Self-Certainty Guided Test-Time Scaling for Web Agents

Xiangyu Zhang, Jingzhuo Zhou, Xuandong Zhao

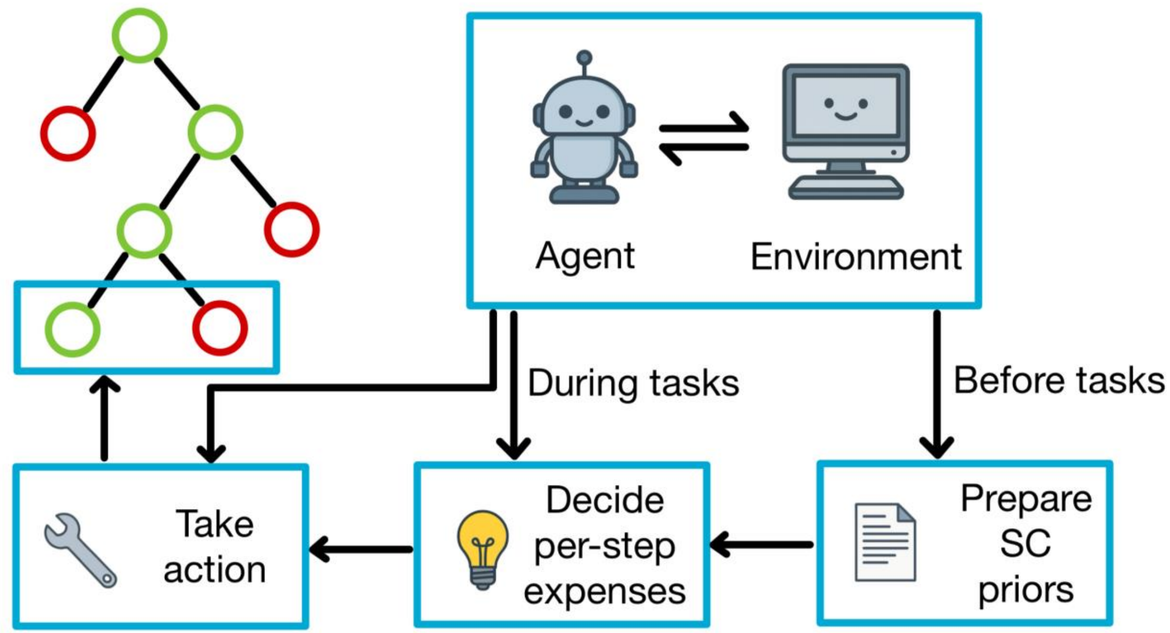
Motivation – Why Adaptive Scaling?

- Previous methods:
  - ReAct
  - ToT
  - Search Agent
  - Reflective MCTS (RMCTS)

- Static inference settings leads to inefficiencies:
  - wasted compute on trivial steps
  - insufficient exploration when the model is uncertain.
- Current agents lack a principled mechanism to adjust computation budgets according to difficulty.

Our Method – Self-Certainty Guided Adaptive Tree Search (SC-ATS)

- Dynamically allocates search budget based on the model's own internal confidence
- SC-ATS modifies the expansion step to condition branching on a self-certainty score.
- This score is derived from the token-level probability distribution of the model's response, allowing the agent to judge how confident it is in its current prediction.
- Scale intelligently—without any external supervision, rewards, or changes to the model weights



Self-Certainty as a Confidence Signal

$$\textbf{Self-certainty} = -\frac{1}{nV}\sum_{i=1}^{n}\sum_{j=1}^{V}\log\left(V \cdot p(j|x, y_{<i})\right)$$

n: the length of the response's length
V: the vocabulary size.

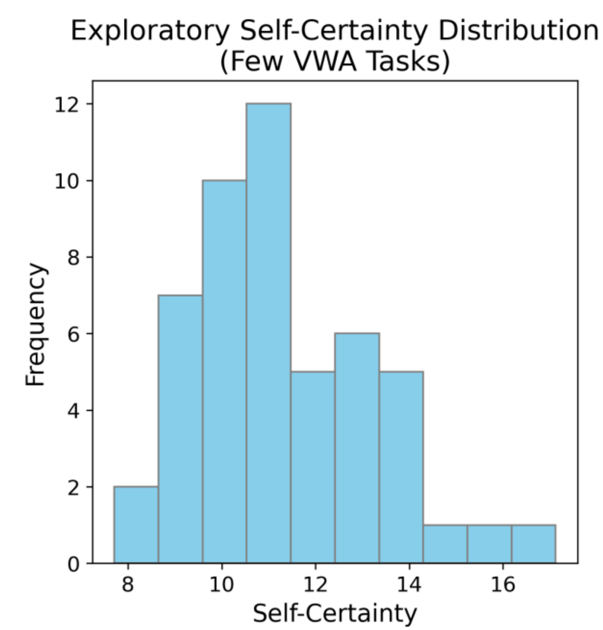
This is equivalent to computing the **cross-entropy** between the **model's predicted token distributions** and **a uniform distribution** over the vocabulary.

[1]: Zhewei Kang, Xuandong Zhao, and Dawn Song. **Scalable Best-of-N Selection for Large Language Models via Self-Certainty.** *arXiv preprint arXiv:2502.18581*, 2025.

- Before Tests: get prior $P(c)$
- During Tests:

$$f : \mathbb{R} \to \mathbb{N}^{+},$$

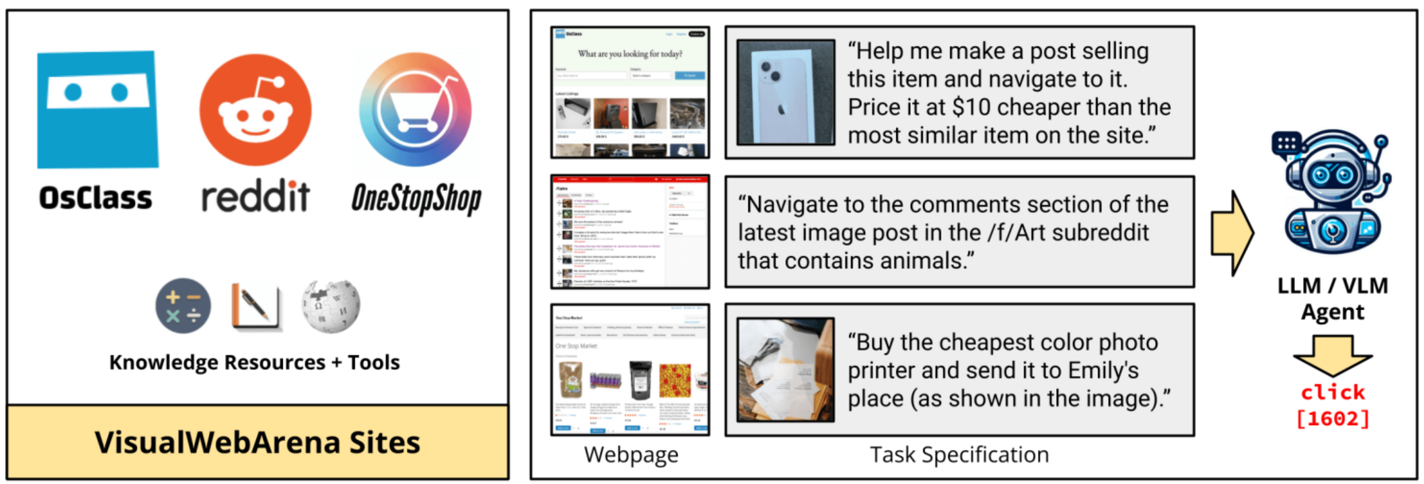
$$b' = f(\hat{c}).$$



Experimental Setup – Benchmarks and Implementation

- 56 tasks from the VisualWebArena
- Model: GPT-4o-mini
- Implementation: based on ExAct

- Top 25% (most confident): $\Delta b = -2$
- 25–50%: $\Delta b = -1$
- 50–75%: $\Delta b = 0$
- 75–100% (least confident): $\Delta b = +1$



$$b' = \max(1, b + \Delta b)$$

[2]: Illustration is from [web-arena-x/visualwebarena: VisualWebArena is a benchmark for multimodal agents.](web-arena-x/visualwebarena)

Results – Improved Efficiency and Success

- Success rate: **3.6% improvement**;
- Prompt token usage: reduced by over **80%**;
- Completion tokens usage: reduced by **70%**.

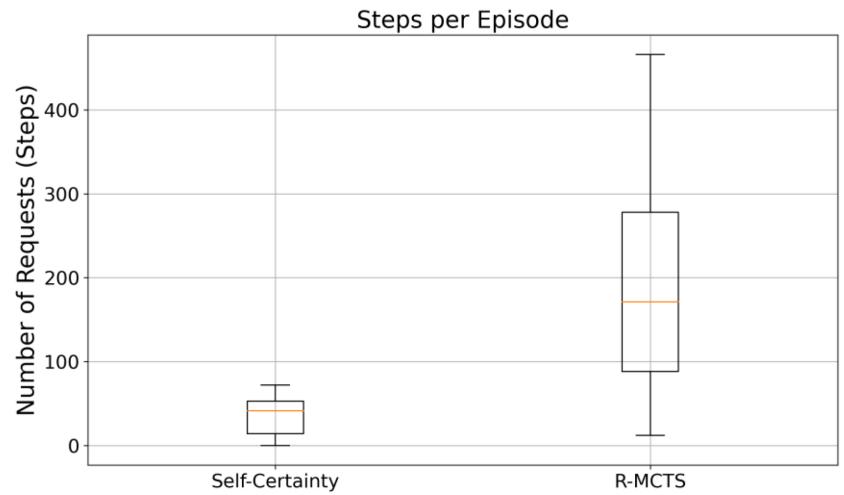


Table 1: Results on VisualWebArena (56 tasks)

Method	Avg. Success Rate ↑	Completion Tokens ↓	Prompt Tokens ↓
Self-Certainty (Ours)	**19.6%**	**33.4K**	**5.5M**
RMCTS (Baseline)	14.3%	285K	27.4M

Conclusion – Lightweight, Scalable Test-Time Adaptation

- **Takeaway**
  - Self-certainty enables task-aware adaptive planning **with low overhead per task**.
  - Our method **outperforms** RMCTS on VisualWebArena while being **cheaper**.
  - Promising direction for *signal-driven* agent control.