LLMs and Tools
Part-3: Agentic Workflow

Advanced Large Language Models

ELL8299 · AIL861 · ELL881

Dinesh Raghu
Senior Researcher & Manager, IBM Research

LLMs and Tools

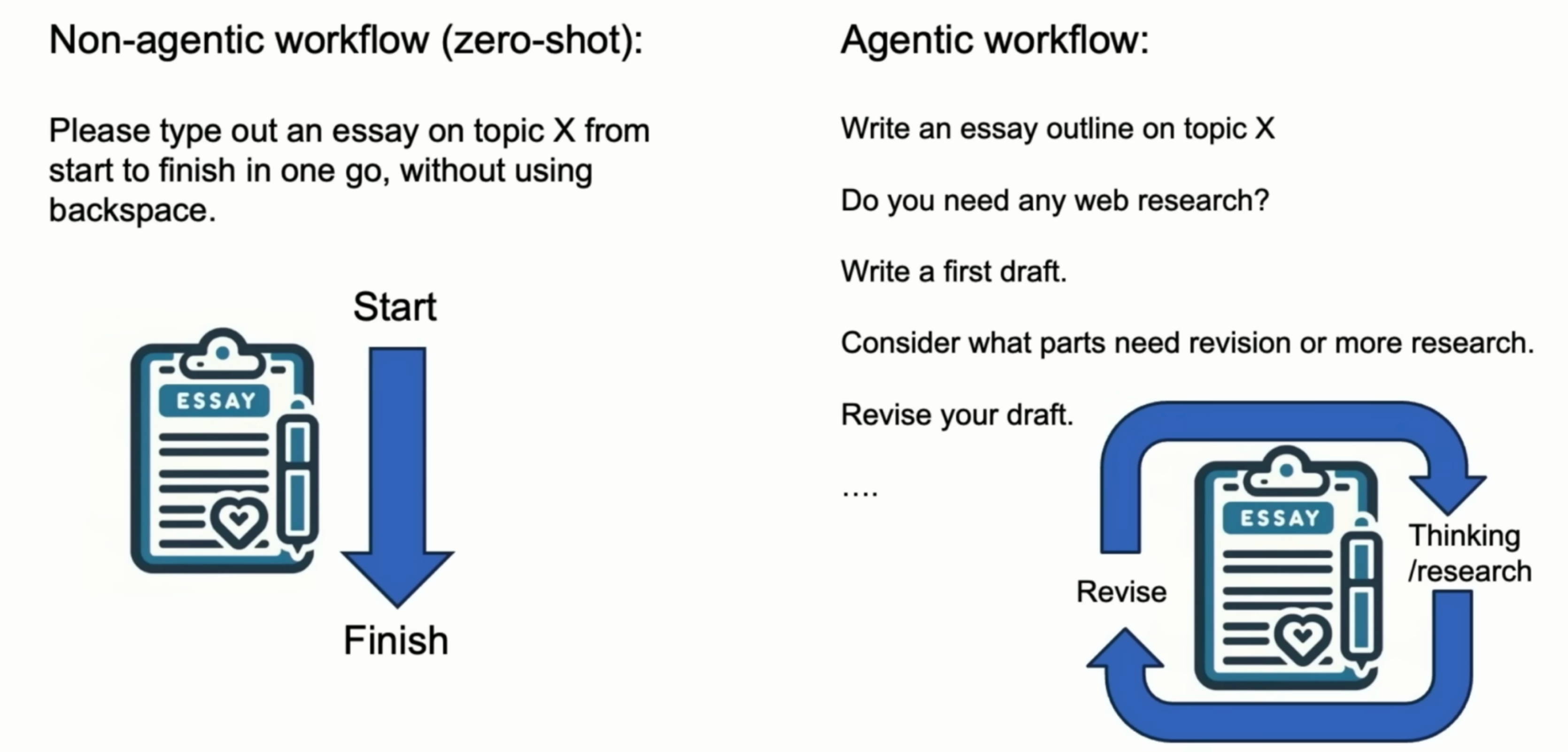
Part 1: Incorporating Tools during Fine-tuning (Tool Augmentation)

Part 2: Teaching LLMs to Use External APIs (Function Calling)

Part 3: Automating Complex Tasks (AI Agents)

Agentic Workflows

Agentic vs Non-Agentic Workflows



Screenshot from [What's next for AI agentic workflows ft. Andrew Ng of AI Fund]

Agentic vs Non-Agentic Workflows

```python
def incr_list(l: list):
    """Return list with elements incremented by 1.
    >>> incr_list([1, 2, 3])
    [2, 3, 4]
    >>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])
    [6, 4, 6, 3, 4, 4, 10, 1, 124]
    """
    return [i + 1 for i in l]
```

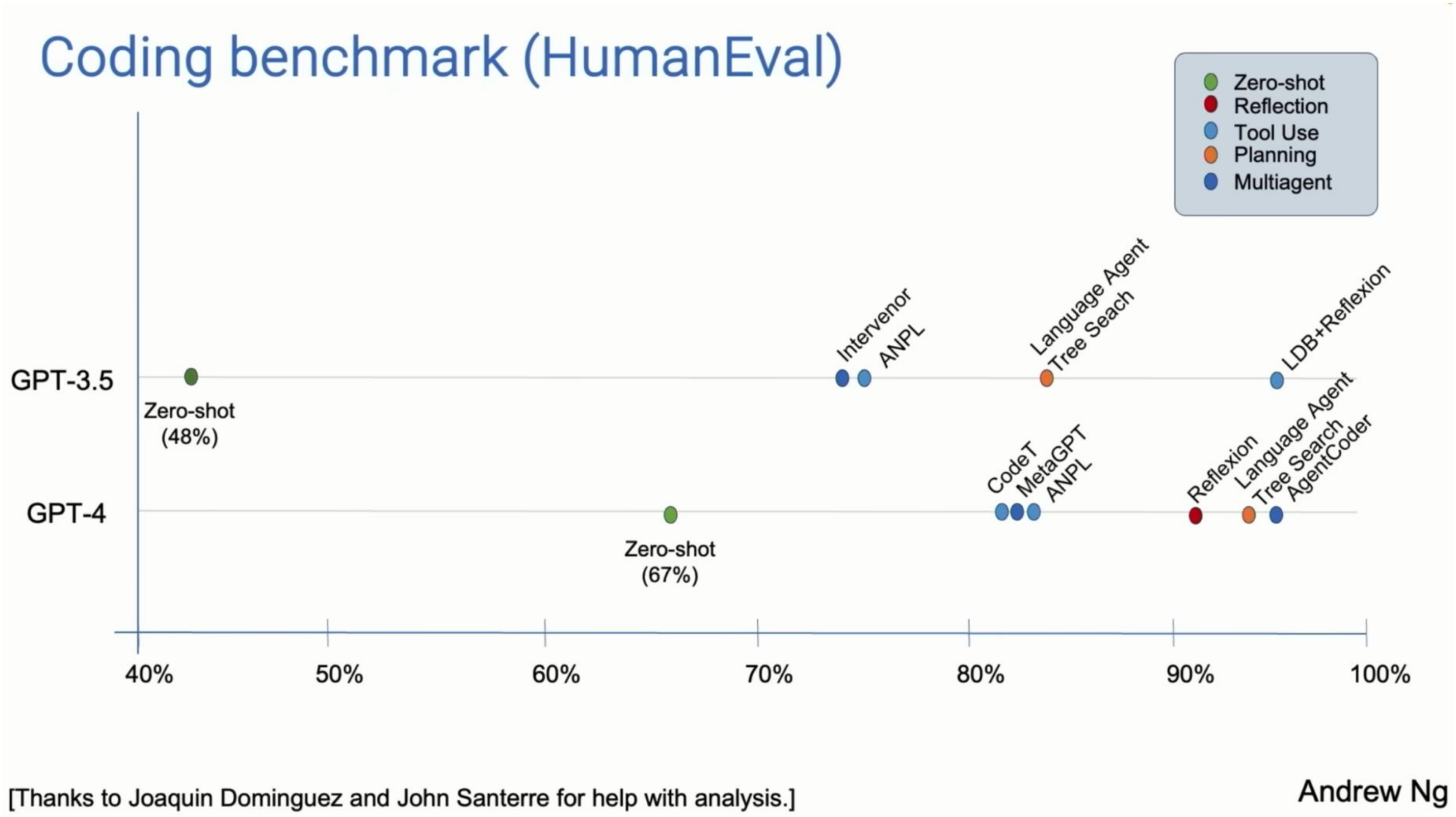
```python
def solution(lst):
    """Given a non-empty list of integers, return the sum of all of the odd elements
    that are in even positions.

    Examples
    solution([5, 8, 7, 1]) ==>12
    solution([3, 3, 3, 3, 3]) ==>9
    solution([30, 13, 24, 321]) ==>0
    """
    return sum(lst[i] for i in range(0,len(lst)) if i % 2 == 0 and lst[i] % 2 == 1)
```

Examples from HumanEval* Dataset

Input to the model is shown with a white background, and a successful model-generated completion is shown in a yellow background.

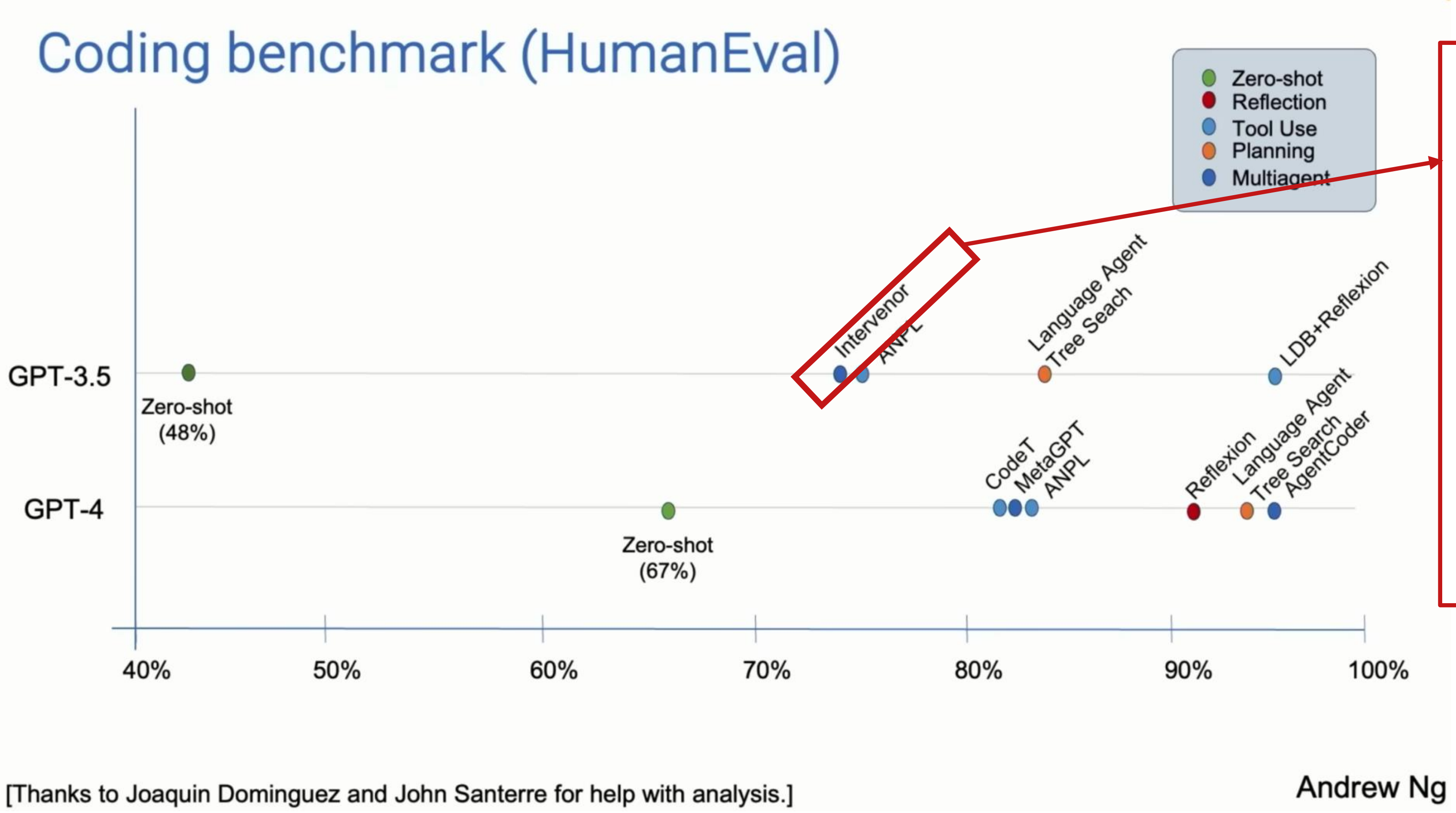
*Evaluating Large Language Models Trained on Code, Chen et. al., 2021

Agentic vs Non-Agentic Workflows



Screenshot from [What's next for AI agentic workflows ft. Andrew Ng of AI Fund]

Agentic vs Non-Agentic Workflows



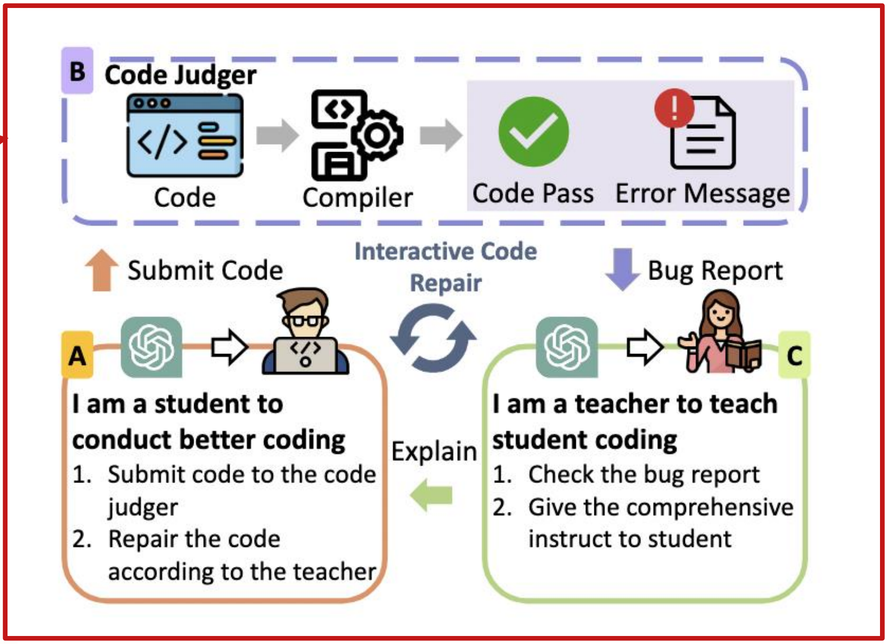


Image Credits: INTERVENOR : Prompting the Coding Ability of Large Language Models with the Interactive Chain of Repair, Wang et al., 2024

Outline

- ReACT
- Self-Refine
- Reflexion

ReAct

Example from HotPotQA
[Yang et. Al., 2018]

Q: To what team was the 2014 NBA Rookie of the Year traded in October 2016?

A: Chicago Bulls

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReAct

Example from HotPotQA
[Yang et. Al., 2018]

Paragraph B: Michael Carter-Williams
Michael Carter-Williams (born October 10, 1991) is an American professional basketball player for the Charlotte Hornets of the National Basketball Association (NBA). He was drafted 11th overall in the 2013 NBA draft by the Philadelphia 76ers, after playing college basketball for the Syracuse Orange. He was named NBA Rookie of the Year in 2014, and has also played for the Milwaukee Bucks and Chicago Bulls.

Q: To what team was the 2014 NBA Rookie of the Year traded in October 2016?

A: Chicago Bulls

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReAct

Example from HotPotQA [Yang et. Al., 2018]

Paragraph A: 2016–17 Chicago Bulls season

The 2016–17 Chicago Bulls season was the 51st season of the franchise in the National Basketball Association (NBA). Former 2011 MVP and 2009 Rookie of the Year Derrick Rose was traded from his hometown team to the New York Knicks. This was the first season without Rose since the 2012–13 season. Dwyane Wade, who played for the Miami Heat from 2003 to 2016, decided to leave the Heat and sign with his hometown team, the Chicago Bulls. The Bulls traded Tony Snell to the Milwaukee Bucks for Michael Carter-Williams on October 17, 2016.

Paragraph B: Michael Carter-Williams

Michael Carter-Williams (born October 10, 1991) is an American professional basketball player for the Charlotte Hornets of the National Basketball Association (NBA). He was drafted 11th overall in the 2013 NBA draft by the Philadelphia 76ers, after playing college basketball for the Syracuse Orange. He was named NBA Rookie of the Year in 2014, and has also played for the Milwaukee Bucks and Chicago Bulls.

Q: To what team was the 2014 NBA Rookie of the Year traded in October 2016?

A: Chicago Bulls

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReAct

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod X

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReAct

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod ✗

(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch ✗

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReAct

What if the model has access to a Search tool for searching over a corpus of documents?

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod ✗

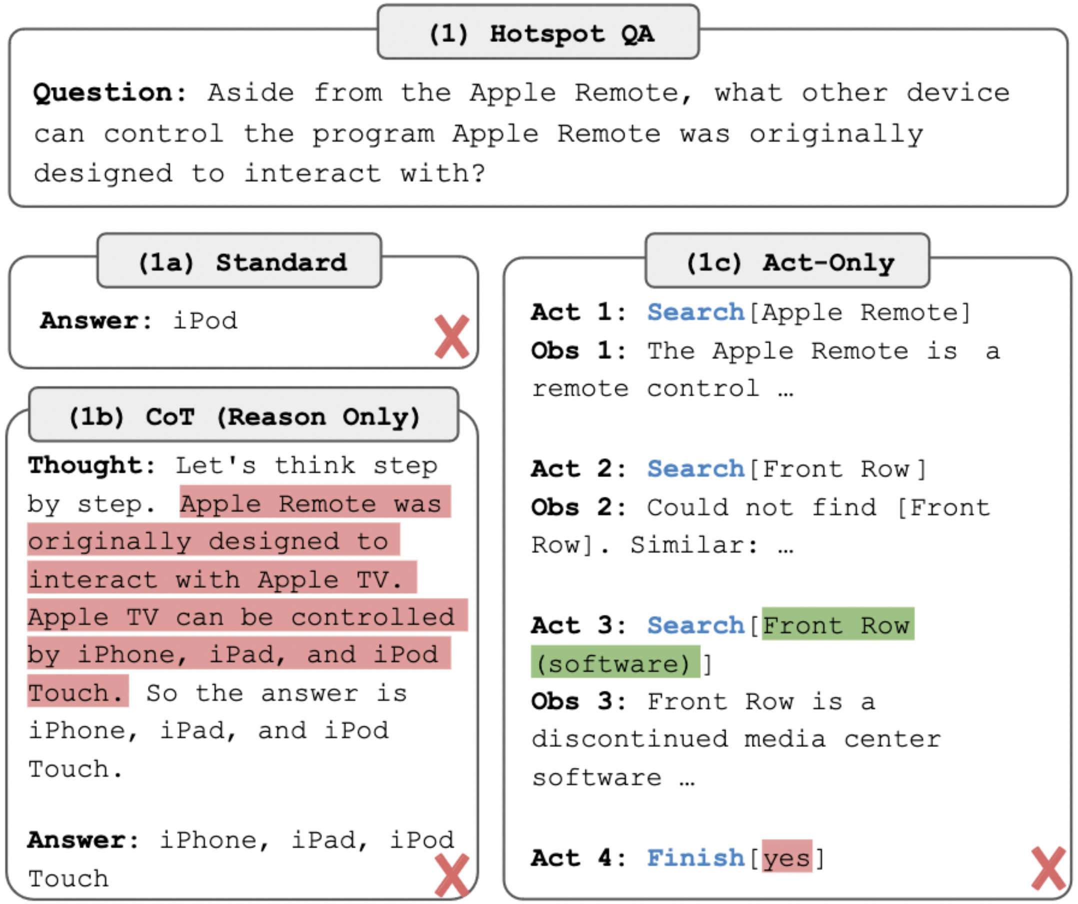
(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch ✗

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

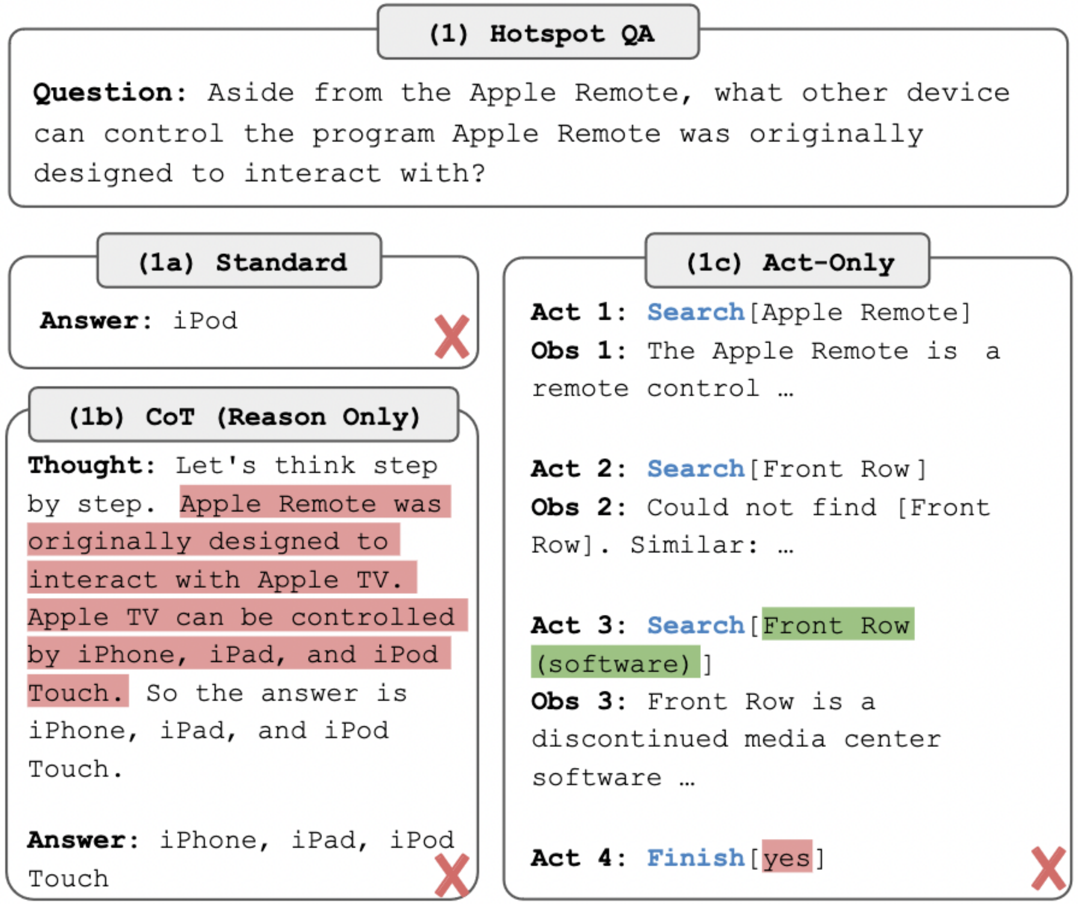
ReAct



What if the model has access to a Search tool for searching over a corpus of documents?

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReACT



Consider an agent interacting with the environment.

`Act:`

- Define action space
  - Search [query]
  - Lookup [keyword]
  - Finish [answer]
- Sample an action based on the question (and the trajectory so far)

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReACT

(1) Hotspot QA

Question: Aside from the Apple Remote, what other device can control the program Apple Remote was originally designed to interact with?

(1a) Standard

Answer: iPod ✗

(1b) CoT (Reason Only)

Thought: Let's think step by step. Apple Remote was originally designed to interact with Apple TV. Apple TV can be controlled by iPhone, iPad, and iPod Touch. So the answer is iPhone, iPad, and iPod Touch.

Answer: iPhone, iPad, iPod Touch ✗

(1c) Act-Only

Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control …

Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: …

Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Act 4: Finish[yes] ✗

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys] ✓

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

Fever

FEVER Prompts

Original	Determine if there is Observation that SUPPORTS or REFUTES a Claim, or if there is NOT ENOUGH INFORMATION.
Claim Answer	Nikolaj Coster-Waldau worked with the Fox Broadcasting Company. SUPPORTS
Claim Answer	Stranger Things is set in Bloomington, Indiana. REFUTES
Claim Answer	Beautiful reached number two on the Billboard Hot 100 in 2003. NOT ENOUGH INFO

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

ReACT

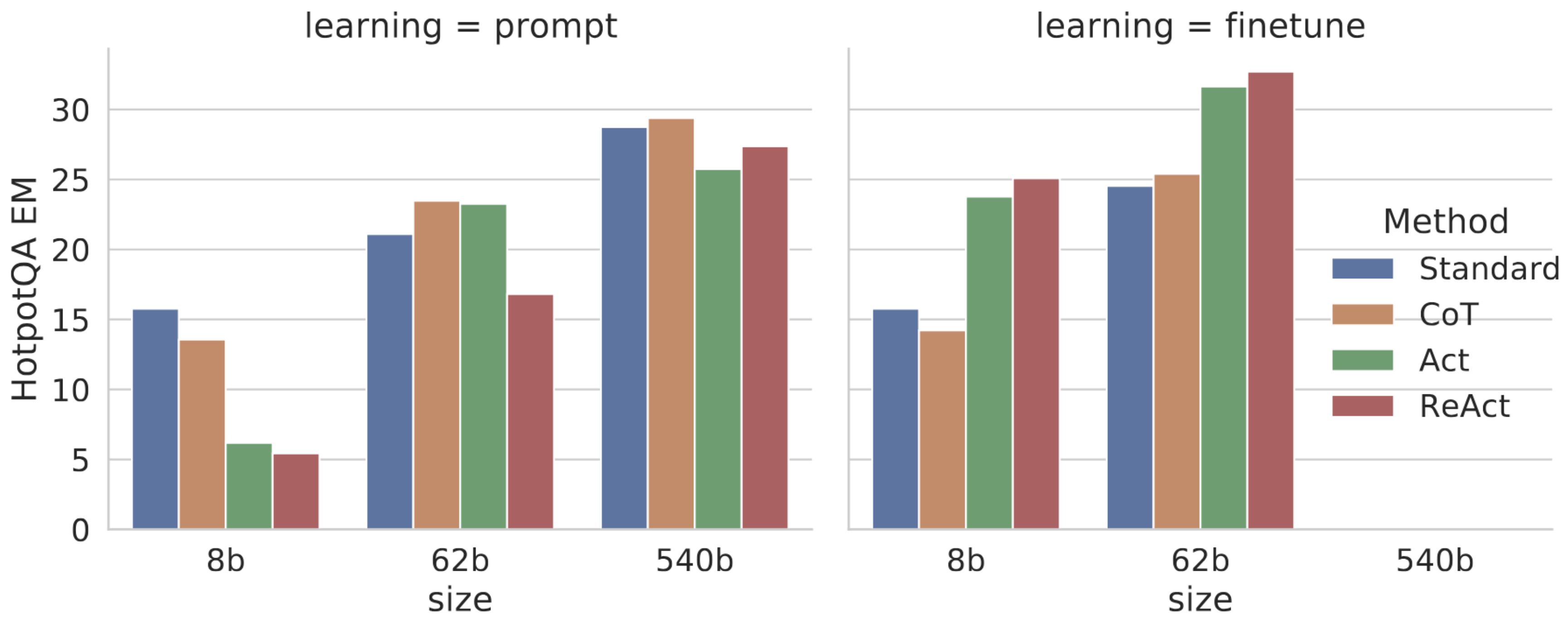
Prompt Method[a]	HotpotQA (EM)	Fever (Acc)
Standard	28.7	57.1
CoT (Wei et al., 2022)	29.4	56.3
CoT-SC (Wang et al., 2022a)	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT-SC → ReAct	34.2	**64.6**
ReAct→ CoT-SC	**35.1**	62.0
Supervised SoTA[b]	67.5	89.5

PaLM-540B prompting results on HotpotQA and Fever

- `ReAct` is better than `Act`
- `CoT` responses contain more hallucinations that `ReAct`
- ReAct makes more reasoning errors than `CoT`
  - repetitively generates the previous though and actions
- `ReAct`'s success is dependent on the search tool

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

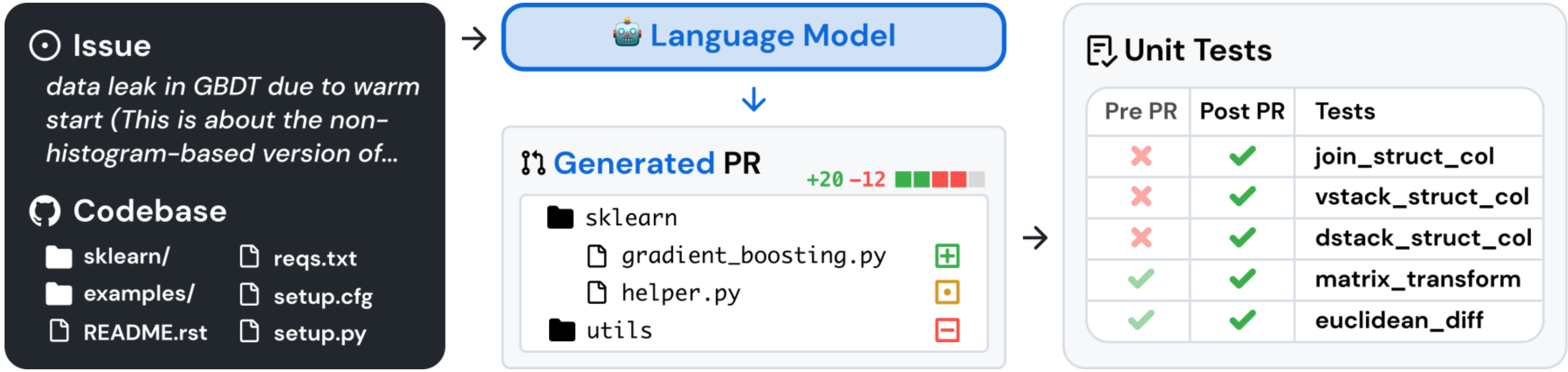
ReACT



Scaling results for prompting and finetuning on HotPotQA with ReAct and baselines

* ReAct: Synergizing Reasoning and Acting in Language Models, Yao et. al., Mar 2023

Agentic Workflow Example: SWE-Bench



1. Benchmark contains 2,294 task instances from 12 different repositories of popular Python packages

2. Supports automated testing based on test cases
   - each issue has associated tests
   - associated tests that originally failed must pass after the patch
   - all other tests must still pass

Image credits: http://www.swebench.com

SWE-Agent uses ReACT

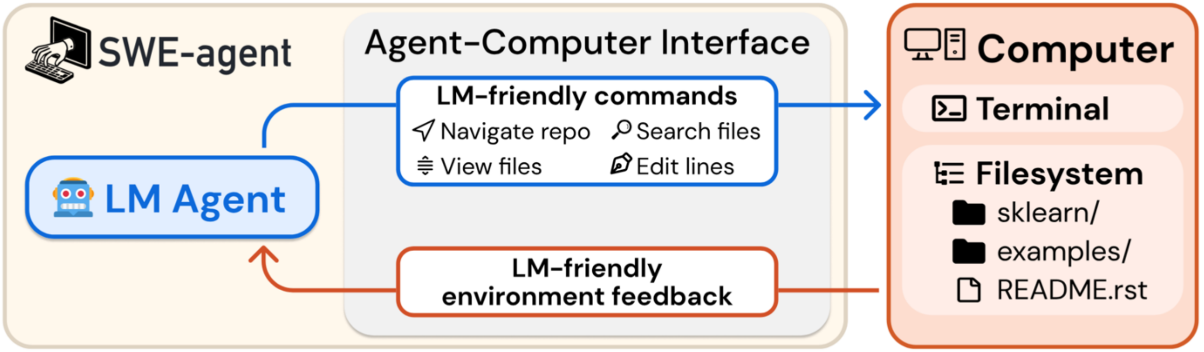
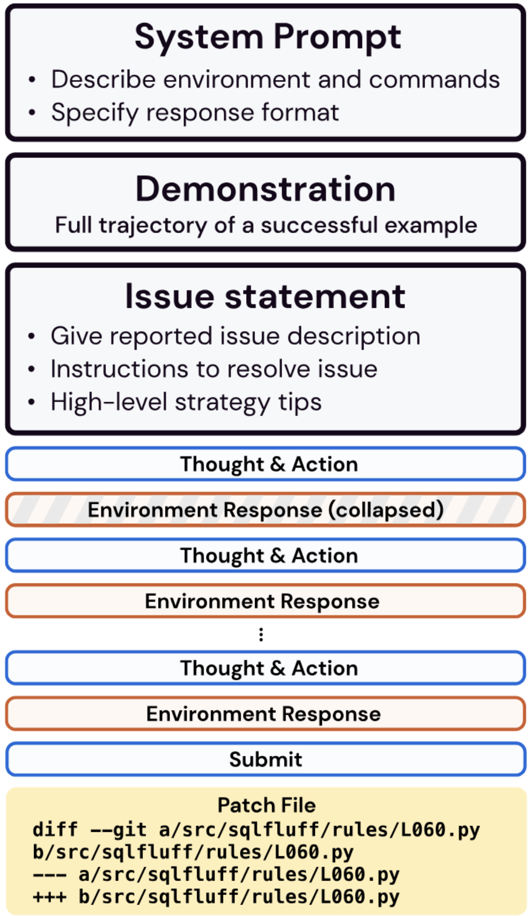


Image credits: http://www.swebench.com; SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024

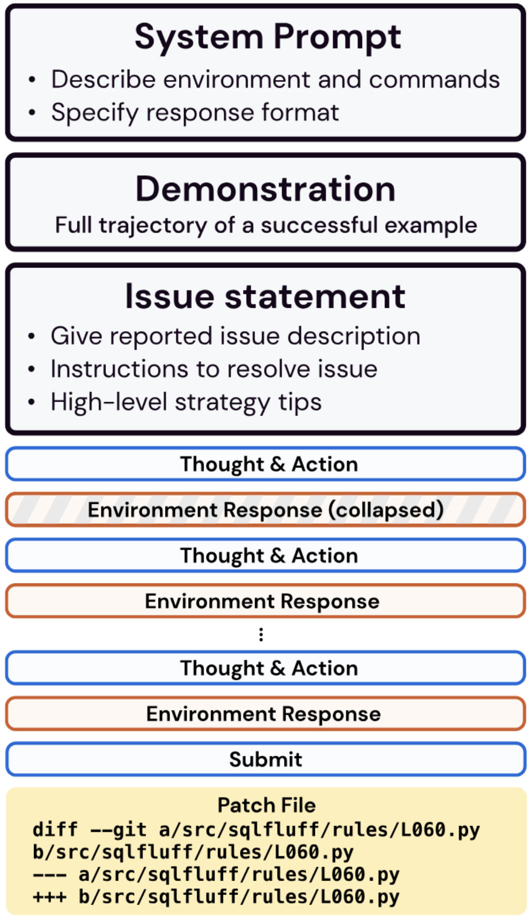
SWE-Agent* uses ReACT



*SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024

SWE-Agent* uses ReACT

Category	Command	Documentation
File viewer	**open** <path> [<line_number>]	Opens the file at the given path in the editor. If line_number is provided, the window will move to include that line.
	goto <line_number>	Moves the window to show line_number.
	scroll_down	Moves the window up 100 lines.
	scroll_up	Moves the window down 100 lines.
Search tools	**search_file** <search_term> [<file>]	Searches for search_term in file. If file is not provided, searches in the current open file.
	search_dir <search_term> [<dir>]	Searches for search_term in all files in dir. If dir is not provided, searches in the current directory.
	find_file <file_name> [<dir>]	Finds all files with the given name in dir. If dir is not provided, searches in the current directory.
File editing	**edit** <n>:<m> <replacement_text> **end_of_edit**	Replaces lines n through m (inclusive) with the given text in the open file. All of the replacement_text will be entered, so make sure your indentation is formatted properly. Python files will be checked for syntax errors after the edit. If an error is found, the edit will not be executed. Reading the error message and modifying your command is recommended as issuing the same command will return the same error.
	create <filename>	Creates and opens a new file with the given name.
Task	**submit**	Generates and submits the patch from all previous edits and closes the shell.



*SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024

SWE-Agent* uses ReACT

File Viewer

```
open atmosphere.py
```

```
(255 more lines above)
256:def gueymard94_pw(temp_air, relative_humidity):
257:  r"""
258:  Calculates precipitable water (cm) from ambient air temperature (C)
259:  and relatively humidity (%) using an empirical model. The
260:  accuracy of this method is approximately 20% for moderate PW (1-3
261:  cm) and less accurate otherwise.
262:
263:  The model was developed by expanding Eq. 1 in [2]_:
264:
265:  .. math::
266:
267:      Pw = 0.1 H_v \rho_v

                    (76 lines omitted)

343:
344:
345:def bird_hulstrom80_aod_bb(aod380, aod500):
346:  """
347:  Approximate broadband aerosol optical depth.
348:
349:  Bird and Hulstrom developed a correlation for broadband aerosol
350:  depth (AOD) using two wavelengths, 380 nm and 500 nm.
351:
352:  Parameters
353:  ----------
354:  aod380 : numeric
355:    AOD measured at 380 nm. [unitless]
(180 more lines below)
```

Find Files

```
find_file atmosphere.py
```

```
Found 2 matches for "atmosphere.py" in /pvlib__pvlib-python/pvlib:
/pvlib__pvlib-python/pvlib/tests/atmosphere.py
/pvlib__pvlib-python/pvlib/atmosphere.py
```

Search Directory

```
search_dir APPARENT_ZENITH_MODELS
```

```
Found 4 matches for "APPARENT_ZENITH_MODELS" in /pvlib__pvlib-
python:
/pvlib__pvlib-python/docs/sphinx/source/whatsnew/v0.3.0.txt (1
matches)
/pvlib__pvlib-python/pvlib/atmosphere.py (2 matches)
/pvlib__pvlib-python/pvlib/location.py (1 matches)
End of matches for "APPARENT_ZENITH_MODELS" in /pvlib__pvlib-
python
```

Search File

```
search_file APPARENT_ZENITH_MODELS
```

```
Found 2 matches for "APPARENT_ZENITH_MODELS" in /pvlib__pvlib-
python/pvlib/atmosphere.py:
Line 12:APPARENT_ZENITH_MODELS = ('simple', 'kasten1966',
'kastenyoung1989',
Line 15:AIRMASS_MODELS = APPARENT_ZENITH_MODELS +
TRUE_ZENITH_MODELS
End of matches for "APPARENT_ZENITH_MODELS" in /pvlib__pvlib-
python/pvlib/atmosphere.py
```

System Prompt
- Describe environment and commands
- Specify response format

Demonstration
Full trajectory of a successful example

Issue statement
- Give reported issue description
- Instructions to resolve issue
- High-level strategy tips

Thought & Action
Environment Response (collapsed)
Thought & Action
Environment Response
⋮
Thought & Action
Environment Response
Submit

Patch File

```
diff --git a/src/sqlfluff/rules/L060.py
b/src/sqlfluff/rules/L060.py
--- a/src/sqlfluff/rules/L060.py
+++ b/src/sqlfluff/rules/L060.py
```

*SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024

SWE-Agent* uses ReACT

Response (Thought + Edit Action)

We need to add a check to ensure that all powers are the same. Let's edit the code to include this check.

```
edit 404:407
    elif (total_degree > 3 and
          len(set(k.exp[k.is_Pow])) == 1):
        if all(coeff[k] == 1):
            diop_type = 'general_sum_even'
end_of_edit
```

System Prompt
- Describe environment and commands
- Specify response format

Demonstration
Full trajectory of a successful example

Issue statement
- Give reported issue description
- Instructions to resolve issue
- High-level strategy tips

Thought & Action
Environment Response (collapsed)
Thought & Action
Environment Response
⋮
Thought & Action
Environment Response
Submit

Patch File
```
diff --git a/src/sqlfluff/rules/L060.py
b/src/sqlfluff/rules/L060.py
--- a/src/sqlfluff/rules/L060.py
+++ b/src/sqlfluff/rules/L060.py
```

*SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024

SWE-Agent* uses ReACT

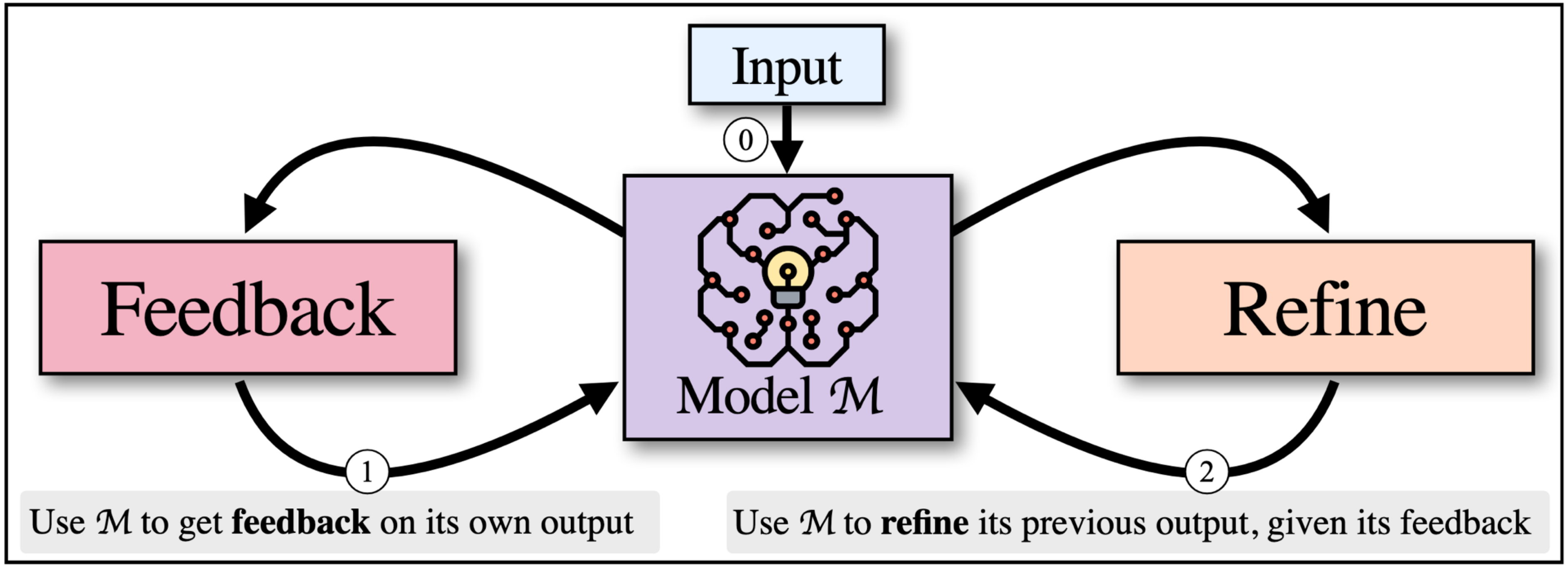
Model	SWE-bench		SWE-bench Lite	
	% Resolved	$ Avg. Cost	% Resolved	$ Avg. Cost
RAG				
w/ GPT-4 Turbo	1.31	0.13	2.67	0.13
w/ Claude 3 Opus	3.79	0.25	4.33	0.25
Shell-only agent				
w/ GPT-4 Turbo	-	-	11.00	1.46
w/o Demonstration	-	-	7.33	0.79
SWE-agent				
w/ GPT-4 Turbo	**12.47**	1.59	**18.00**	1.67
w/ Claude 3 Opus	10.46	2.59	13.00	2.18

*SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024

Outline

- ReACT
  - reasoning + act to arrive at the final answer
- Self-Refine
- Reflexion

Self-Refine



* SELF-REFINE:Iterative Refinement with Self-Feedback, Madaan et. al., May 2023

Self-Refine

(a) **Dialogue:** x, $\mathbf{y_t}$

```
User: I am interested
in playing Table
tennis.

Response: I'm sure
it's a great way to
socialize, stay active
```

(b) **FEEDBACK** $\mathbf{fb}$

```
Engaging: Provides no
information about table
tennis or how to play it.

User understanding: Lacks
understanding of user's
needs and state of mind.
```

(c) **REFINE** $\mathbf{y_{t+1}}$

```
Response (refined): That's
great to hear (...) ! It's
a fun sport requiring
quick reflexes and good
hand-eye coordination.
Have you played before, or
are you looking to learn?
```

(d) **Code optimization:** x, $\mathbf{y_t}$

```
Generate sum of 1, ..., N
def sum(n):
    res = 0
    for i in range(n+1):
        res += i
    return res
```

(e) **FEEDBACK** $\mathbf{fb}$

```
This code is slow as
it uses brute force.
A better approach is
to use the formula
... (n(n+1))/2.
```

(f) **REFINE** $\mathbf{y_{t+1}}$

```
Code (refined)

def sum_faster(n):
  return (n*(n+1))//2
```

* SELF-REFINE:Iterative Refinement with Self-Feedback, Madaan et. al., May 2023

Self-Refine

Algorithm 1 SELF-REFINE algorithm

Require: input x, model $\mathcal{M}$, prompts $\{p_{\text{gen}}, p_{\text{fb}}, p_{\text{refine}}\}$, stop condition $\text{stop}(\cdot)$

$y_0 = \mathcal{M}(p_{\text{gen}} \| x)$ ▷ Initial generation (Eqn. 1)

for iteration t $\in 0, 1, \ldots$ **do**

  $fb_t = \mathcal{M}\left(p_{\text{fb}} \| x \| y_t\right)$ ▷ Feedback (Eqn. 2)

  if $\text{stop}(fb_t, t)$ **then** ▷ Stop condition

    break

  else

    $y_{t+1} = \mathcal{M}\left(p_{\text{refine}} \| x \| y_0 \| fb_0 \| \ldots \| y_t \| fb_t\right)$ ▷ Refine (Eqn. 4)

  end if

end for

return y_t

* SELF-REFINE:Iterative Refinement with Self-Feedback, Madaan et. al., May 2023

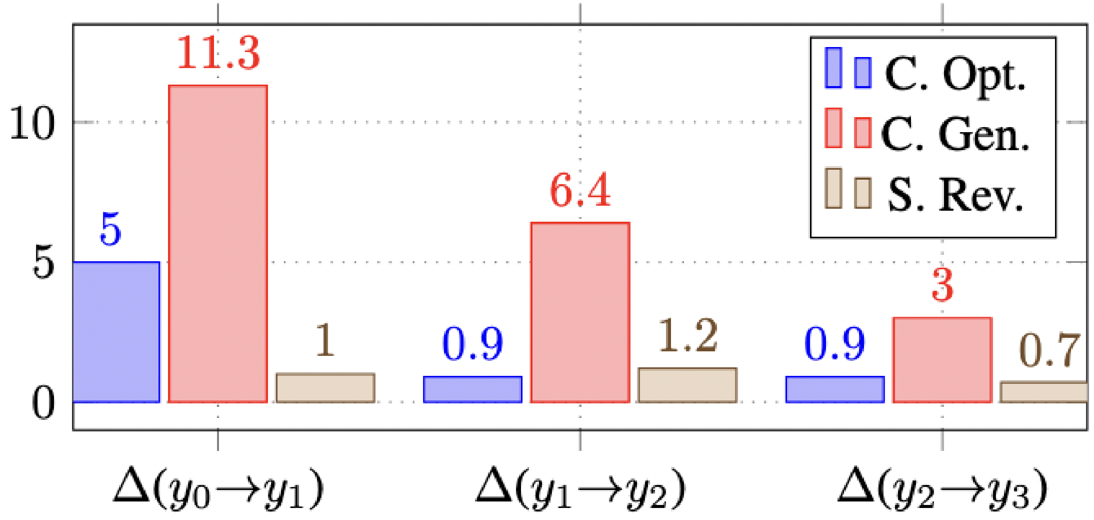
Self-Refine

	GPT-3.5		ChatGPT		GPT-4	
Task	Base	+SELF-REFINE	Base	+SELF-REFINE	Base	+SELF-REFINE
Sentiment Reversal	8.8	**30.4** (↑21.6)	11.4	**43.2** (↑31.8)	3.8	**36.2** (↑32.4)
Dialogue Response	36.4	**63.6** (↑27.2)	40.1	**59.9** (↑19.8)	25.4	**74.6** (↑49.2)
Code Optimization	14.8	**23.0** (↑8.2)	23.9	**27.5** (↑3.6)	27.3	**36.0** (↑8.7)
Code Readability	37.4	**51.3** (↑13.9)	27.7	**63.1** (↑35.4)	27.4	**56.2** (↑28.8)
Math Reasoning	**64.1**	**64.1** (0)	74.8	**75.0** (↑0.2)	92.9	**93.1** (↑0.2)
Acronym Generation	41.6	**56.4** (↑14.8)	27.2	**37.2** (↑10.0)	30.4	**56.0** (↑25.6)
Constrained Generation	28.0	**37.0** (↑9.0)	44.0	**67.0** (↑23.0)	15.0	**45.0** (↑30.0)

* SELF-REFINE:Iterative Refinement with Self-Feedback, Madaan et. al., May 2023

Self-Refine

Task	y_0	y_1	y_2	y_3
Code Opt.	22.0	27.0	27.9	**28.8**
Sentiment Rev.	33.9	34.9	36.1	**36.8**
Constrained Gen.	29.0	40.3	46.7	**49.7**

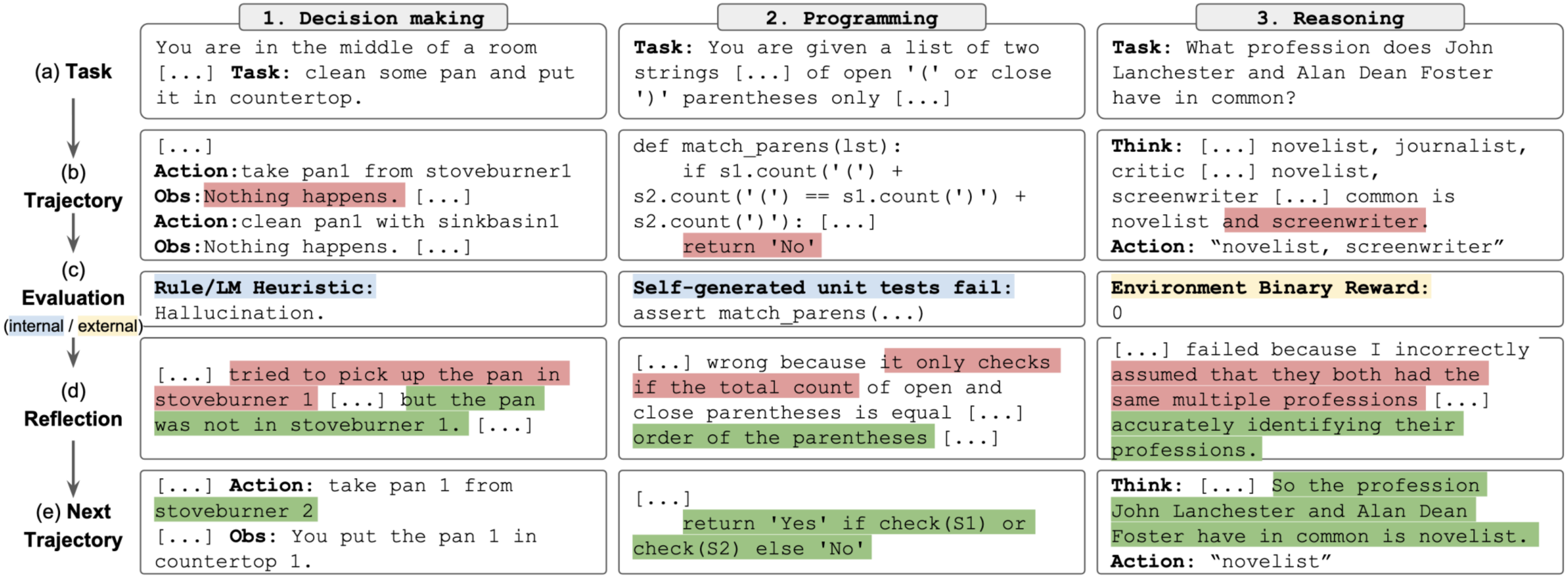


* SELF-REFINE:Iterative Refinement with Self-Feedback, Madaan et. al., May 2023

Outline

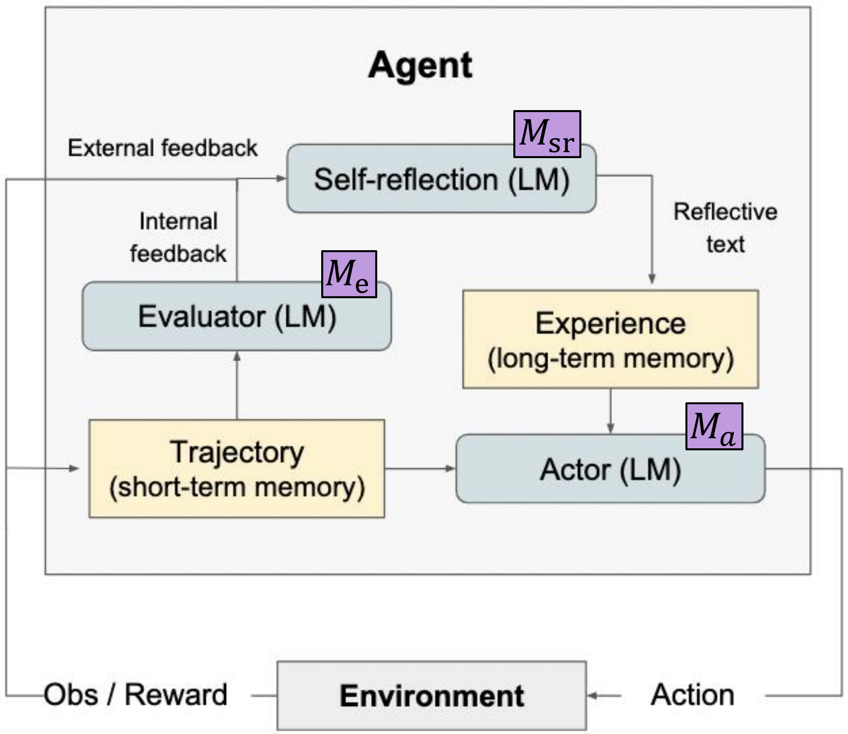
- ReACT
  - reasoning + act to arrive at the final answer
- Self-Refine
  - iteratively improving initial results based on model feedback
  - can be combined with ReACT
- Reflexion

Reflexion



* Reflexion: Language Agents with Verbal Reinforcement Learning, Noah et. al., Oct 2023

Reflexion

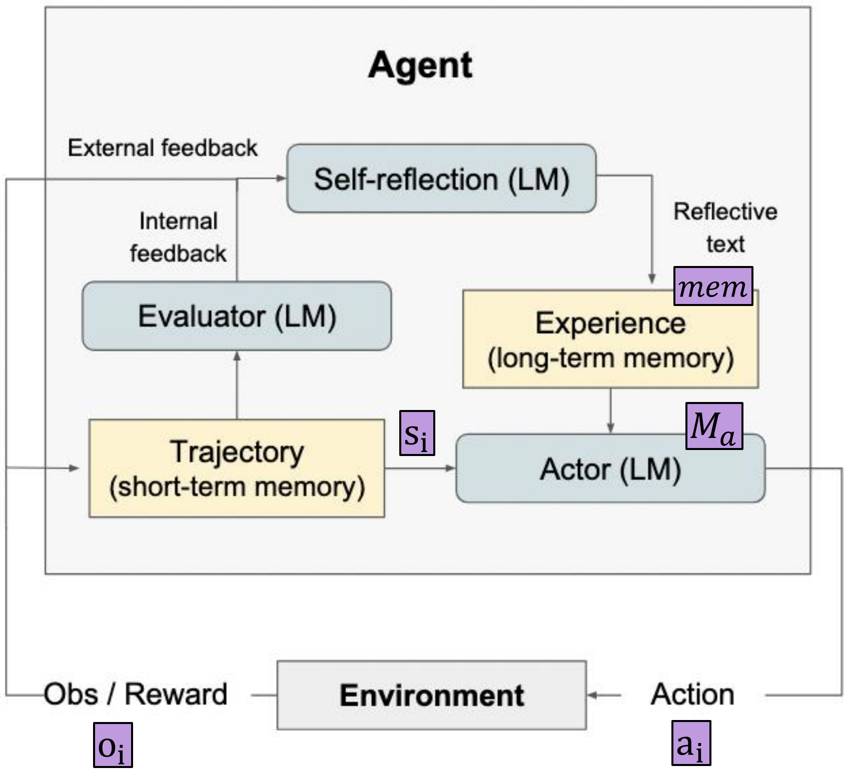


Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection: M_a, M_e, M_{sr}

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion



Algorithm 1 Reinforcement via self-reflection

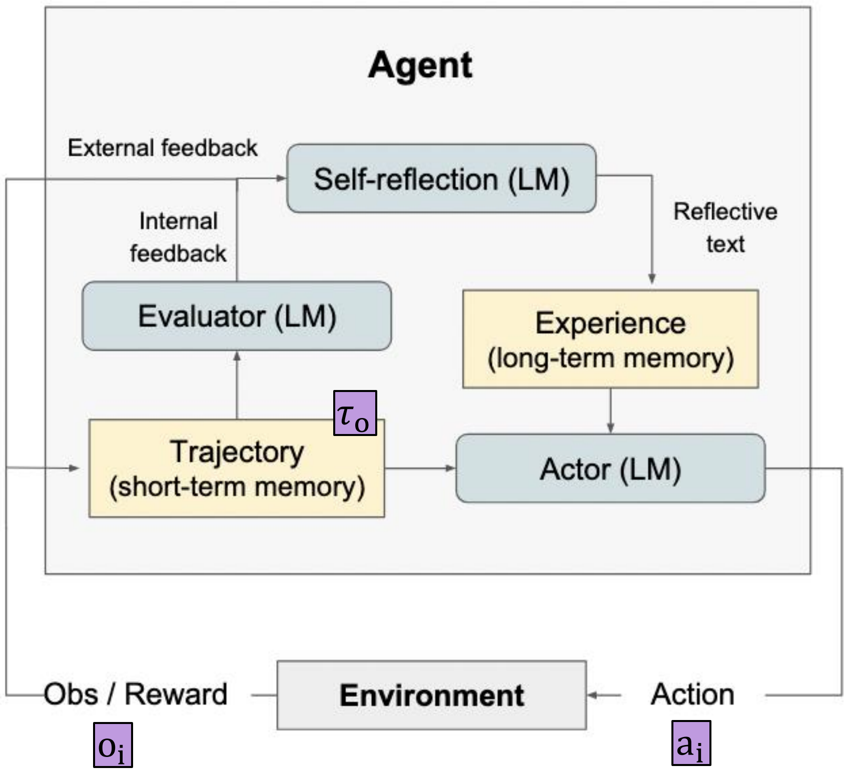
Initialize Actor, Evaluator, Self-Reflection: M_a, M_e, M_{sr}

$\boxed{\text{Initialize policy } \pi_\theta(a_i|s_i), \theta = \{M_a, mem\}}$

$$s_i = [a_o, o_o, \dots, a_{i-1}, o_{i-1}]$$

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion



Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection: M_a, M_e, M_{sr}

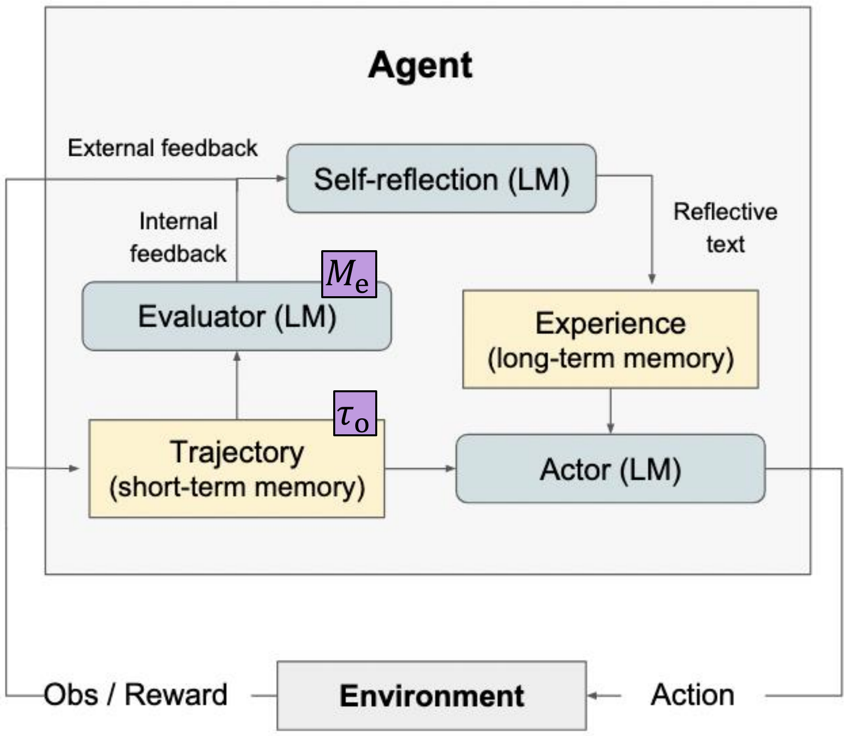
Initialize policy $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$

Generate initial trajectory using π_θ

$$\tau_o = [a_o, o_o, \ldots, a_i, o_i]$$

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion

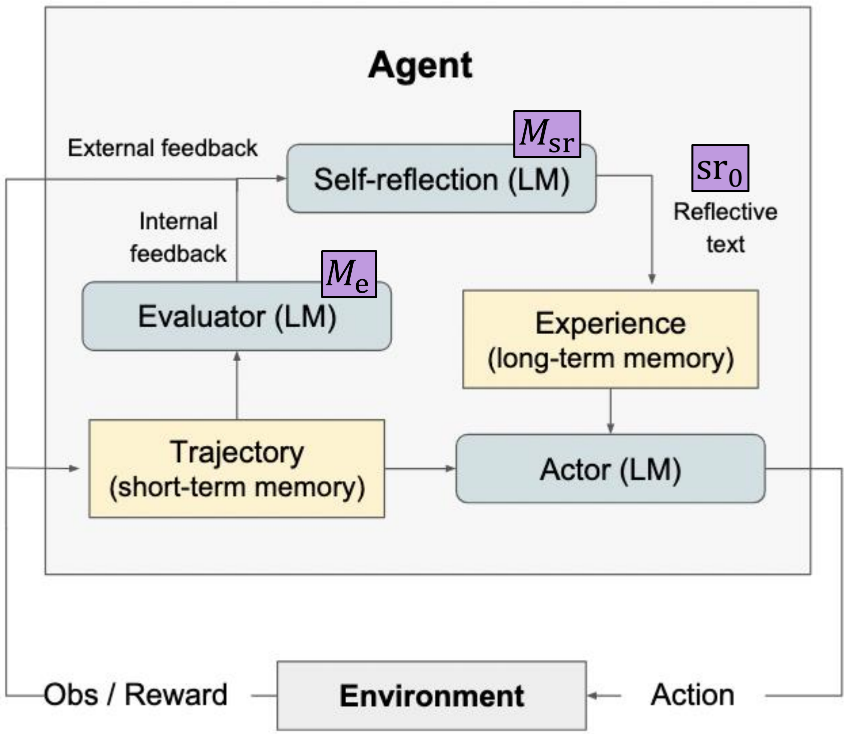


Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection:
M_a, M_e, M_{sr}
Initialize policy $\pi_\theta(a_i|s_i)$, $\theta = \{M_a, mem\}$
Generate initial trajectory using π_θ
Evaluate τ_0 using M_e

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion

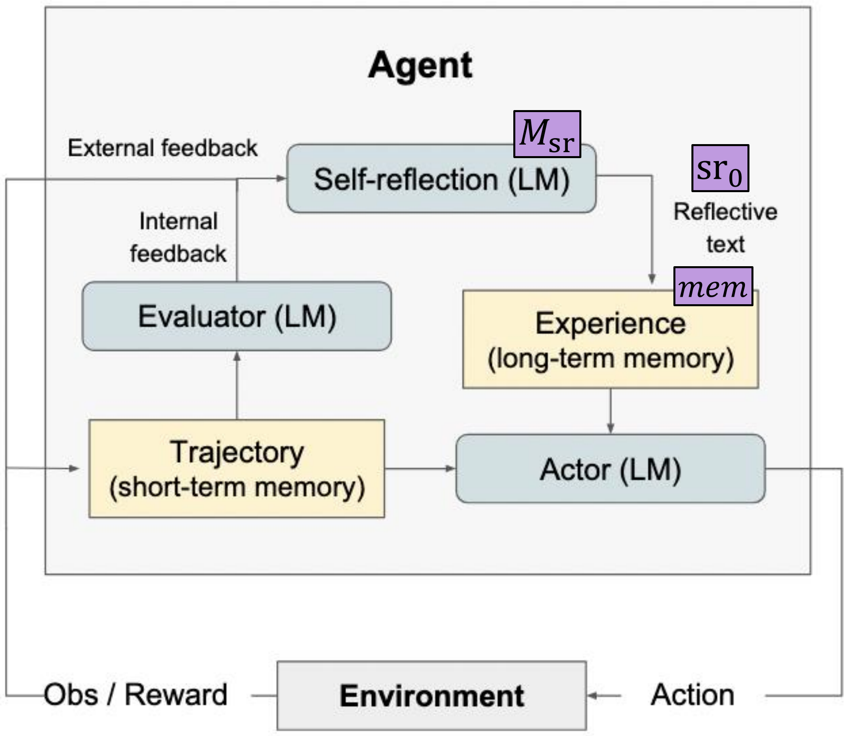


Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection: M_a, M_e, M_{sr}
Initialize policy $\pi_\theta(a_i|s_i)$, $\theta = \{M_a, mem\}$
Generate initial trajectory using π_θ
Evaluate τ_0 using M_e
Generate initial self-reflection sr_0 using M_{sr}

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion

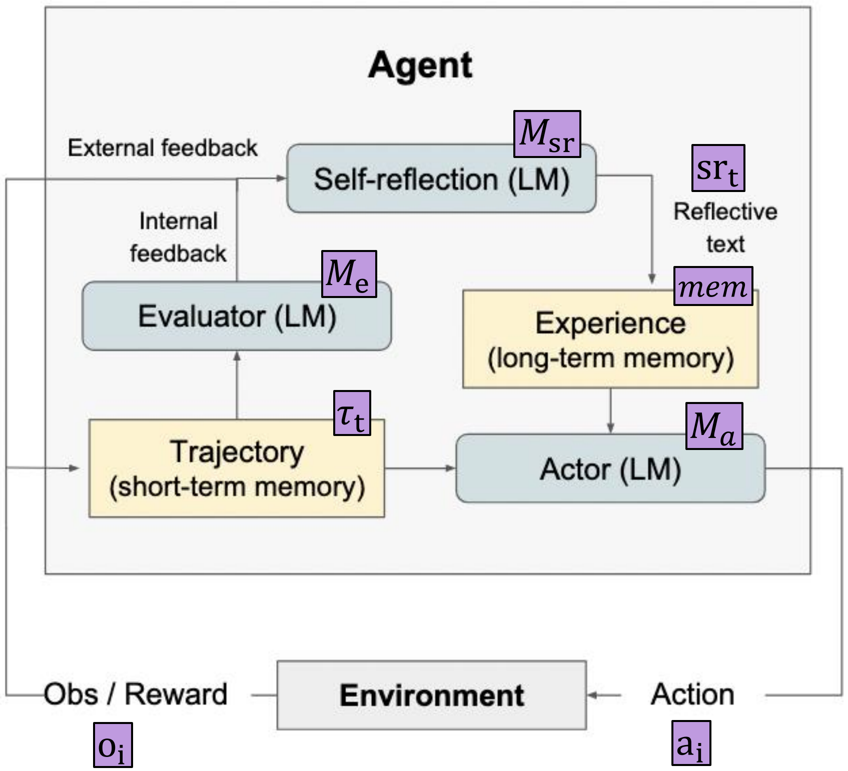


Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection: M_a, M_e, M_{sr}
Initialize policy $\pi_\theta(a_i|s_i)$, $\theta = \{M_a, mem\}$
Generate initial trajectory using π_θ
Evaluate τ_0 using M_e
Generate initial self-reflection sr_0 using M_{sr}
Set $mem \leftarrow [sr_0]$

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion



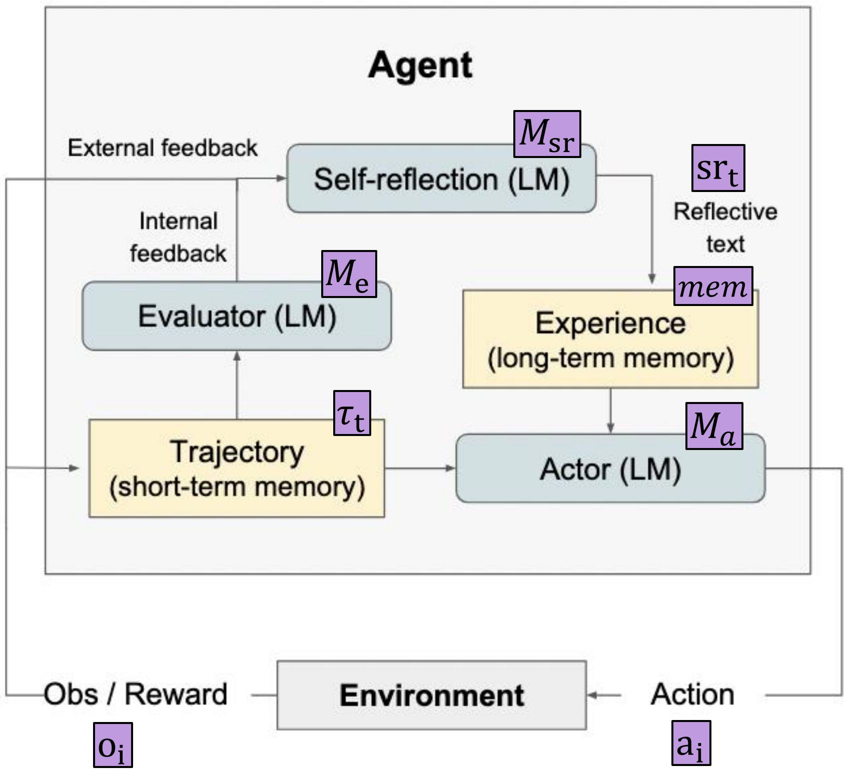
Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection:
M_a, M_e, M_{sr}
Initialize policy $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$
Generate initial trajectory using π_θ
Evaluate τ_0 using M_e
Generate initial self-reflection sr_0 using M_{sr}
Set $mem \leftarrow [sr_0]$
Set $t = 0$
while M_e not pass or $t <$ max trials **do**

end while
return

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion



Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection:
M_a, M_e, M_{sr}
Initialize policy $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$
Generate initial trajectory using π_θ
Evaluate τ_0 using M_e
Generate initial self-reflection sr_0 using M_{sr}
Set $mem \leftarrow [sr_0]$
Set $t = 0$
while M_e not pass or $t <$ max trials **do**
    Generate $\tau_t = [a_0, o_0, \ldots a_i, o_i]$ using π_θ
    Evaluate τ_t using M_e
    Generate self-reflection sr_t using M_{sr}
    Append sr_t to mem
    Increment t
end while
return

Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion

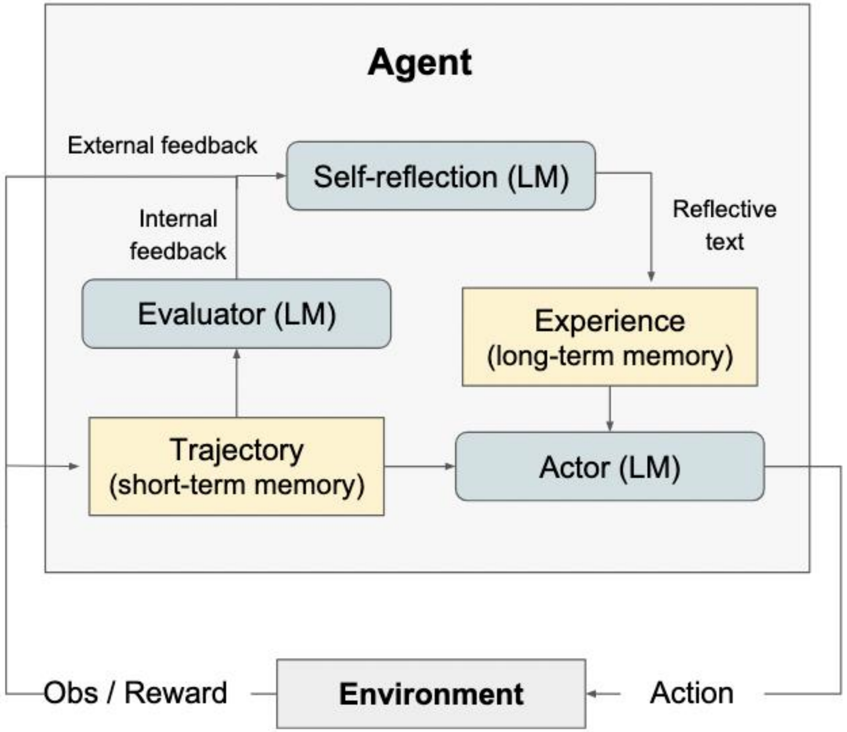
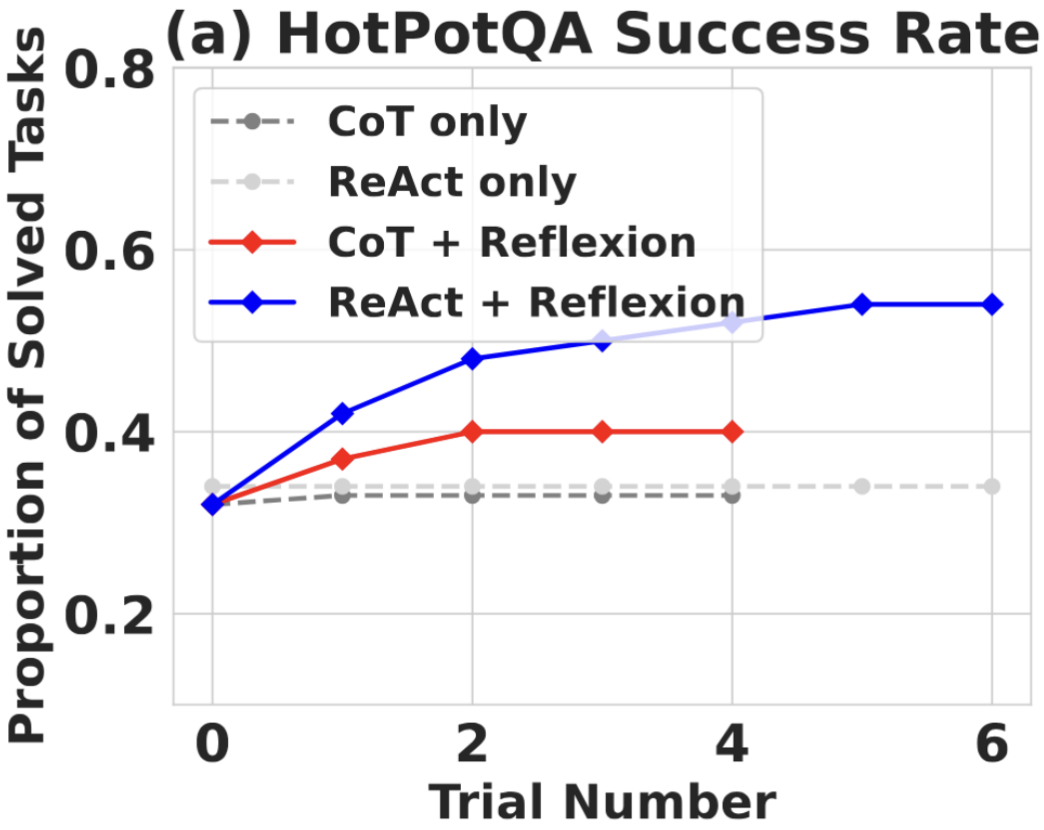


Image source: https://langchain-ai.github.io/langgraph/tutorials/reflexion/reflexion/

Reflexion



* Reflexion: Language Agents with Verbal Reinforcement Learning, Noah et. al., Oct 2023

Reflexion

Benchmark + Language	Base	Reflexion
HumanEval (PY)	0.80	**0.91**
MBPP (PY)	**0.80**	0.77
HumanEval (RS)	0.60	**0.68**
MBPP (RS)	0.71	**0.75**

Overall accuracy for HumanEval and MBPP

* Reflexion: Language Agents with Verbal Reinforcement Learning, Noah et. al., Oct 2023

Outline

- ReACT
  - reasoning + act to arrive at the final answer
- Self-Refine
  - iteratively improving initial results based on model feedback
  - can be combined with ReACT
- Reflexion
  - iteratively improving initial results based on model feedback
  - uses of tools and LLMs for reflection
  - can be combined with ReACT