Can Agents Generalize to the Open World? Unveiling the Fragility of Static Training in Tool Use

Weiming Wu*, Song-Lin Lv*, Rui Zhu, Zi-Jian Cheng, Lan-Zhe Guo

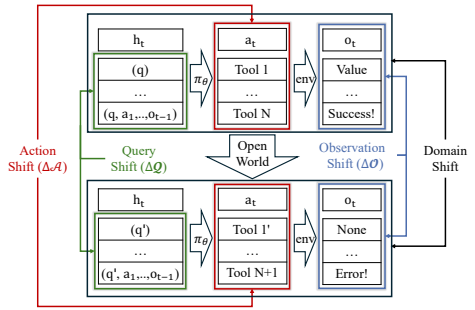
ICML International Conference On Machine Learning

LAMDA Learning And Mining from DatA

OpenAgent Setting

Motivation

- **Closed Environment Bias:** The existing agent evaluation benchmarks assume that the test environment is closed and static, while shifts in the real world may make the agent very vulnerable when facing distribution changes.
- **Unknown Boundaries:** There is no controllable framework to systematically display the performance of different fine-tuning models in the open environment.



▲ The possible shifts during the operation of Agents.

OpenAgent Setting

Interactive Agent $M = < \mathcal{Q}, \mathcal{A}, \mathcal{O}, \pi >$ faces compounding, cascading shifts along agent process across four dimensions:

- **Intent Shift in Query Space ($\Delta\mathcal{Q}$):** $\mathbf{P}_{\text{train}}(\mathcal{Q}) \neq \mathbf{P}_{\text{test}}(\mathcal{Q})$. Initial query may misinterpretations propagate and compound errors across subsequent trajectory steps.
- **Structural Shift in Action Space ($\Delta\mathcal{A}$):** $\mathcal{A}_{train} \neq \mathcal{A}_{test}$. Non-stationary tool spaces involving surface drift (renaming), semantic conflict (altered behavior), or structural reconfiguration.
- **Dynamics Shift in Observation Space ($\Delta\mathcal{O}$):** $\mathcal{O}_{train} \neq \mathcal{O}_{test}$.The feedback channel encounters novel return formats, unexpected errors, or null values.
- **Compositional Domain Shift ($\Delta\mathcal{D}$):** $(\mathcal{Q}, \mathcal{A}, \mathcal{O})_{\text{train}} \rightarrow (\mathcal{Q}, \mathcal{A}, \mathcal{O})_{test}$. All elements shift jointly into a new domain while preserving the latent problem-solving structure G. The agent must transfer the underlying reasoning topology over surface patterns.

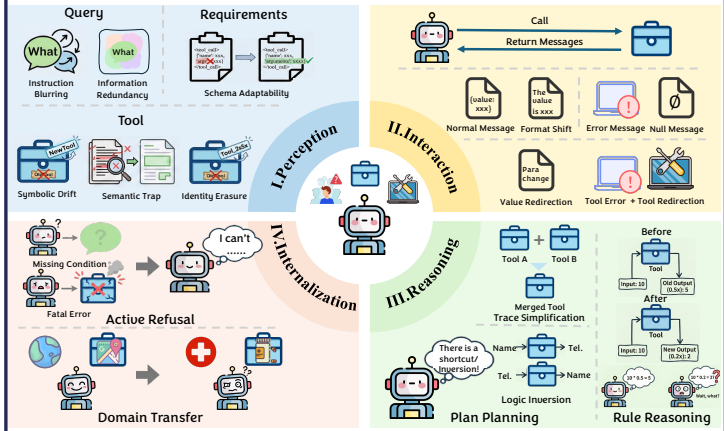
OpenAgent Evaluation Tiers

I. Perception Generalization: The first tier evaluates whether the agent understands real intents of queries and tool functions despite surface changes.

II. Interaction Generalization: The second tier checks whether the agent can understand feedback and adjust actions during unexpected feedbacks.

III. Reasoning Generalization: The third tier assesses whether the agent can think logically and make new plans instead of following train steps.

IV. Internalization Generalization: The last tier tests whether the agent grasps core problem-solving logic to know its ability limits and handle new domain situations.

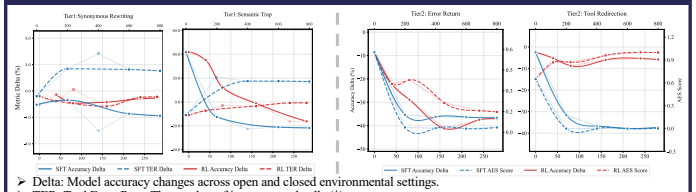


Implementation

We build a Python sandbox to simulate controlled tool interactions. Models are then trained via **Supervised Fine-tuning (SFT)** and **Reinforcement Learning (RL)**, and their generalization abilities are finally evaluated on shifted test sets.

- **Close Environment**: Refers to standard setups where training and test sets share identical query distributions, unchanged toolsets, and error-free execution environments.
- **Open Environments:** Reflects dynamic real-world scenarios featuring ambiguous queries, changed tool names, abnormal feedback, modified tool logic, and even complete domain shifts.

Observations



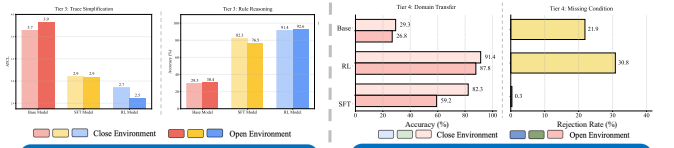
- Delta: Model accuracy changes across open and closed environmental settings.
- TER: Tool Error Rate. The number of incorrect tool calls (↓).
- AEC: Active Exploration Score. An indicator for measuring the error correction capability of a model (↑).

Tier 1: Tool Perception

RL achieves semantic grounding through interaction while SFT relies on brittle symbolic anchoring that degrades as fitting increases.

Tier 2: Interaction with Guidance

RL leverages explicit guidance for dynamic policy adaptation, while SFT exhibits trajectory inertia and often hallucinates outcomes under corrective feedback.



Tier 3: Reasoning Generalization

RL exhibits superior adaptability: it can flexibly invoke new tools to streamline the average tool-calling Length (ATCL) and demonstrates stronger reasoning capabilities when faced with updated tool usage rules.

Tier 4: Boundary Awareness

Both paradigms struggle with boundary awareness in unsolvable states, but exhibit distinct failure patterns: SFT often fails to perceive fatal feedback, whereas RL perceives the failure but still favors forced completion.

Solution

Model	Tier-1 Acc Δ↑	Tier-2 Acc Δ↑	Tier-3 Acc Δ↑	Tier-4 RR↑
Performance across training stages				
Base	-29.8	-8.5	-8.5	12.2
SFT-200	-67.7	-48.2	-39.9	0.3
+ PAFT	**+28.6**	**+26.5**	**+22.7**	**99.3**
SFT-400	-53.9	-45.4	-32.5	0.0
+ PAFT	**+5.6**	**+4.9**	-2.8	**97.8**
SFT-600	-51.3	-46.4	-33.0	0.1
+ PAFT	-2.5	-2.9	-10.7	**99.6**
SFT-800	-50.4	-45.3	-28.0	0.2
+ PAFT	-4.1	-5.3	-9.8	**99.6**

We propose the **Perturbation-Augmented Fine-Tuning (PAFT)** for SFT model, that performs poorly in open environment. The core idea of PAFT is to inject controlled perturbations at the trajectory level, helping the model learn to reason in abnormal and dynamic environments. For example:

$$\tau_{\text{orig}} = \{a_i, o_i\} \xrightarrow{\mathcal{G}_{env}} \tau' = \{a_i, o_{\text{change}}, a'_i, o_i\}$$

The Table show that PAFT successfully reduces performance drops when facing various open-environment shifts and effectively restoring the agent's robustness and active refusal abilities in complex environments.

✓ Weiming Wu (wuwm23@smail.nju.edu.cn)
✓ Song-Lin Lv (lvsl@lamda.nju.edu.cn, job market candidate).

Project Page: **Code:** **WeChat:**