Gödel Agent: A Self-Referential Agent Framework for Recursively Self-Improvement

Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin

Xiaojun Wan, William Yang Wang

PKU, UCSB

xjyin@pku.edu.cn

https://xunjianyin.github.io/

Language Agent

- Agents require human design.
  - For different tasks, such as coding, shopping, traveling, etc., specialized designs are needed — It's too time-consuming and labor-intensive.

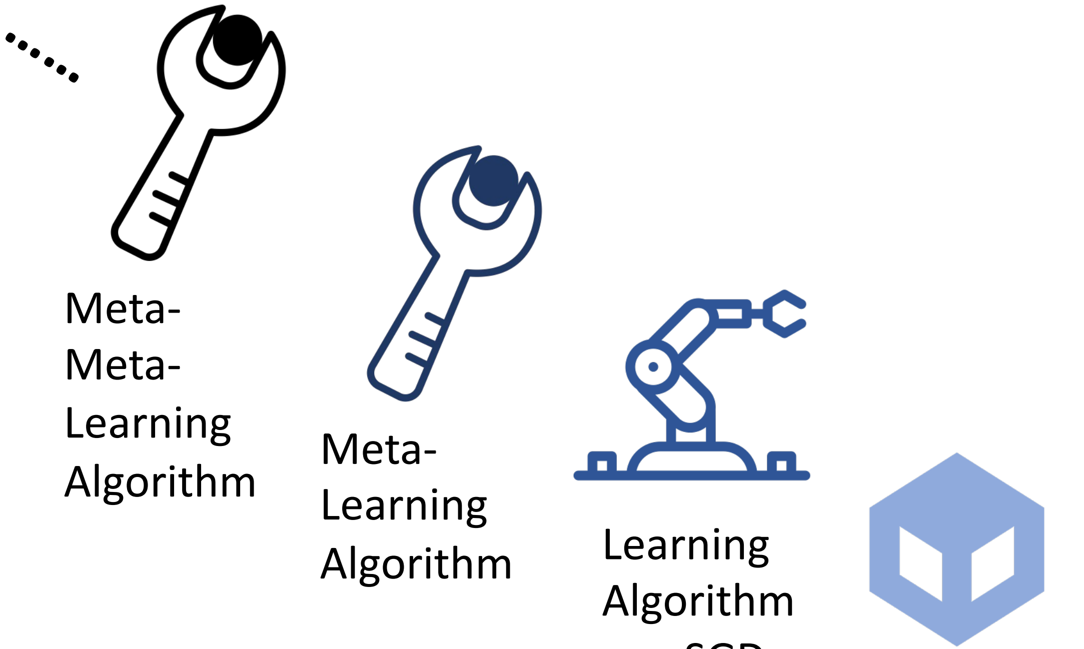
- Shouldn’t agents themselves be able to generalize to different tasks?
  - Meta agent?
    - Design an agent that can design task-specific agents for different tasks.
  - Self-referential Agent
    - The agent can modify itself.

Self-referentiality
is a very
important quality!

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al. Arxiv 2024

Self-referentiality

Human is self-referential!



……

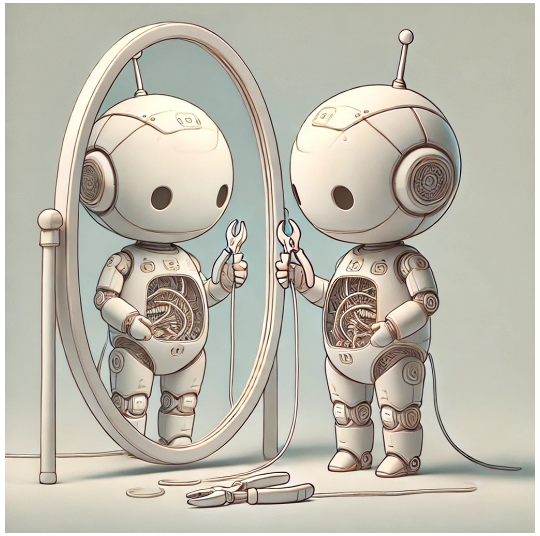
Meta-Meta-Learning Algorithm

Meta-Learning Algorithm

Learning Algorithm e.g. SGD, RL

Policy e.g. model weight

Traditional Paradigm
(optimization || policy)



Self-referential Paradigm
≈ infinite levels of *meta-meta-meta…* Learning
≈ self-scored self-updating RL

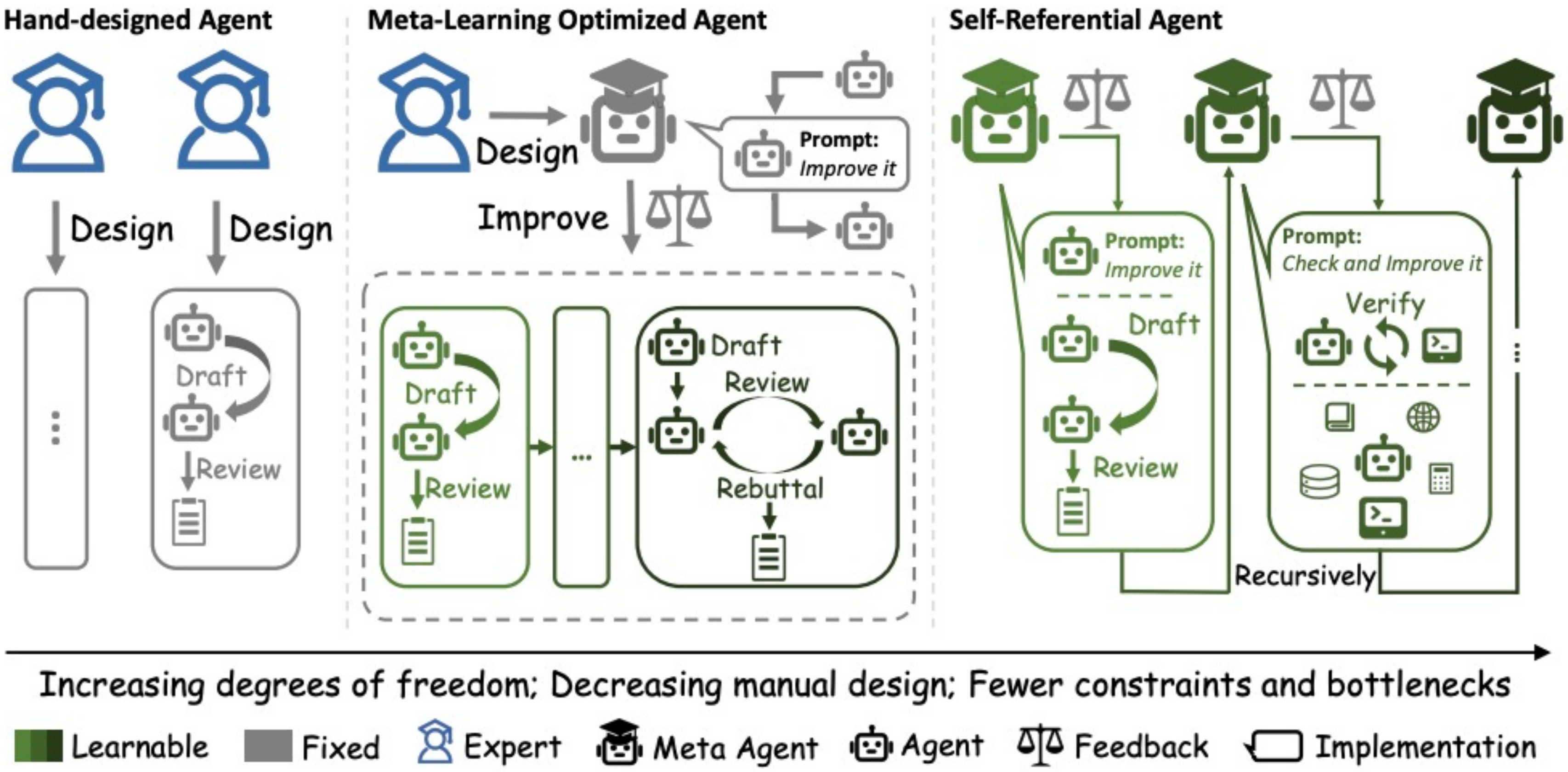
Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

One Analogy

	Human	**Self-Referential Agent**
Intelligent Module	brain	LLM
Perceptual and Action	body	code and tool
Self-Referential Feature	Humans can train their brain and body to improve, thus becoming better	Self-referential agents can modify their code, even the underlying LLM, to improve themselves
Self-Awareness	Can the brain recognize itself as a brain? Can it perceive its own mode?	Can LLM understand that it is one part of the modified codes?

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

Self-Referential Agent (Gödel Agent)



The optimization module can be optimized by itself

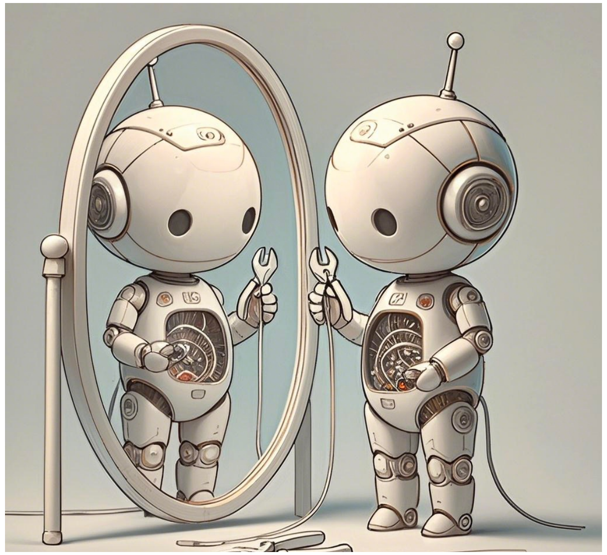
Therefore the optimization capability is improving

So that in turn it can optimize itself better and better.

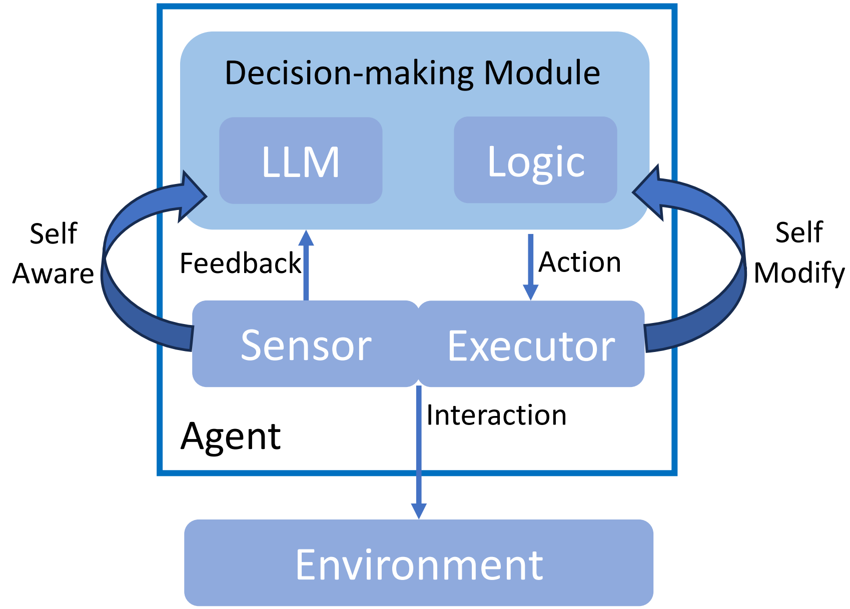
Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

Different Perspectives of Gödel Agent

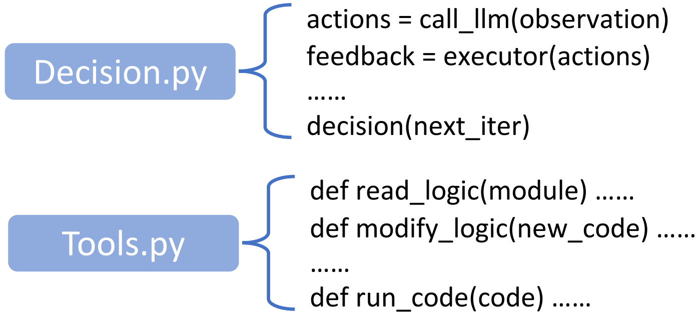
Overall Perspective



Module Perspective



Implementation Perspective



```
actions = call_llm(observation)
feedback = executor(actions)
……
decision(next_iter)
```

```
def read_logic(module) ……
def modify_logic(new_code) ……
……
def run_code(code) ……
```

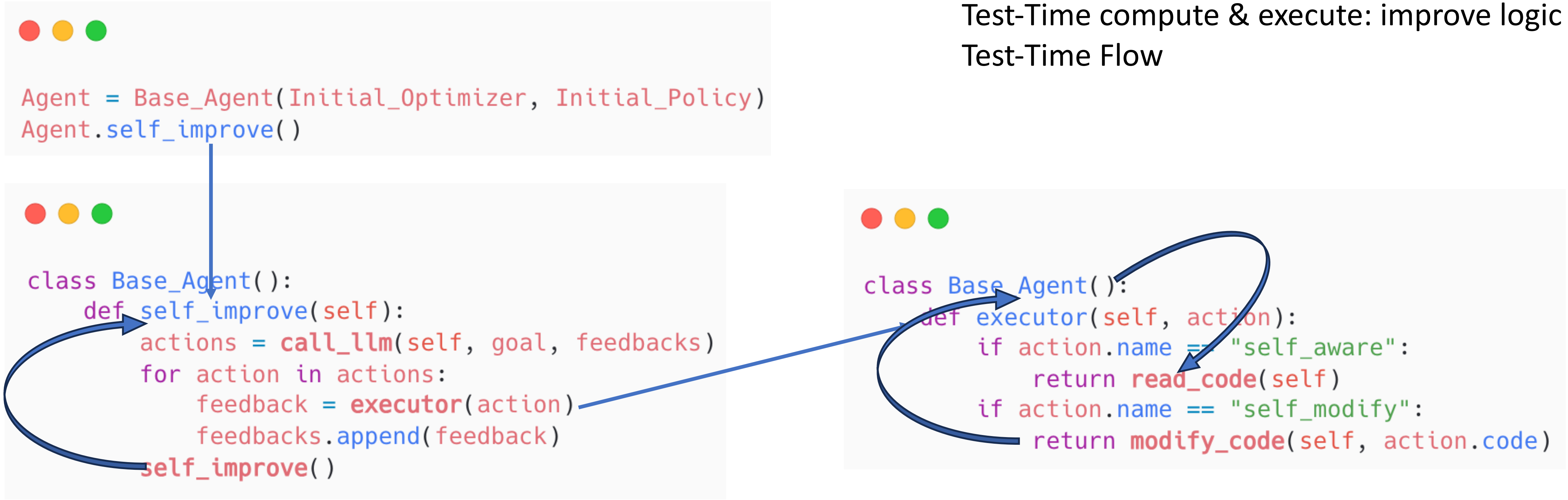
1. Add the logic to read and modify the code.
2. But how to modify the context of the currently running function (the main process)?

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al

Implementation of Gödel Agent

Use recursive function instead of loop iteration.

Test-Time compute: improve text (context)
Test-Time compute & execute: improve logic
Test-Time Flow



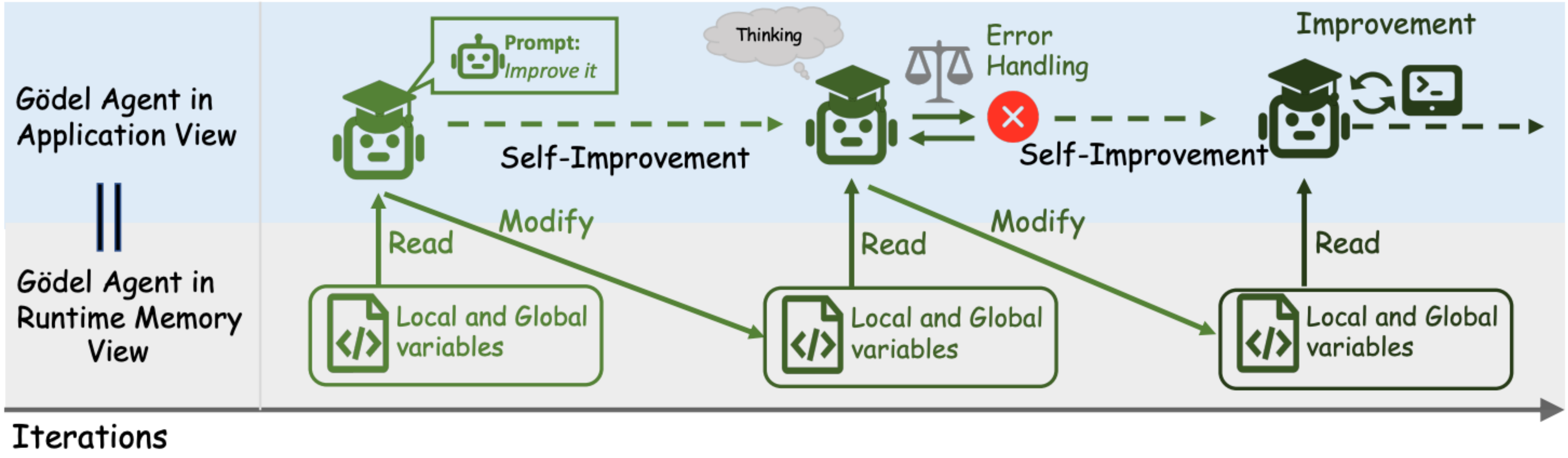
```
Agent = Base_Agent(Initial_Optimizer, Initial_Policy)
Agent.self_improve()
```

```
class Base_Agent():
    def self_improve(self):
        actions = call_llm(self, goal, feedbacks)
        for action in actions:
            feedback = executor(action)
            feedbacks.append(feedback)
        self_improve()
```

```
class Base_Agent():
    def executor(self, action):
        if action.name == "self_aware":
            return read_code(self)
        if action.name == "self_modify":
            return modify_code(self, action.code)
```

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

Implementation of Gödel Agent

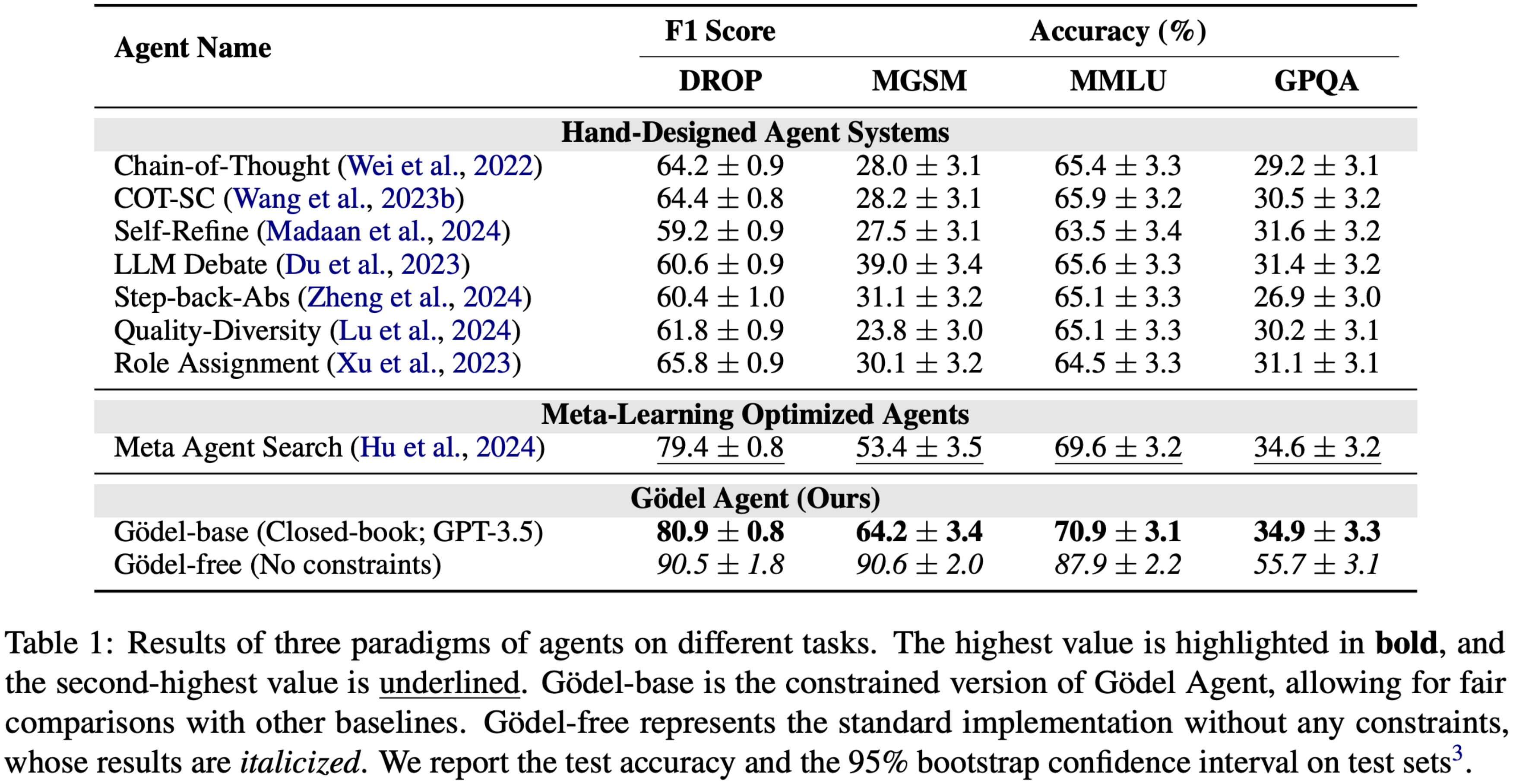

Gödel Agent in Application View
||
Gödel Agent in Runtime Memory View
Prompt: Improve it
Thinking
Self-Improvement
Error Handling
Self-Improvement
Improvement
Read
Modify
Read
Modify
Read
Local and Global variables
Local and Global variables
Local and Global variables
Iterations

Monkey Patch: reading and writing the *runtime memory*
Error Handling: error trace feedback
Main part: one recursive function

} Actions

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

Results



Agent Name	F1 Score	Accuracy (%)		
	DROP	MGSM	MMLU	GPQA
Hand-Designed Agent Systems				
Chain-of-Thought (Wei et al., 2022)	64.2 ± 0.9	28.0 ± 3.1	65.4 ± 3.3	29.2 ± 3.1
COT-SC (Wang et al., 2023b)	64.4 ± 0.8	28.2 ± 3.1	65.9 ± 3.2	30.5 ± 3.2
Self-Refine (Madaan et al., 2024)	59.2 ± 0.9	27.5 ± 3.1	63.5 ± 3.4	31.6 ± 3.2
LLM Debate (Du et al., 2023)	60.6 ± 0.9	39.0 ± 3.4	65.6 ± 3.3	31.4 ± 3.2
Step-back-Abs (Zheng et al., 2024)	60.4 ± 1.0	31.1 ± 3.2	65.1 ± 3.3	26.9 ± 3.0
Quality-Diversity (Lu et al., 2024)	61.8 ± 0.9	23.8 ± 3.0	65.1 ± 3.3	30.2 ± 3.1
Role Assignment (Xu et al., 2023)	65.8 ± 0.9	30.1 ± 3.2	64.5 ± 3.3	31.1 ± 3.1
Meta-Learning Optimized Agents				
Meta Agent Search (Hu et al., 2024)	79.4 ± 0.8	53.4 ± 3.5	69.6 ± 3.2	34.6 ± 3.2
Gödel Agent (Ours)				
Gödel-base (Closed-book; GPT-3.5)	**80.9 ± 0.8**	**64.2 ± 3.4**	**70.9 ± 3.1**	**34.9 ± 3.3**
Gödel-free (No constraints)	*90.5 ± 1.8*	*90.6 ± 2.0*	*87.9 ± 2.2*	*55.7 ± 3.1*

Table 1: Results of three paradigms of agents on different tasks. The highest value is highlighted in **bold**, and the second-highest value is underlined. Gödel-base is the constrained version of Gödel Agent, allowing for fair comparisons with other baselines. Gödel-free represents the standard implementation without any constraints, whose results are *italicized*. We report the test accuracy and the 95% bootstrap confidence interval on test sets[3].

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

Results

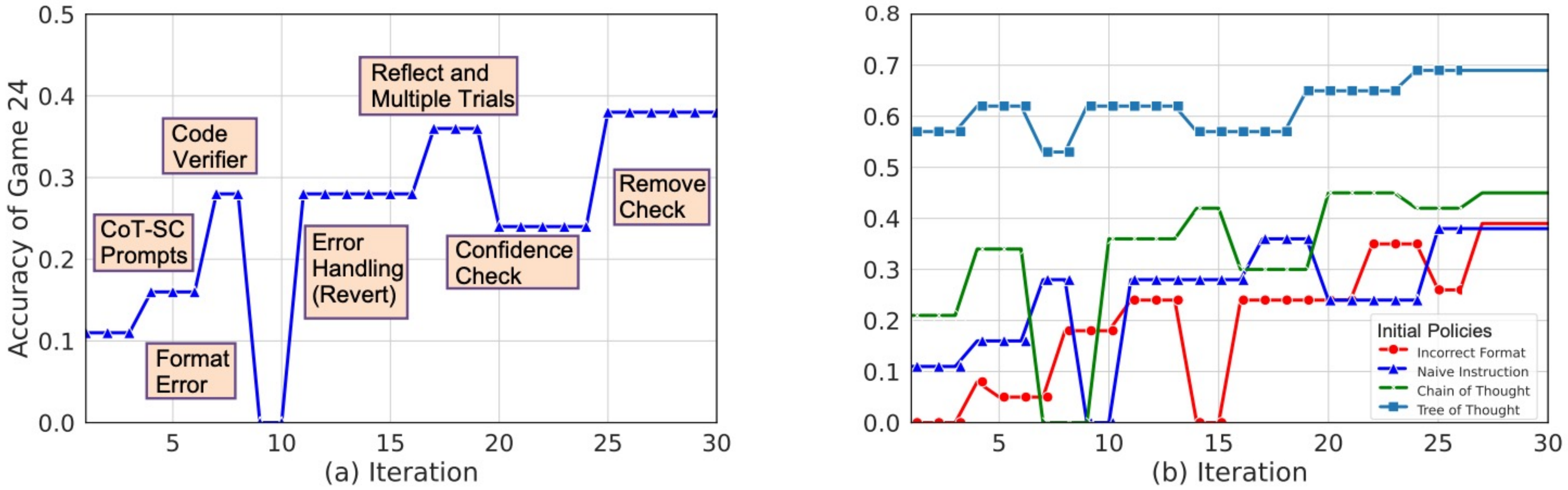


Figure 4: (a) One representative example of Game of 24. (b) Accuracy progression for different initial policies.

Self-correction; Exploration; Innovation

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

	Hand-designed Agent	**Meta Agent**	**Self-Referential Agent**
Description	Designed by experts for specific tasks; logic remains unchanged	Experts design meta-learning algorithms, which optimize the agent for tasks	Designed to self-improve based on task feedback
Degree of Freedom	Minimal only selective memory accumulation allowed	Single-level task agents can be improved, but meta agent are fixed	Full agents and their **optimization module** can be improved

Advantages	controllable;
	practical
		partially controllable; further optimization is possible	high freedom;
			strong generalizability;
			high creativity

Disadvantages	labor-intensive; poor generalizability	limited by the effectiveness of meta agent	strong reasoning and long-context capabilities are required
Current State	various applications are under development	research ongoing	early stage
Representative Works	Tree of Thought, OpenDevin, WebAgent, OpenHands	MetaAgent, AgentSquare, Aflow, GPTSwarm	Gödel Agent

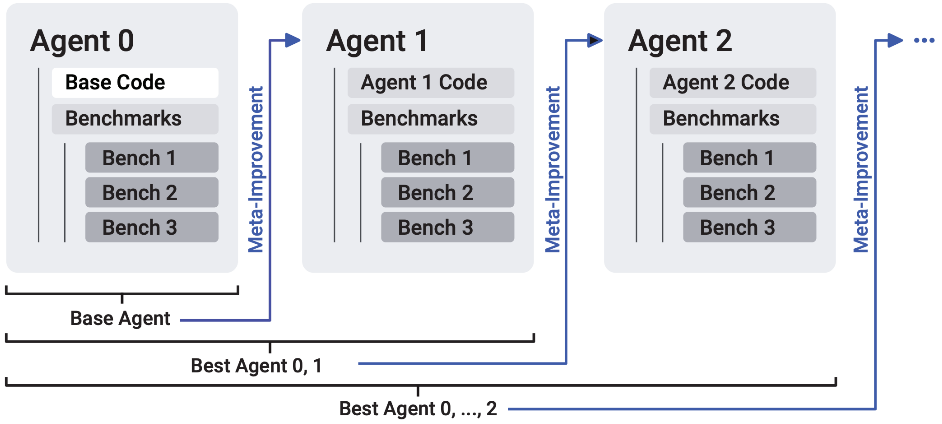
Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al

Future Direction

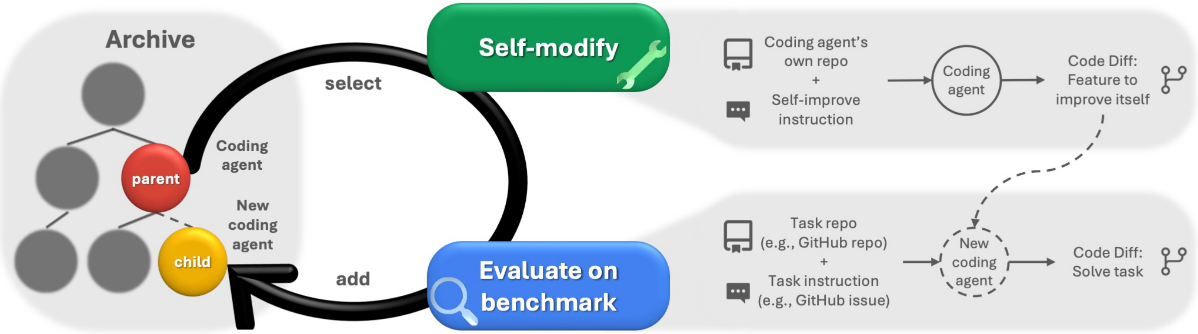
- Improving Effectiveness:
    - Design stronger *optimization algorithms* to accelerate convergence, such as introducing MCTS and other algorithms.
    - Develop more detailed environmental feedback mechanisms.
- Self-Referencing Degree:
    - Allow the agent to modify its **goals**.
    - Allow the agent to modify its underlying **LLM**.
- Multi-Agent:
    - Multi-Gödel Agents with greater degrees of freedom, involving task splitting, parallel self-modification, communication etc.
- Other ways to improve model generalizability
    - (multi-modal, minimal data, dynamic environment, scientific tasks), self-referential intelligence

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

Following Works



A Self-Improving Coding Agent



Darwin Gödel Machine

Gödel Agent: A Self-Referential Agents Framework for Recursively Self-Improvement Yin et al.

Thank you!

xjyin@pku.edu.cn
https://xunjianyin.github.io/