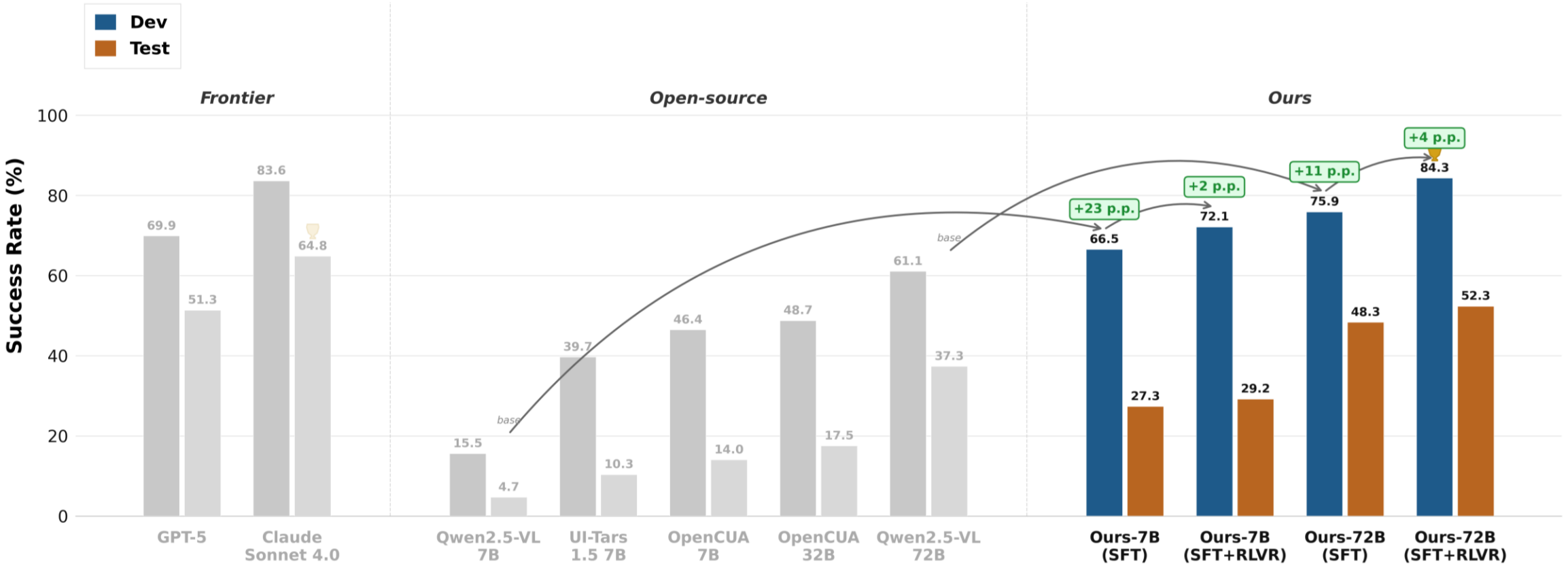
WARC-Bench is challenging for open VLMs

The performance is seen to be even more under-whelming on open-source models



Bridging the gap b/w open and closed models

We train our own models that achieve results competitive w/ frontier models



Effects of RLVR – Better Accuracy, Efficiency

- RLVR on 1059 tasks, improves *form filling, menu navigation, table manipulation,* and *datepicking*
- Gains driven by better visual grounding and exploration capabilities (fewer overall actions, more compound operations).
- 0.94 fewer steps per task compared to the SFT model

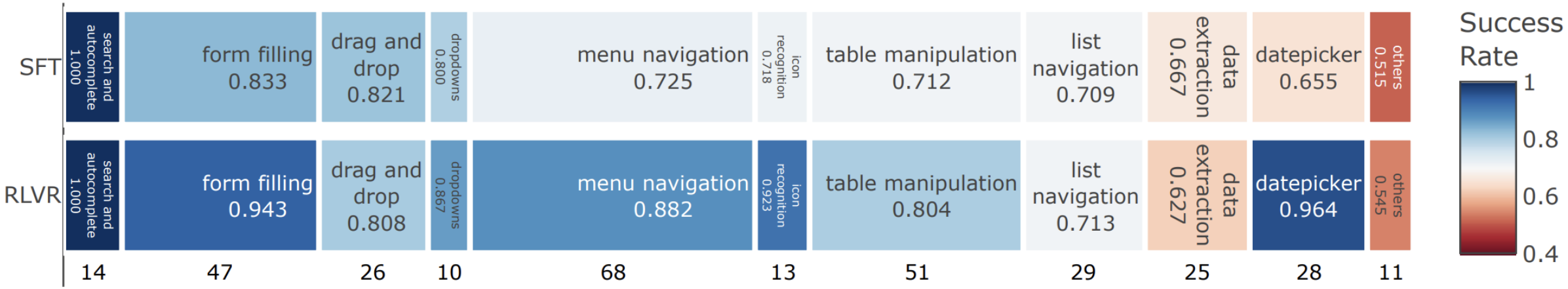


Fig 4a: Success rate comparison for different subtask categories. Numbers are reported on dev split.

Comparing Multimodal GUI Benchmarks

What makes our benchmark unique?

Existing Benchmarks	WARC-Bench (Ours)
ScreenSpot v2 → Single step only	✅ Multi-step subtasks
MiniWob++ → Simple, single widget	✅ Realistic, multi-widget websites
WebArena → Long horizon, 5 envs	✅ 91 envs, easily scalable to more
Online-Mind2Web → Real websites	✅ Sandboxed, reproducible reward

ArXiv: https://arxiv.org/pdf/2510.09872, ICLR 2026

Table 3: WARC-Bench provides unique insights

Model	WARC-Bench (test)	WebArena (no map)	MiniWoB++	ScreenSpot V2
Qwen2.5-VL-7B-Instruct	4.67%	3.07%	12.53%	51.62%
Qwen2.5-VL-72B-Instruct	37.33%	15.68%	53.87%	**88.05%**
GPT-5	51.33%	**34.06%**	52.27%	26.39%
Claude Sonnet 4.0	**64.83%**	**37.96%**	**71.73%**	**85.06%**
Ours-7B (SFT+RLVR)	29.17%	7.31%	36.27%	75.81%
Ours-72B (SFT+RLVR)	**52.33%**	26.80%	**59.20%**	82.44%

- *ScreenSpot*, *MiniWob*++ correlate poorly with complex web navigation ability (*WebArena*), while WARC-Bench tracks it better

ArXiv: https://arxiv.org/pdf/2510.09872, ICLR 2026

Table 4: Task decomposition helps weak planners

Experiment: We compare hierarchical (hier.) agents using a planner and a subtask executor v/s a simple step-by-step agent (SVA) on long-horizon tasks (WA-Lite)

#	Planner Model	Subtask Executor	Agent	WA-Lite SR%
1	N/A	Qwen2.5-VL-72B-Instruct	SVA	16.17 ± 1.14
2	Qwen2.5-VL-72B-Instruct	Qwen2.5-VL-72B-Instruct	Hier. [P, E]	22.39 ± 2.58
3	Qwen2.5-VL-72B-Instruct	**Ours-72B-RLVR**	Hier. [P, E]	**24.63 ± 0.75**
4	N/A	Claude-4.0-Sonnet	SVA	**36.32 ± 1.14**
5	Claude-4.0-Sonnet	Qwen2.5-VL-72B-Instruct	Hier. [P, E]	31.84 ± 0.87
6	Claude-4.0-Sonnet	Claude-4.0-Sonnet	Hier. [P, E]	35.32 ± 1.14
7	Claude-4.0-Sonnet	**Ours-72B-RLVR**	Hier. [P, E]	35.82 ± 1.29

- F1: Weaker models like Qwen2.5-VL-72B benefit from hier. agent design (+8 p.p.)
- F2: Claude Sonnet 4.0 does not benefit from task decomposition. But we can offload subtasks to a capable model to reduce latency with no accuracy loss!

Future Directions

- Use community support to collect more WARC environments and tasks
- Integrate subtask models into hierarchical web-agent frameworks to evaluate benefits
- Explore more RL algorithms and compare (GiGPO, DAPO, GSPO, GRPO etc.)

Please contact the authors for any notes (*sansri264@gmail.com*)
Thank you!