AGENT WORKFLOW MEMORY

Presenter: Ziqian Huang, Lingyu Kong

Problem

Current LM-based agents:

- Solve each task independently
- Rely on primitive actions (CLICK, TYPE, SCROLL, …)
- Do not extract reusable routines
- Struggle with long-horizon, compositional tasks

Core Limitation:

No mechanism to abstract and reuse workflows across tasks.

Key Question:

Can agents automatically induce reusable workflows from past successful trajectories?

Motivation

Real-world digital tasks are:

- Long-horizon
- Compositional
- Repetitive across domains

Examples:

- Booking flights
- Searching & filtering products
- Filling forms
- Navigating admin dashboards

Without reusable workflows:

- Poor generalization
- Inefficient exploration
- No improvement over time

Desired Capability:

Agents that accumulate and reuse skills across tasks.

Why current agents fail? (Prior Works)

In-context learning / Retrieved Relevant Examples

(e.g., Synapse, MindAct)

- Retrieve full example trajectories
- Follow similar past tasks
- Strong dependence on example similarity

❌ They reuse *instances*, not *abstracted skills*.

Human-Engineered Workflows

(e.g., SteP)

- Manually written hierarchical policies
- Domain-specific design

❌ Require human supervision and domain knowledge.

Rule-based Subroutine Extraction

(e.g., Ellis et al., 2023)

- Extract frequently occurring action/program patterns
- Deduplicate identical sequences
- Add them as reusable library functions
- Improve future search efficiency

❌ Limited to structural pattern matching;

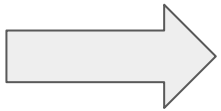
no semantic abstraction across tasks.

What Is a Workflow?

“Find a Hilton hotel near this location and show walking directions.”

Concrete trajectory:

- Search for location
- Click nearby hotels
- Filter by name
- Show route



Abstracted workflow:

- Search for {entity}
- Navigate to relevant section
- Apply filter {criteria}
- Display {output}

A workflow abstracts reusable subroutines from concrete trajectories.

AWM Core Idea

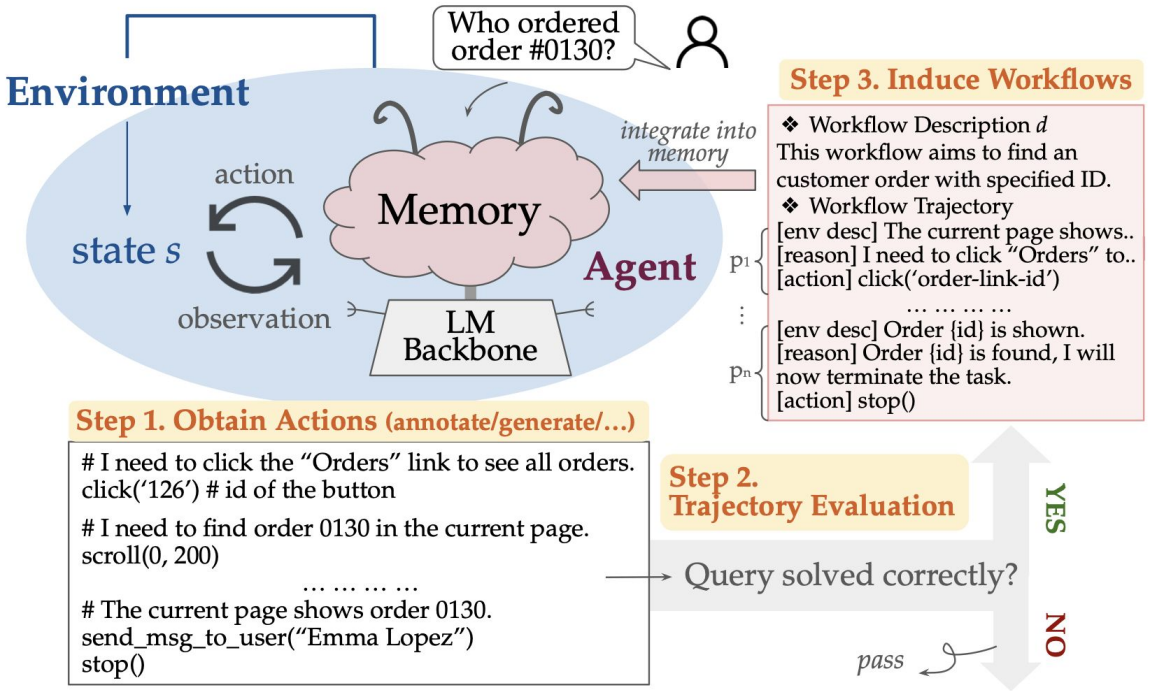
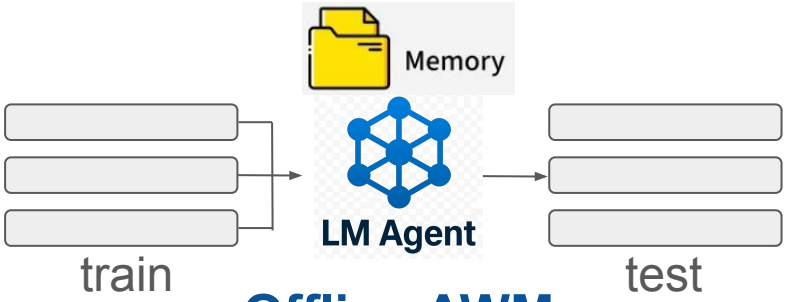


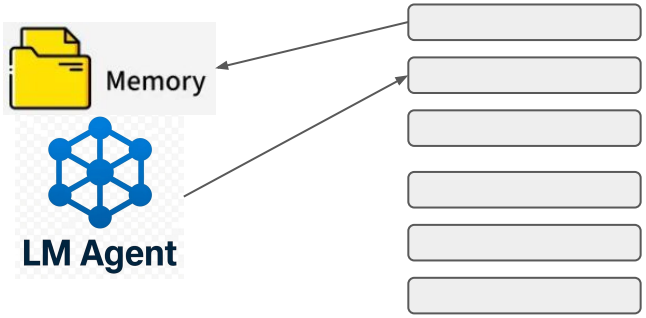
Figure 2: Illustration of our AWM pipeline: an agent takes actions to solve given queries, induces workflows from successful ones, and integrates them into memory.

Offline vs Online AWM



Offline AWM

Training Data
⇩
Extract successful trajectories
⇩
Induce workflows
⇩
Store in memory
⇩
Freeze memory
⇩
Test-time retrieval only



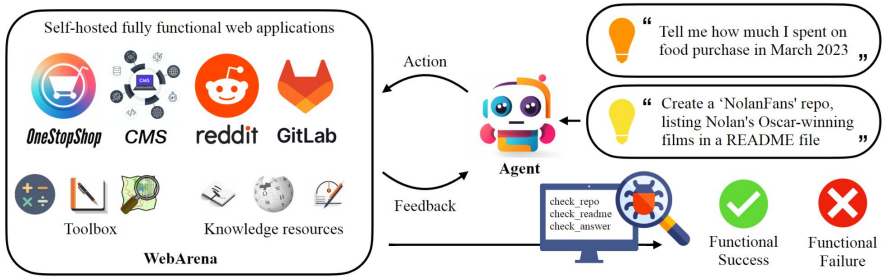
Online AWM

Task 1
⇩
If success ⇨ Induce workflow
⇩
Add to memory

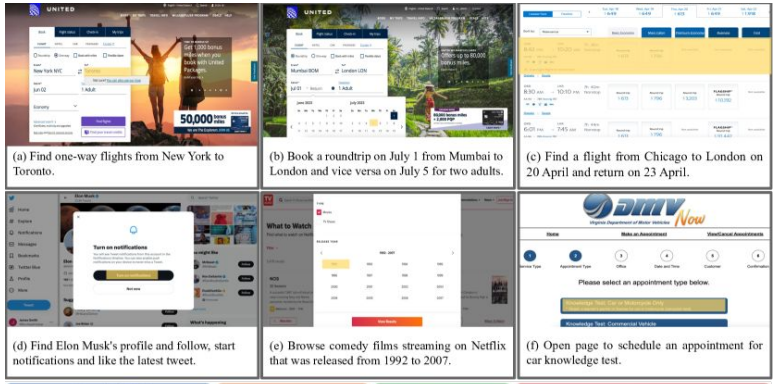
Task 2
⇩
Induce again

Task 3
⇩
…

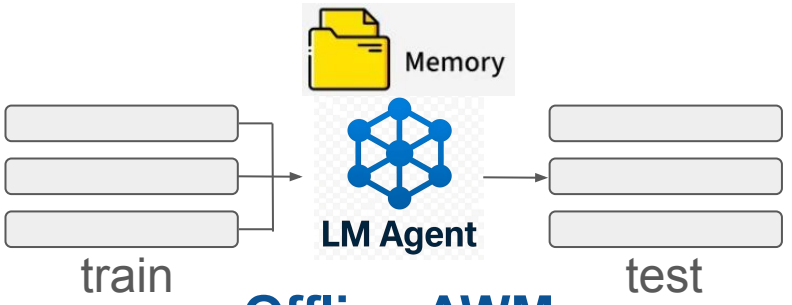
Experiment Results - What to Watch For



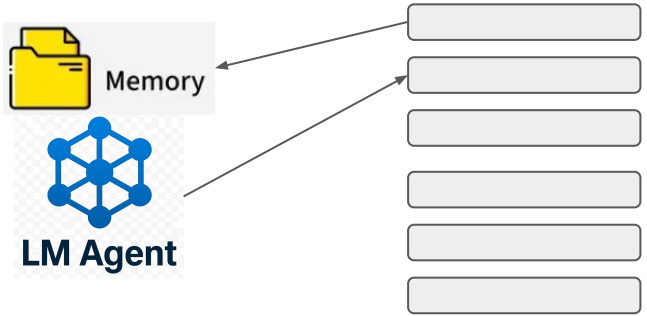
WebArena



Mind2Web



Offline AWM



Online AWM

Experiment Results - WebArena (Online)

WebArena: naïve LM-agent baseline
AutoEval: LM-agent with LM-evaluator for action refinement
BrowserGym: SOTA LM-agent without domain knowledge
SteP: LM-agent with 14 selected workflows

Table 1: Task success rate on WebArena using `gpt-4`, and score breakdown on five website splits.

Method	Total SR	Shopping	CMS	Reddit	GitLab	Maps	# Steps
With human engineered workflows							
*SteP (Sodhi et al., 2023)	33.0	**37.0**	24.0	**59.0**	**32.0**	30.0	-
Autonomous agent only							
WebArena (Zhou et al., 2024)	14.9	14.0	11.0	6.0	15.0	16.0	-
AutoEval (Pan et al., 2024)	20.2	25.5	18.1	25.4	28.6	31.9	46.7
BrowserGym (Drouin et al., 2024)	23.5	-	-	-	-	-	-
BrowserGym$_{ax-tree}$	15.0	17.2	14.8	20.2	19.0	25.5	7.9
AWM (OURS)	35.5	30.8	29.1	50.9	31.8	**43.3**	**5.9**

Experiment Results - WebArena (Online)

Grouping examples by templates and randomly choose one from each group. Example: search for the location of GT/Emory

Table 2: Task success rate on the cross-template subset of WebArena, as well as the result breakdown on each website split. We mark the number of examples for each website split under the name.

Method	Total SR	Shopping (51)	CMS (45)	Reddit (24)	GitLab (45)	Maps (32)
With human engineered workflows						
*SteP (Sodhi et al., 2023)	32.1	**26.5**	**29.3**	**52.2**	27.3	36.4
Autonomous agent only						
AutoEval (Pan et al., 2024)	23.2	12.2	17.1	21.7	**31.8**	36.4
BrowserGym$_{ax-tree}$	20.5	10.4	17.8	23.1	27.3	28.6
AWM (OURS)	**33.2**	24.5	**29.3**	**52.2**	**31.8**	**39.4**
AWM (OURS)	35.5	30.8	29.1	50.9	31.8	**43.3**

Experiment Results - WebArena (Online)

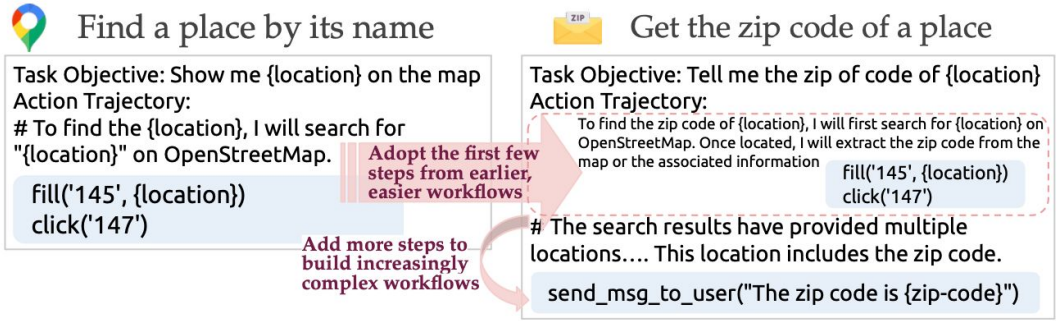


Figure 6: AWM builds increasingly complex workflows over time, by learning from past examples and earlier workflows.

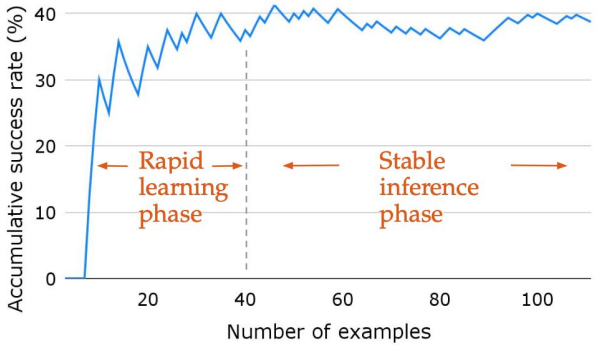


Figure 5: AWM enables rapid learning from a small amount of data, i.e., about 40 queries, using WebArena map test split as an example.

Experiment Results - Mind2Web

- **AWM** has better Elem Acc, Step SR, and SR
- **MindAct** has better Action F1 (**AWM** biased by workflows in memory)

Table 3: AWM offline results on Mind2Web cross-task dataset. Elem Acc and SR are short for element accuracy and success rate. We footnote the GPT variant used by each method, `3.5` and `4` stands for `gpt-3.5-turbo` and `gpt-4`, respectively. We highlight the best result within the same model variant.

Method	Elem Acc	Action F_1	Step SR	SR
$\text{MindAct}_{3.5}$	20.3	**56.6**	17.4	0.8
$\text{CogAgent}_{3.5}$	-	-	18.6	-
$\text{Synapse}_{3.5}$	34.0	-	30.6	2.4
$\text{AWM}_{3.5}$	**39.0**	52.8	**34.6**	**2.8**
MindAct_{4}	41.6	**60.6**	36.2	2.0
AWM_{4}	**50.6**	57.3	**45.1**	**4.8**

Experiment Results - Mind2Web

1. Same Task: search for keyboard/screen on Amazon
2. Same Website: search for keyboard/buy a screen on Amazon
3. Same Domain: search for keyboard on eBay/buy a screen on Amazon

Cross-X: trained on same X but tested on other X

Table 4: Success rate on Mind2Web cross-task, cross-website, and cross-domain generalization test, using `gpt-4` model. EA is short for element accuracy and AF_1 is short for action F_1.

Method	*Cross-Task*				*Cross-Website*				*Cross-Domain*			
	EA	AF_1	Step SR	SR	EA	AF_1	Step SR	SR	EA	AF_1	Step SR	SR
MindAct*	41.6	**60.6**	36.2	2.0	35.8	**51.1**	30.1	2.0	21.6	**52.8**	18.6	1.0
$AWM_{offline}$	**50.6**	57.3	**45.1**	**4.8**	41.4	46.2	33.7	**2.3**	36.4	41.6	32.6	0.7
AWM_{online}	50.0	56.4	43.6	4.0	**42.1**	45.1	**33.9**	1.6	**40.9**	46.3	**35.5**	**1.7**

Online AWM seems more robust under large distribution shift

Experiment Results - Workflow Representations

Rule vs **LM** based workflow instruction

- **Rule**: CLICK → TYPE → CLICK
- **LM**: Navigate to settings → Fill in the blank

Table 6: AWM results with different workflow induction methods on Mind2Web cross-task dataset.

Method	Elem Acc	Action F_1	Step SR	SR
MindAct_4	41.6	**60.6**	36.2	2.0
$\text{AWM}_{4,rule}$	49.5	57.0	43.4	2.0
$\text{AWM}_{4,lm}$	**50.6**	57.3	**45.1**	**4.8**

Experiment Results - Workflow Representations

Rule vs **LM** based workflow instruction

- **Rule**: CLICK → TYPE → CLICK
- **LM**: Navigate to settings → Fill in the blank

Table 6: AWM results with different workflow induction methods on Mind2Web cross-task dataset.

Method	Elem Acc	Action F_1	Step SR	SR
MindAct_4	41.6	**60.6**	36.2	2.0
$\text{AWM}_{4,rule}$	49.5	57.0	43.4	2.0
$\text{AWM}_{4,lm}$	**50.6**	57.3	**45.1**	**4.8**

Experiment Results - Workflow Representations

Code vs **Text** based workflow representations

- **Code**: CLICK({button-id}) → TYPE({field-id}, text)
- **Text**: Click the OK button

Table 7: Mind2Web cross-task results with AWM using code and text workflows.

Method	Elem Acc	Action F_1	Step SR	SR
MindAct	41.6	**60.6**	36.2	2.0
AWM	50.6	57.3	45.1	**4.8**
AWM_{text}	**51.2**	57.4	**45.4**	3.6

Experiment Results - Workflow Representations

Integrate workflows into high-level actions

- **Primitive Actions:** CLICK, TYPE…
- **High-Level Actions:** Find_Place_in_GoogleMap

Table 9: Mind2Web results with AWM_{AS} variant that alters the action space besides memory augmentation. All methods use `gpt-4`.

Method	Elem Acc	Action F_1	Step SR	SR
MindAct	41.6	**60.6**	36.2	2.0
AWM	50.6	57.3	45.1	**4.8**
AWM_{AS}	**51.8**	56.7	**46.4**	3.6

Experiment Results - Environment Representations

Desc vs **HTML** based environment representations

- **Desc**: natural language description
- **HTML**: filtered HTML code

Table 8: Mind2Web results using GPT-3.5-turbo with different environment representations.

Desc.	HTML	Elem Acc	Act F_1	Step SR	SR
✓	✗	**39.0**	52.8	**34.6**	**2.8**
✗	✓	38.1	**54.0**	33.8	**2.8**
✓	✓	37.1	51.3	32.9	2.0

Takeaways and Discussions

Some Takeaways:

- LM summarized and injected workflows can improve LM-agents' performance on long-horizon tasks
- High-level macro-actions may become clumsy, while combinable small actions can be flexible.
- Online induced workflows can better generalize to shifted domains

Takeaways and Discussions

Discussions and Concerns:

- How to determine the granularity of the the workflows and actions
  - Flexible vs informative
- How to identify and remove the workflows that are polluted or outdated
  - Effective volume of the memory?