

K y o t o A I

~AI Project from Kyoto~

CEO & Founder Kentaro Seki



Background

Recent AI research shows that LLMs can enhance their thinking through reinforcement learning, but deep thinking with corporate documents and code remains challenging. Our company has developed SEIMEI, a model integrating AI Agents with reinforcement learning of search engine, to provide to businesses.

Research Result

Using bigcodebench, we created over 1,000 AI Agents from training data. By applying reinforcement learning to their search engines, we observed an approximately 16% improvement in accuracy on evaluation data.

Application

This approach can be applied to AI Agents capable of advanced thinking, enabling tasks such as corporate document analysis, code analysis, debugging, auto-generation, and automated presentation materials.

Content

0 1 What's KyotoAI?

0 2 Current Issues on LLM

0 3 KyotoAI Model SEIMEI

0 4 Research Result

0 5 Application

01

W h a t ' s K y o t o A I ?

KyotoAI. Inc

Through research focused on memory in machine learning, we aim to achieve AGI, the ultimate goal of AI research, and contribute to the development of a prosperous society.

Name	KyotoAI.Inc
Foundation	Mar/5th/2024
CEO	Kentaro Seki
Address	Shimogyo Ward, Kyoto City
Email	seki.kentaro@kyotoai.org

02

Current Issues On LLM

The current limitations of LLM inference

Current LLMs are surpassing humans in simple question-and-answer tasks.

However, the use cases for complex tasks involving AI Agents and RAG still face limitations.

I will show you 3 examples

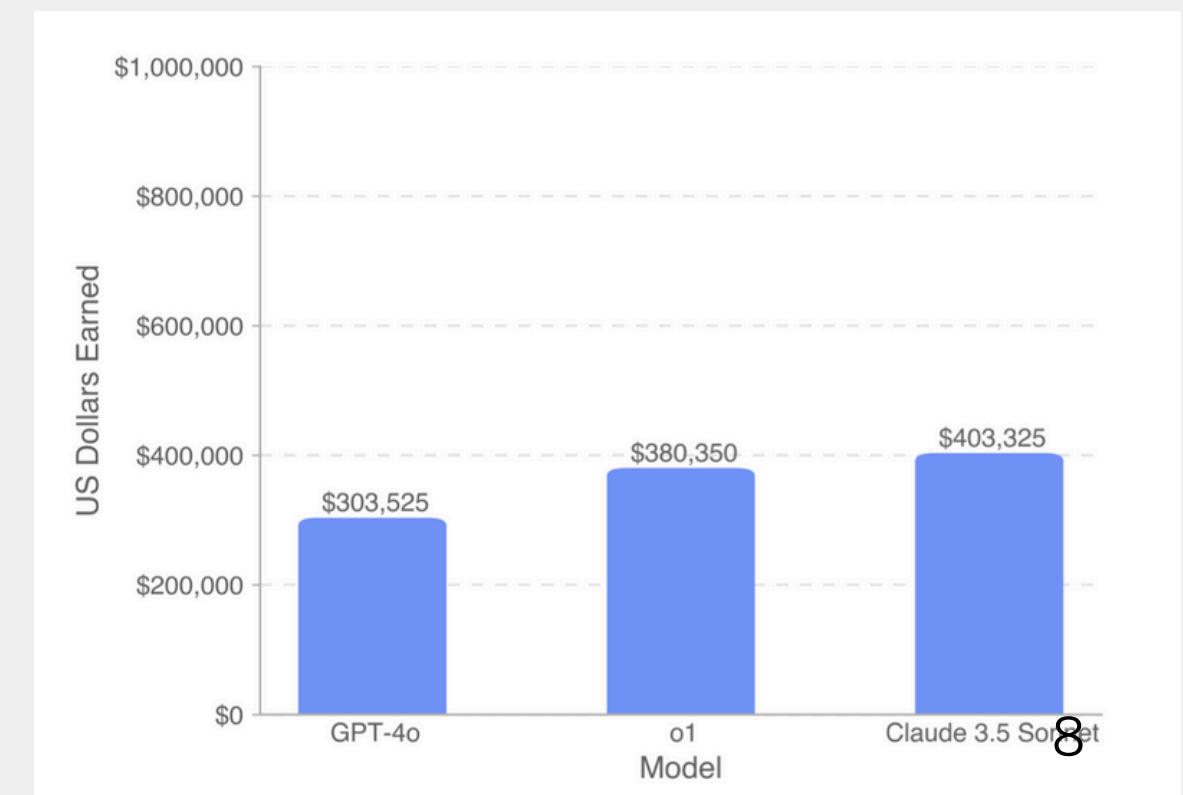


The current limitations of LLM inference ①

SWE-Lancer Benchmark [[OpenAI arXiv:2502.12115](https://arxiv.org/abs/2502.12115)]

this paper showed that the performance of RAG on tasks required of actual software engineers still does not match that of real engineers.

While LLM can surpass humans in simple coding problems during hackathons, using them in initial tasks results in a sudden drop in accuracy.

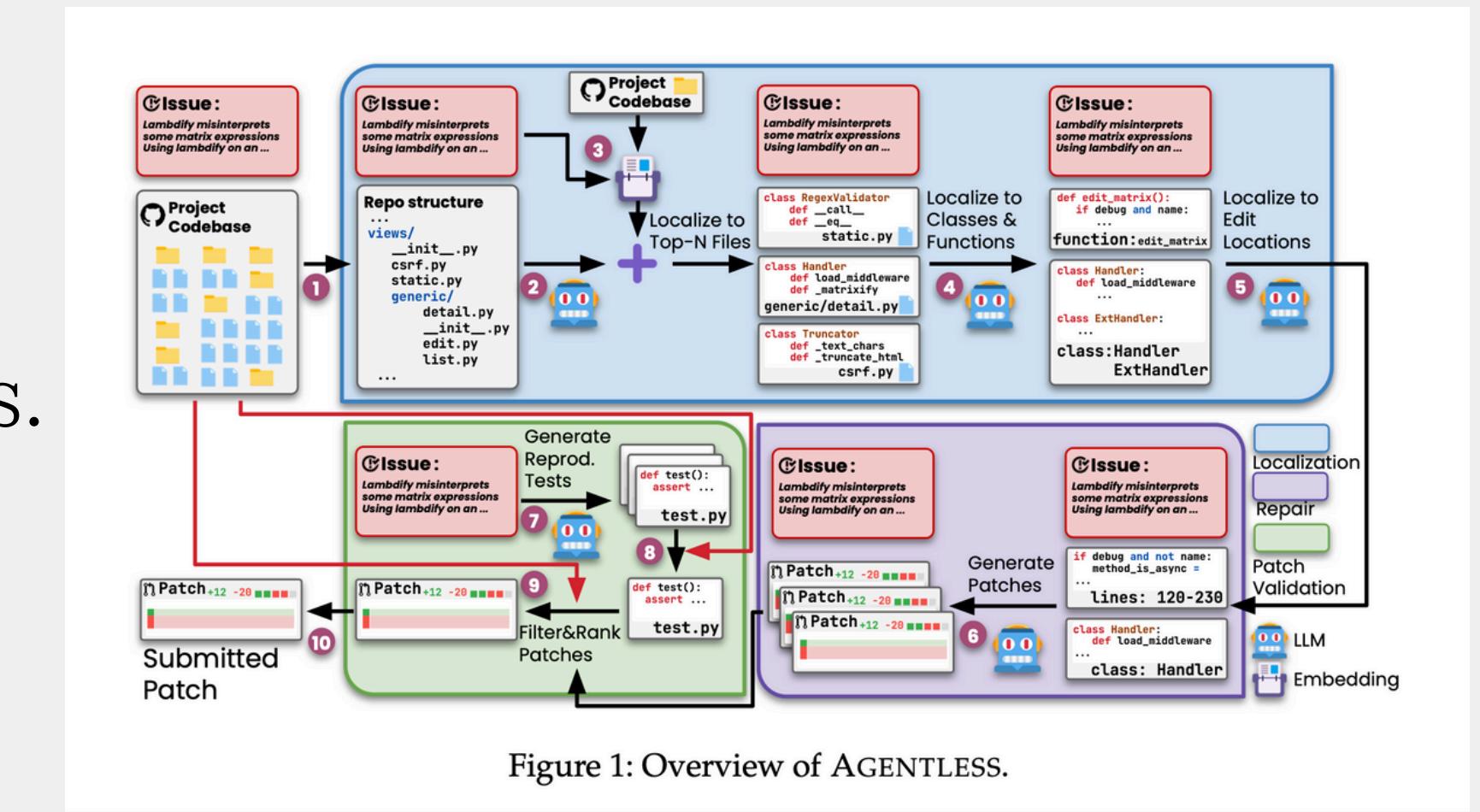


The current limitations of LLM inference ②

Agentless [C. Xia arXiv:2407.01489]

This paper showed that simpler systems with fewer Agents have higher accuracy compared to systems with many Agents.

As the number of AI Agents increases, they become unstable in handling long thought processes.



The current limitations of LLM inference ③

Our Research: AI Agent on Nuclear Fusion Simulation

A screenshot of a GitHub repository page for 'gkvp-developers/gkvp'. The repository has 5 stars and 10 forks. The 'main' branch is selected. The 'src/main' folder contains several sub-directories like benchmarks, extra_tools, lib, and run. The commit history shows numerous commits from 'smaeyama' over the past 4 years, mostly related to NetCDF file additions and modifications. A specific commit from 10 months ago is highlighted.

Name	Last comment message	Last comment date
...		
gkvp_advncf90	Update History is remarked.	10 months ago
gkvp_bndryf90	Source files for NetCDF are added in src/	3 years ago
gkvp_clockf90	Source files for NetCDF are added in src/	3 years ago
gkvp_callf90	equib_type = n-alpha-shift is added.	3 years ago
gkvp_callimpf90	equib_type = n-alpha-shift is added	3 years ago
gkvp_dcf90	equib_typering is added. gkvp_gearn.f90 is teste...	10 months ago
gkvp_ebf90	Source files for NetCDF are added in src/	3 years ago
gkvp_f0_68_advncf90	gkvp_f0_68	4 years ago
gkvp_f0_68_bndryf90	gkvp_f0_68	4 years ago
gkvp_f0_68_callf90	Treat tracer particle (f=0). Field-particle operat...	3 years ago
gkvp_f0_68_ebf90	gkvp_f0_68	4 years ago
gkvp_f0_68_ftt_bndryf90	gkvp_f0_68	4 years ago
gkvp_f0_68_ftt_bndryf90	Source files for NetCDF are added in src/	3 years ago
gkvp_f0_68_ftt_bndryf90	Source files for NetCDF are added in src/	3 years ago
gkvp_f0_68_headerf90	Marked shoot/bndry for JFID 1, while NetCDF do...	3 years ago
gkvp_f0_68_headerf90	Source files for NetCDF are added in src/	3 years ago
gkvp_f0_68_headerf90	Source files for NetCDF are added in src/	3 years ago
gkvp_f0_68_headerf90	Update History is remarked.	10 months ago
gkvp_ftt90	Source files for NetCDF are added in src/	3 years ago
gkvp_mainf90	equib_typering is added. gkvp_geom.f90 is teste...	10 months ago
gkvp_mainpf90	Source files for NetCDF are added in src/	5 years ago
gkvp_outf90	Source files for NetCDF are added in src/	3 years ago
gkvp_mpif90	equib_typering is added. gkvp_geom.f90 is teste...	10 months ago
gkvp_nef90	Update History is remarked.	10 months ago
gkvp_sheaflowf90	Source files for NetCDF are added in src/	3 years ago
gkvp_tpa90	Source files for NetCDF are added in src/	3 years ago
gkvp_transf90	Source files for NetCDF are added in src/	3 years ago
gkvp_vmeconf90	equib_typering is added. gkvp_geom.f90 is teste...	10 months ago
gkvp_vmeconf90	gkvp_f0_68	4 years ago
gkvp_ztmerf90	Source files for NetCDF are added in src/	3 years ago

Analyze

AI Agent
RAG

Dialog

Complex Tasks

Ex. Convert the coordinate
of the simulation

Explain what happens
when running the code

Big Simulation Code

The current limitations of LLM inference ③

Result :

Not accurate on complex problems

Cause :

- AI Agent cannot think coherently
- So many things to think over many files

Name	Last Modified	File Size
Answer.py	last month	1.3 KB
CheckInf.py	last month	4.1 KB
ChunkSurvey.py	last month	1.7 KB
CollectCodefileToModify.py	last month	3.6 KB
CollectInfoToModifyCode.py	last month	3.1 KB
FileSurvey2.py	last month	1.7 KB
MetaModifyCode.py	last month	5.9 KB
MetaSurvey.py	last month	6.4 KB
MetaSurvey2.py	last month	6.3 KB
MetaSurvey2CheckInfo.py	last month	2.1 KB
ModifyCodeChunk.py	last month	4 KB
ModifyCodeChunk2.py	last month	4 KB
ModifyCodeFile.py	last month	4.9 KB
ModifyCodeFile2.py	last month	5 KB
QuickSummary.py	last month	3.3 KB
SelfRoute.py	last month	3.6 KB
SummarizeCode.py	last month	2.9 KB
SummarizeInit.py	last month	2.7 KB

```

61     if file_path == survey_file_path:
62         summarize_code = SummarizeCode(self)
63         experts.append(summarize_code("chunk_id":i))
64         chunk_ids.append(i)
65
66     code_summaries = await asyncio.gather(*experts)
67
68     code_summaries_text = ""
69     for i, code_summary in enumerate(code_summaries):
70         code_summaries_text += f"""`{code_summary(i)}\n"""
71 {code_summary}
72 """
73
74 prompt = f"""## CODE SUMMARIES:
75 {code_summaries_text}
76
77 ## FILE META INFO:
78 |-path: {survey_file_path}
79
80
81 ## QUERY:
82 ```query
83 {query}
84 ```

85
86
87 You are an advanced language model tasked with identifying the part of the code that should be modified based on a given query. You are provided
with some code summaries above, which are summarized from original code snippets. Please follow the instructions and output format below.

88
89
90
91 ## Instructions:
92 1. **Analyze the Query and Code Summaries**: Carefully read and understand the provided query and code summaries.

```

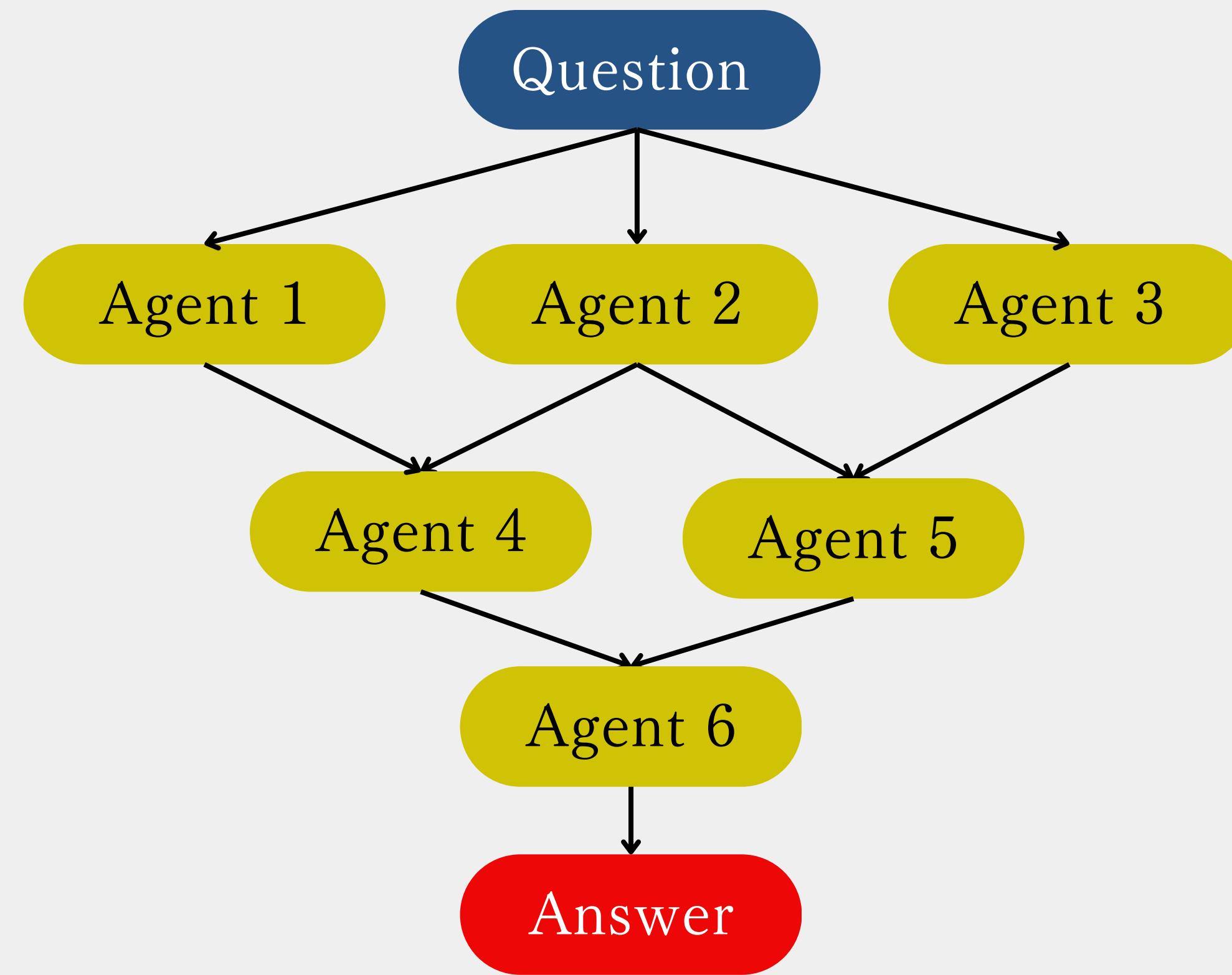
prompts

Why Can't AI Agent Think Deeply ?

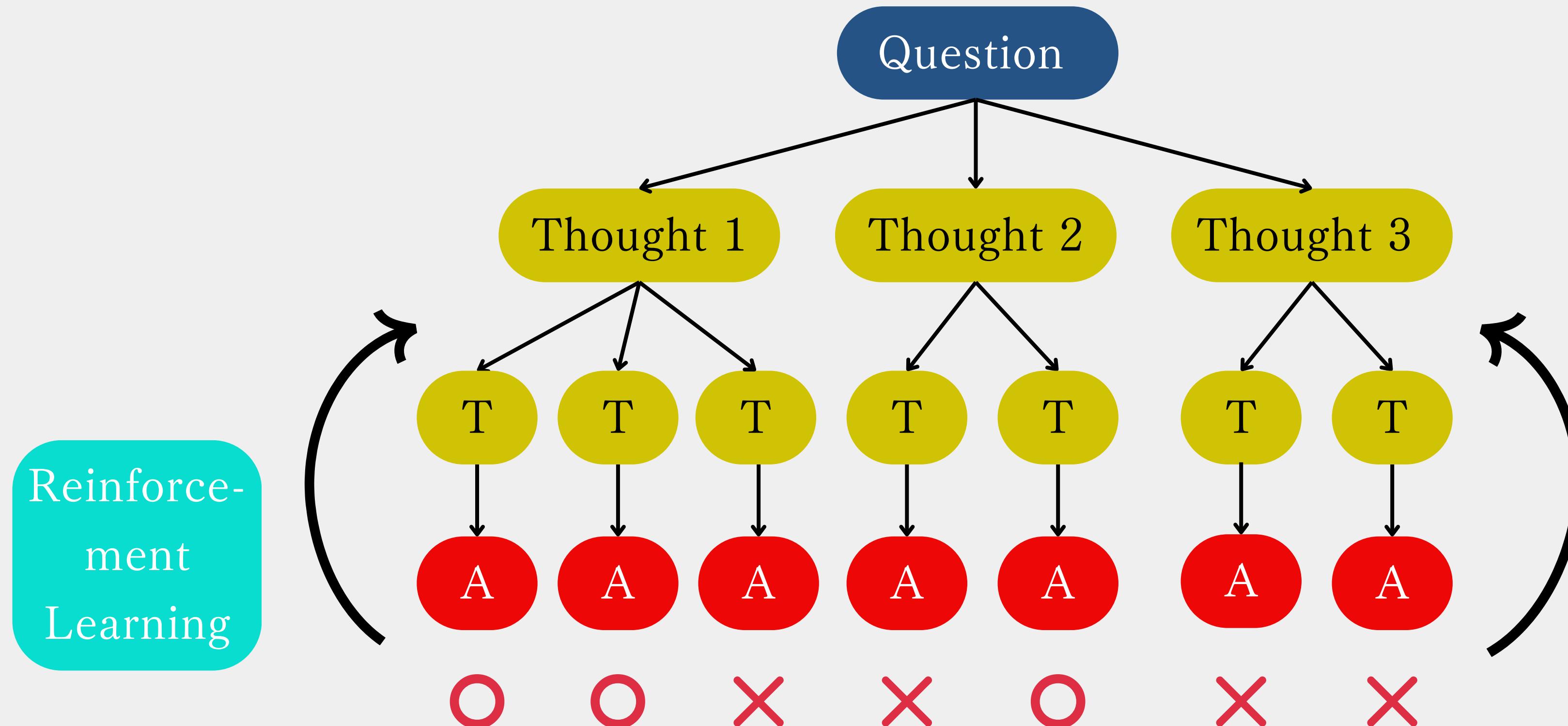
Let's compare AI Agent model of reasoning
with Deepseek-r1 training method [[DeepSeek-AI arXiv:2501.12948](#)]



AI Agent Model



Deepseek Model



Strength of DeepSeek-r1

Learn by Practice: Reinforcement Learning (RL) on Complex Tasks

But, AI Agent is not optimized through Reinforcement Learning



Weaknesses of Deepseek-rl

- ① Requires too many labeled dataset
- ② RL of LLM is too expensive...
- ③ It's not designed for a specific AI Agent system...

=> Each company cannot make it on its own



03

Kyoto AI Model SEIMEI

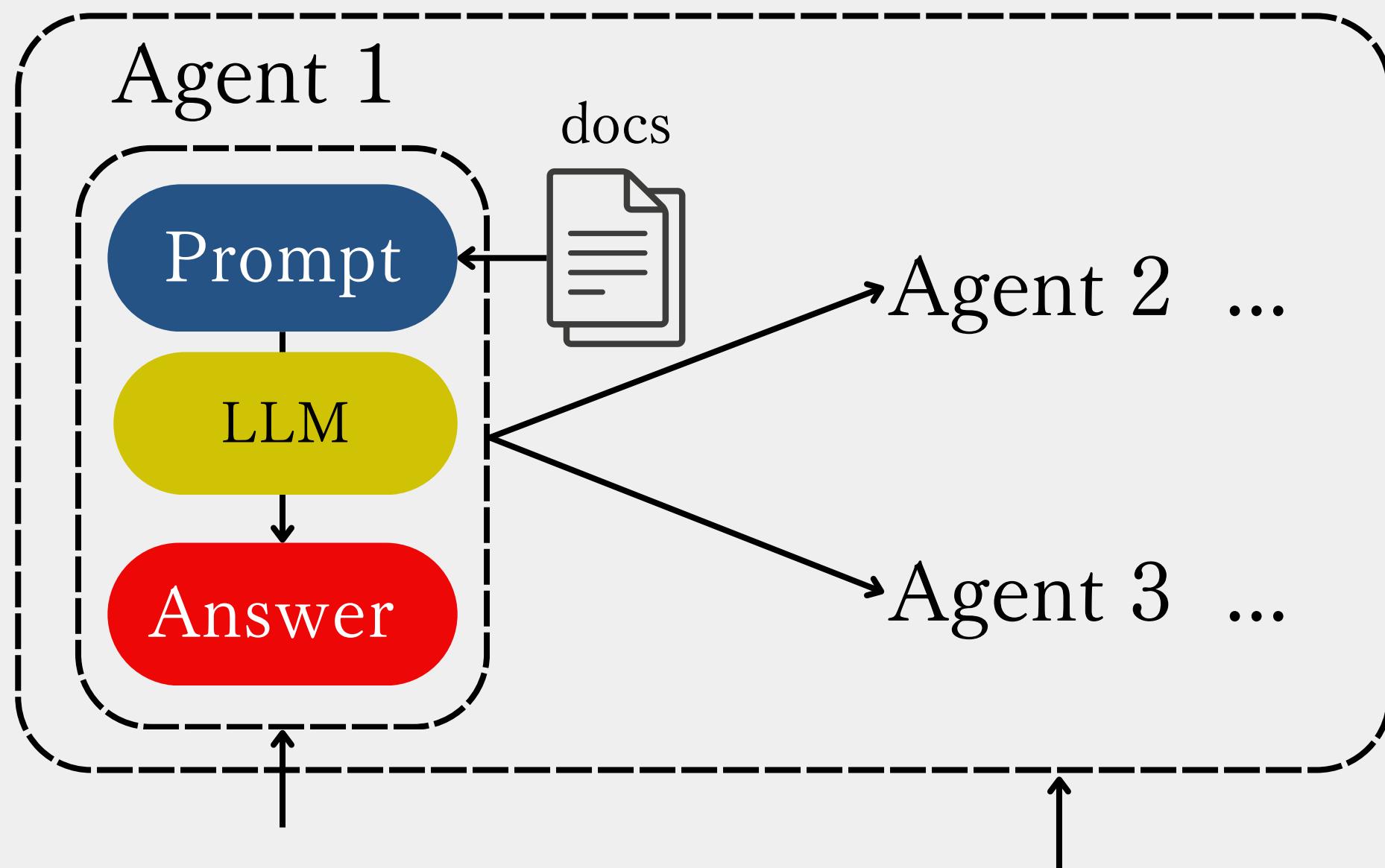
Idea

What if we integrate AI Agents by **Search Engine**?



Structure of Agent

Whole System



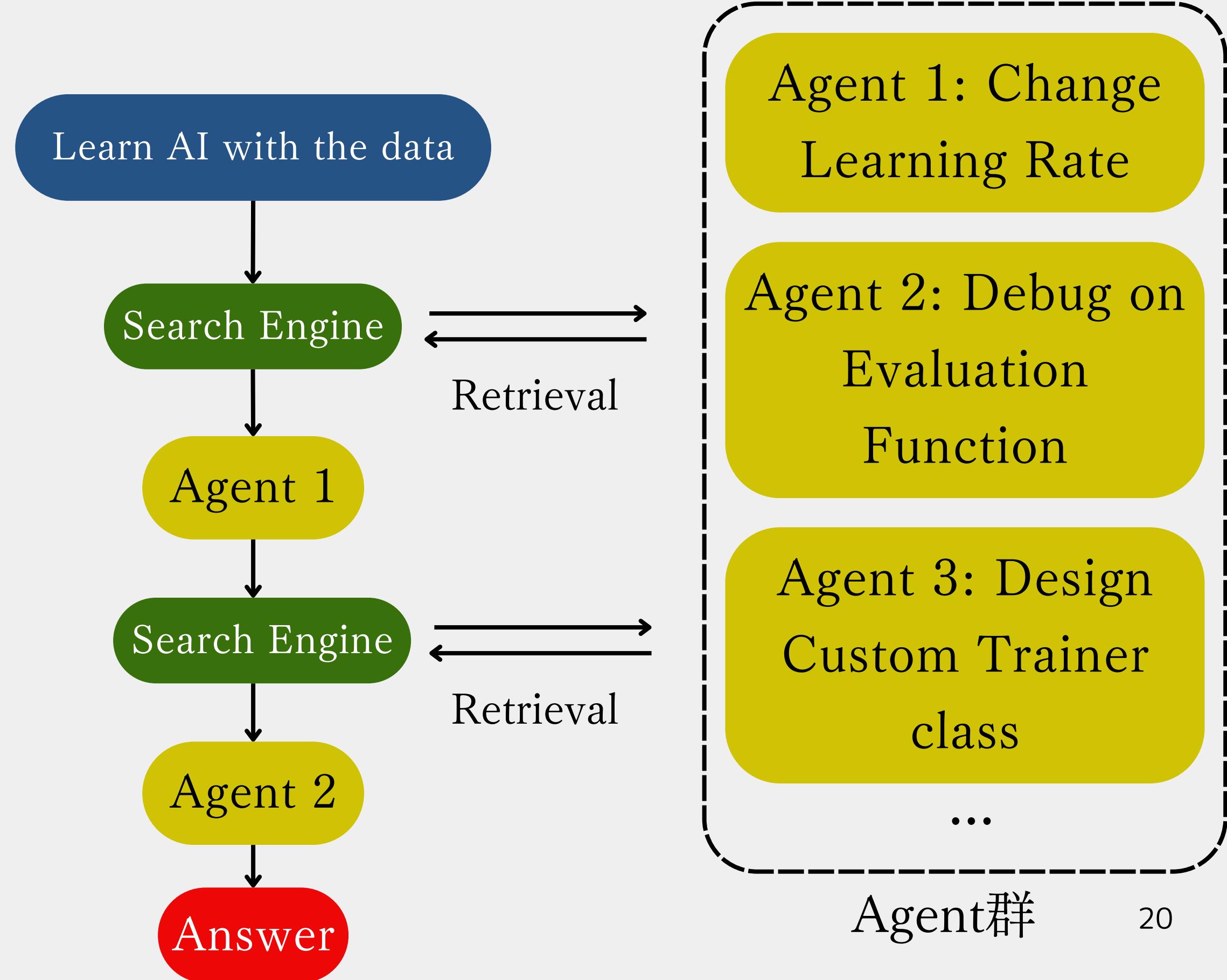
Deepseek-r1
optimized

No RL yet...

Reinforcement Learning on AI
Agent Path may improve the
system!!

Example

Search Engine decides
the **path of chain of
thoughts**



We call this model **SEIMEI**

(Search-Engine-Integrated Multi-Expert Inference)

Strength of SEIMEI

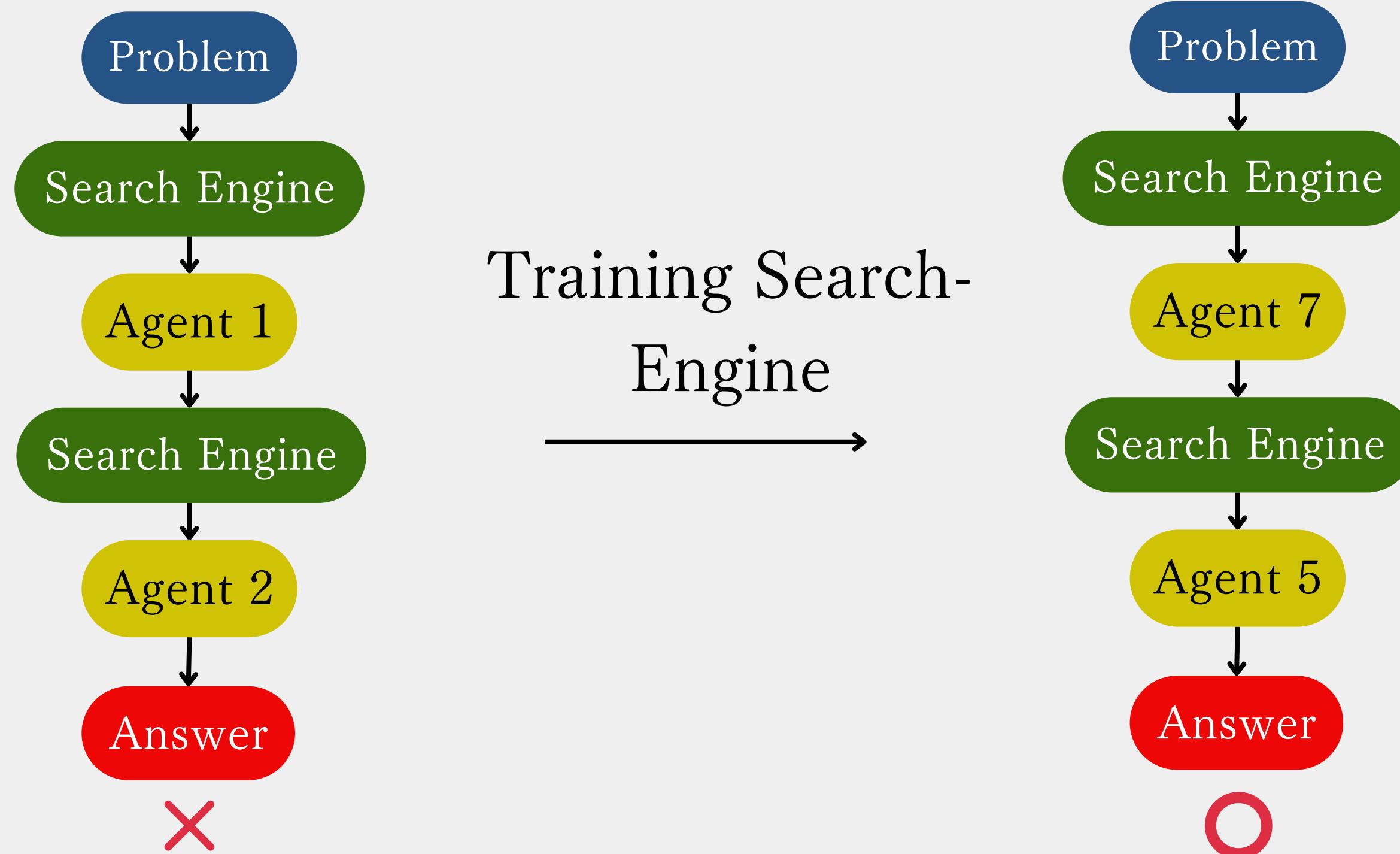
- ① Reasoning path is optimized by training search engine (like Deepseek-r1)
- ② System is **directly** improved by adding Agents
- ③ Human can modify the agent (**More open AI system**)
- ④ From Agentless to **Agentmore**
- ⑤ Training Search Engine is **1000~10000** times cheaper than LLM

=> SEIMEI lets a lot more people be able to create
AI system optimized for their tasks!!!!



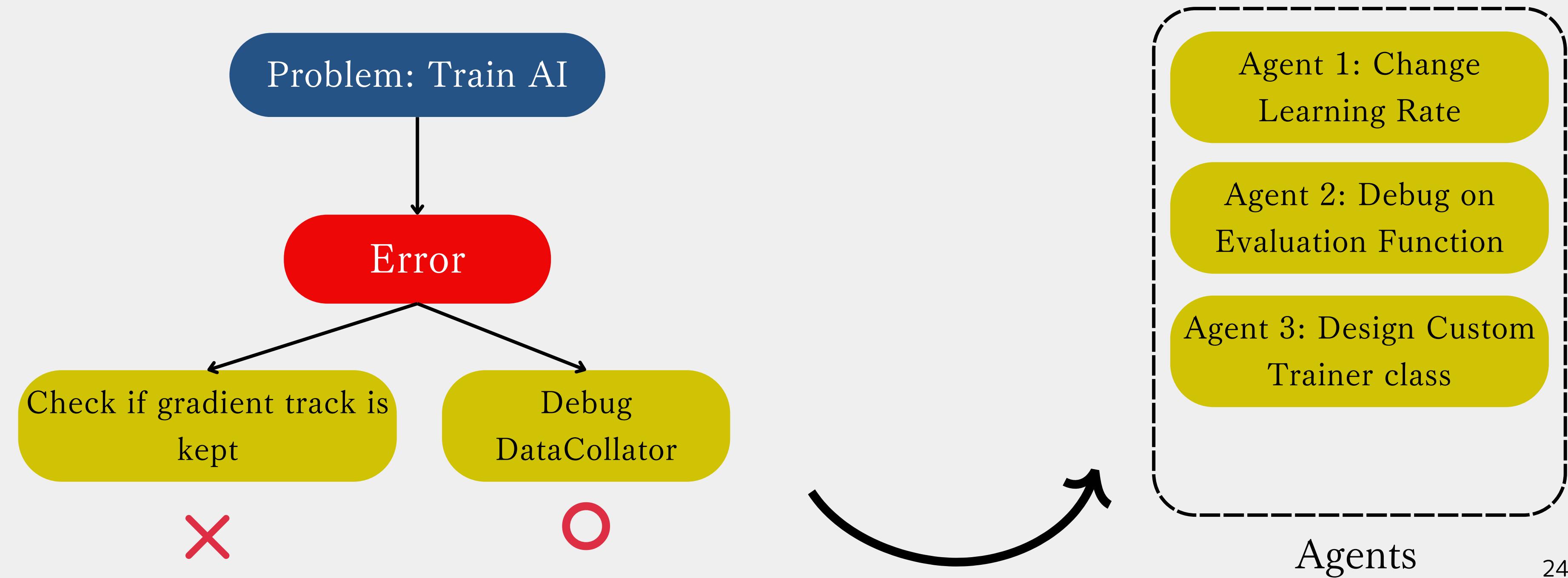
Strength of SEIMEI ①

Optimize Thinking Process by Training Search Engine



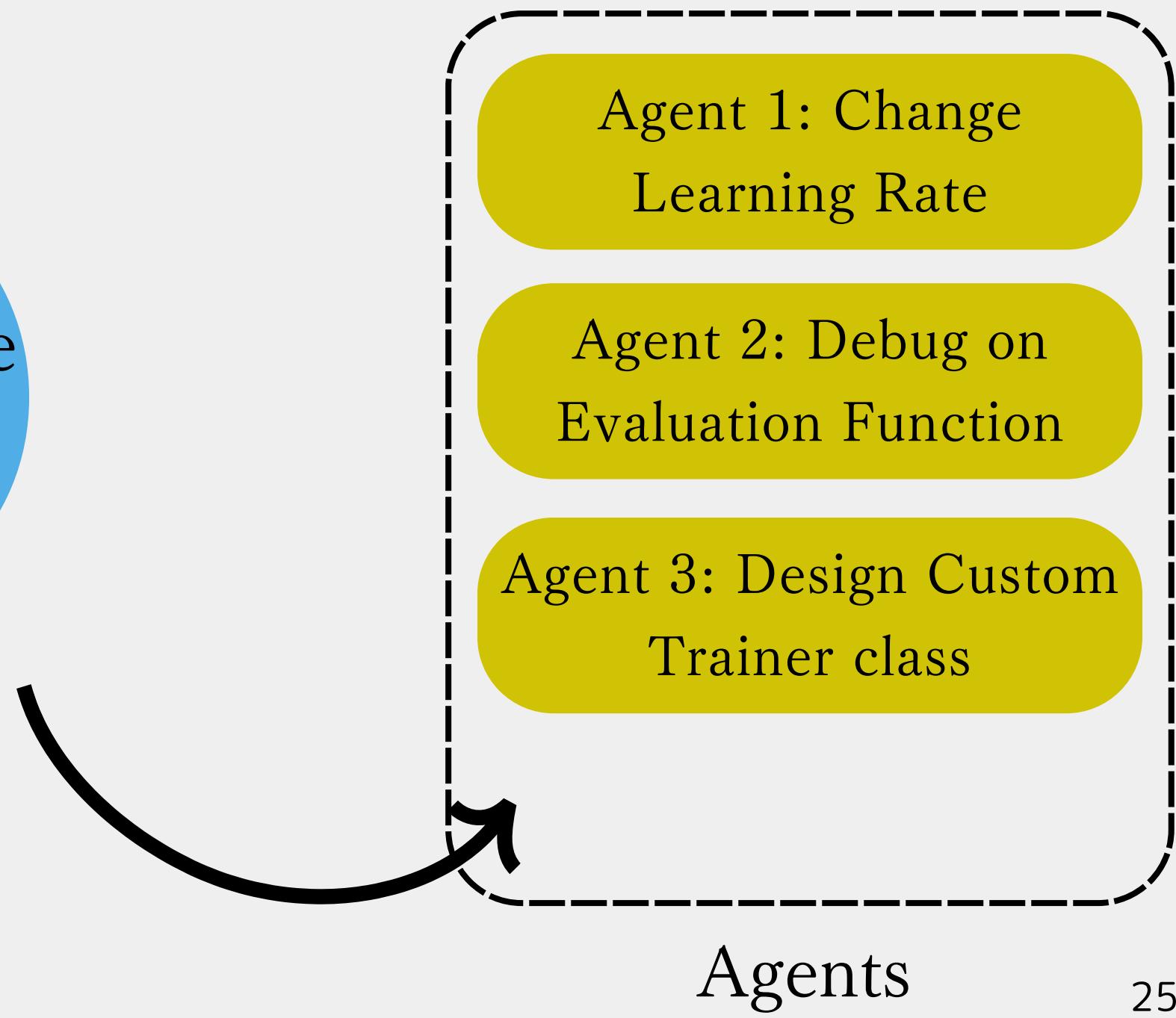
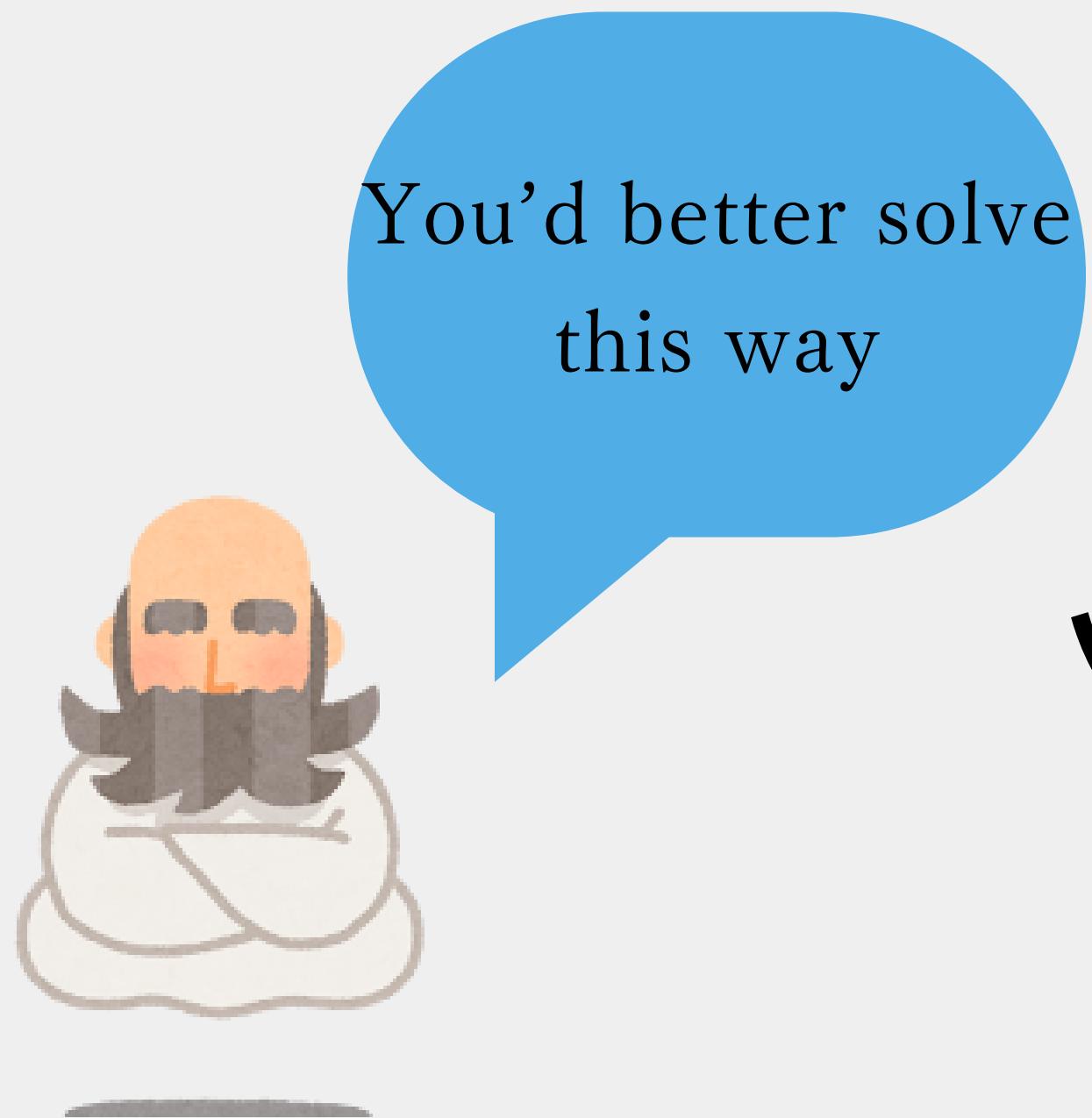
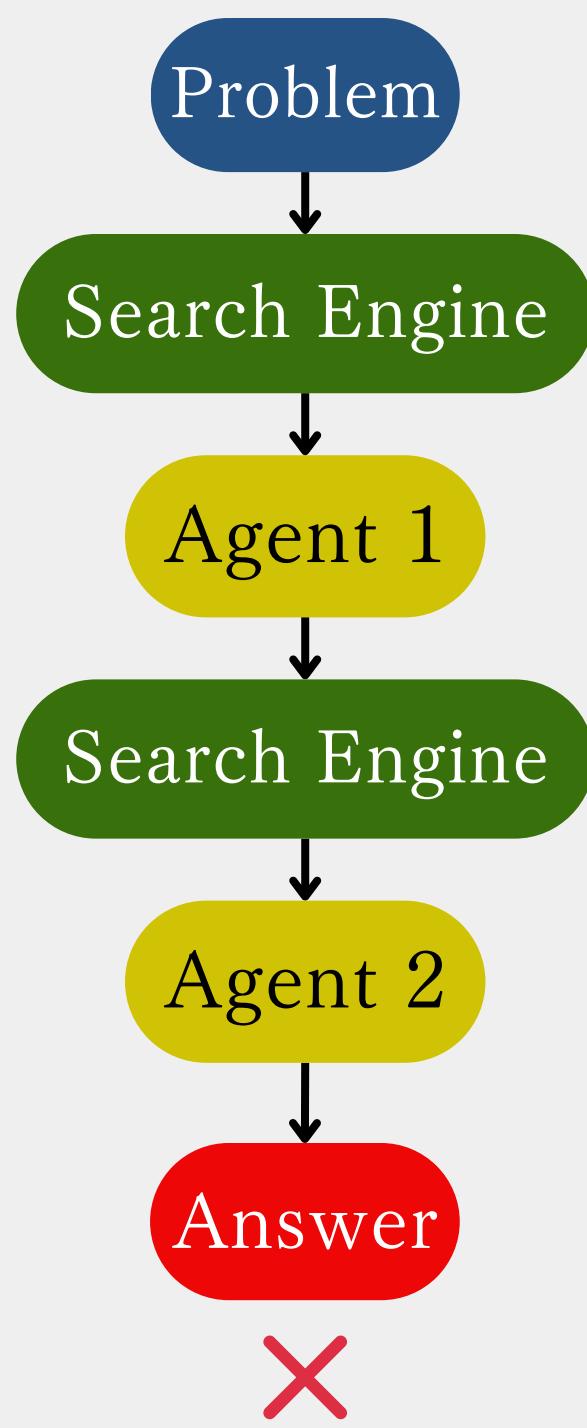
Strength of SEIMEI ②

Learn More Directly by Adding Agent



Strength of SEIMEI ③

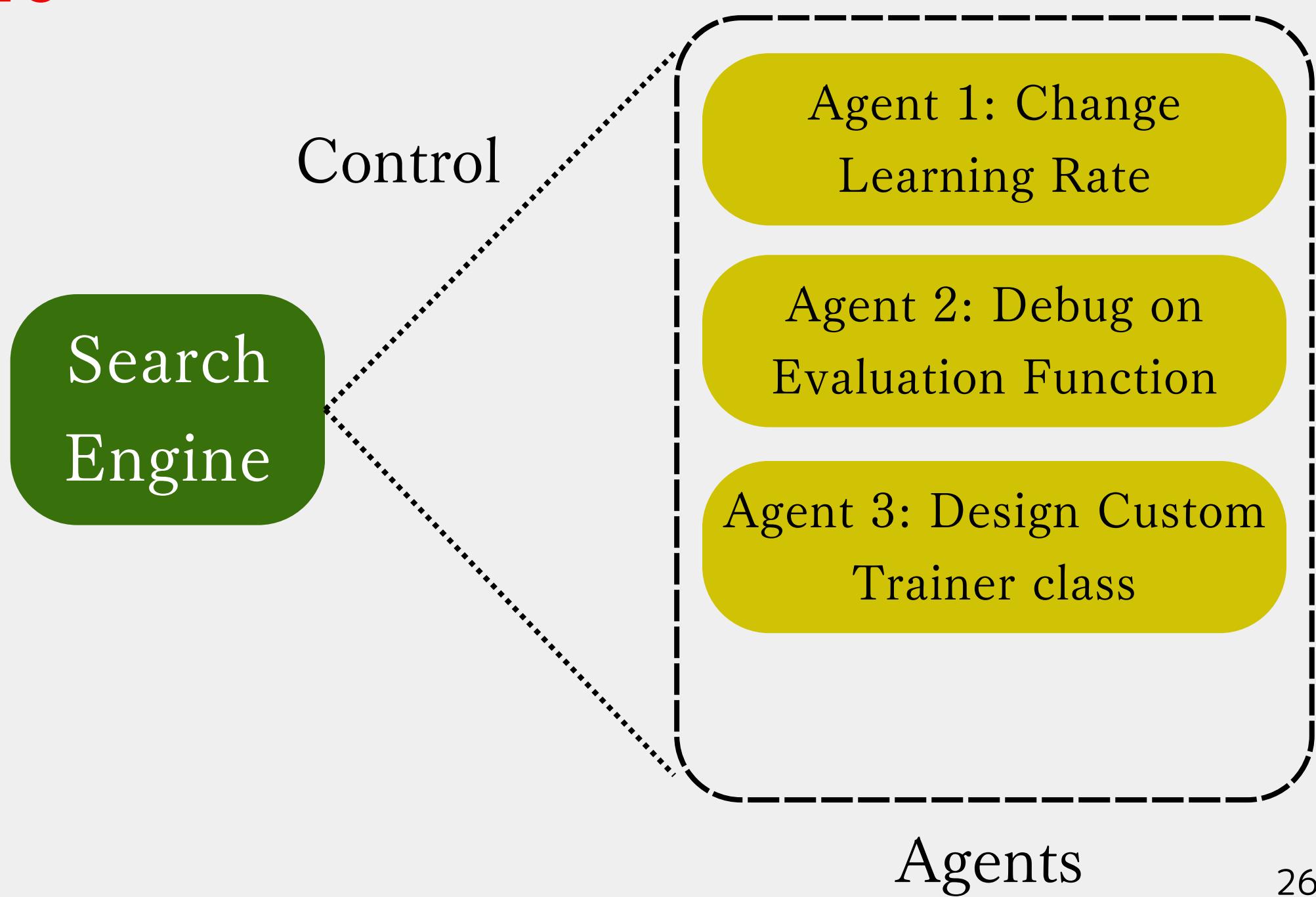
Human Expert can Improve the model through Agent



Strength of SEIMEI ④

From Agentless to **Agentmore**

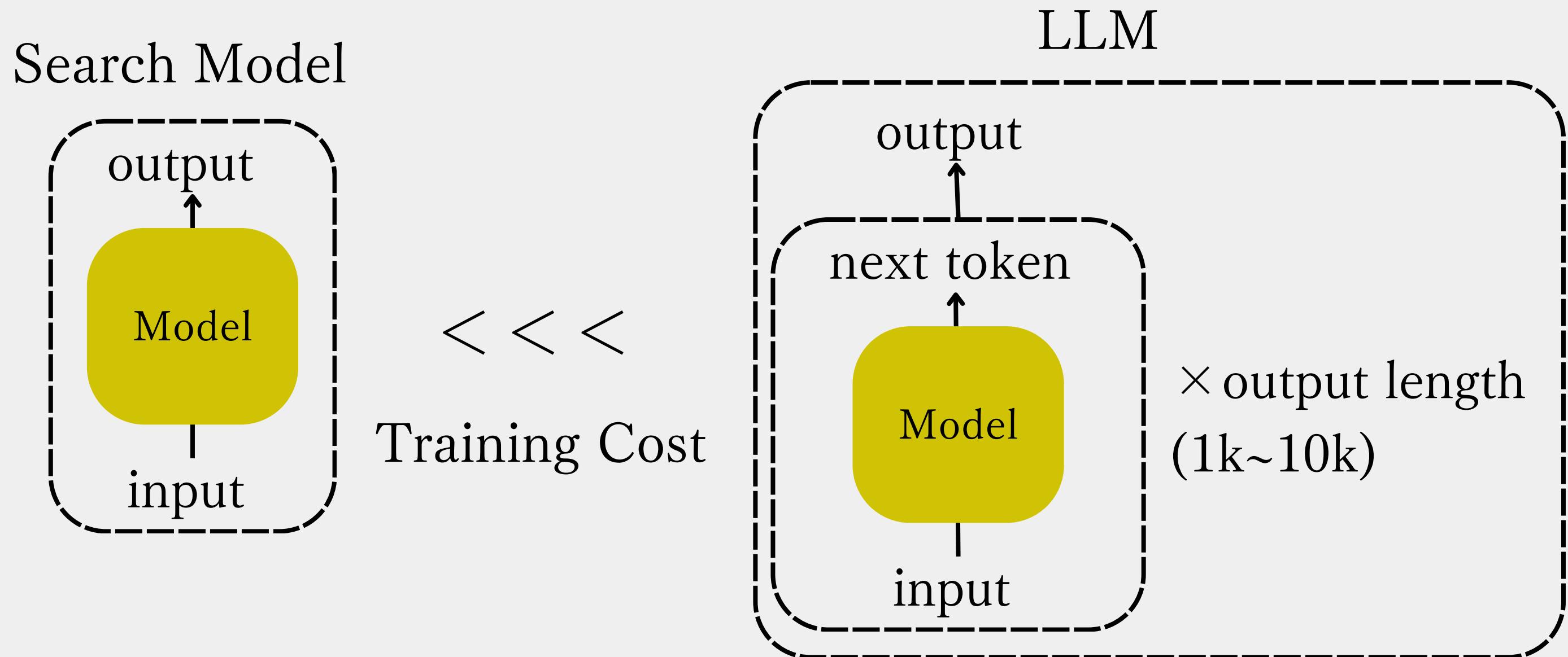
Overcome the instability of more agents by search engine controlling the agents



Strength of SEIMEI ⑤

Training Cost : Search Model <<< LLM

Even some
children can
train search
model with
their pocket
money



04

R e s e a r c h R e s u l t

Research Result Summary

Result of Training Search Model in SEIMEI

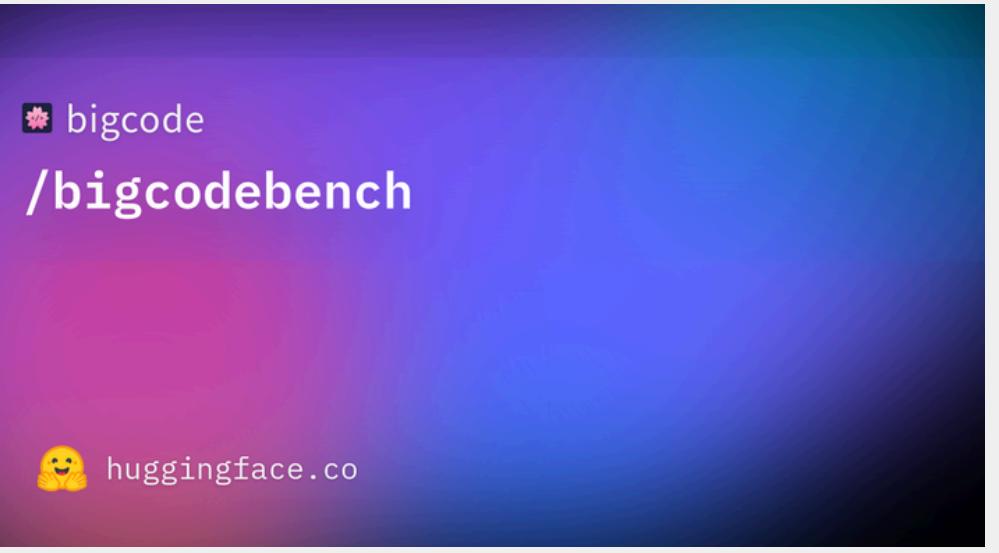
3 Key Discoveries:

1. Proprietary Search Model Achieves Stable Training
2. Proprietary Training Method Improves Evaluation Accuracy
3. Search Model Improves Code Generation by **16%** on
bigcodebench



Dataset

- bigcodebench : dataset for python coding tasks



The screenshot shows a Python code snippet for making an HTTPS GET request:

```
import http.client
import socket
import ssl

def task_func(server_name, server_port, path):
    """
    Makes an HTTPS GET request to a specified
    server and path, and retrieves the response.
    """
    Parameters: ...
    Returns: ...
    Raises: ...
    Requirements: ...
    Examples: ...
    """

    server_name (str): Name of the server
    to which the request is made
    server_port (int): Port number of the
    server to which the request is made
    path (str): Path to the HTTP request

    Returns
    str: Response body from the server

    Raises
    ssl.SSLError: on SSL handshake error
```

The interface includes sections for **Parameters**, **Requirements**, **Examples**, **Test Case Class**, and **Raises**.

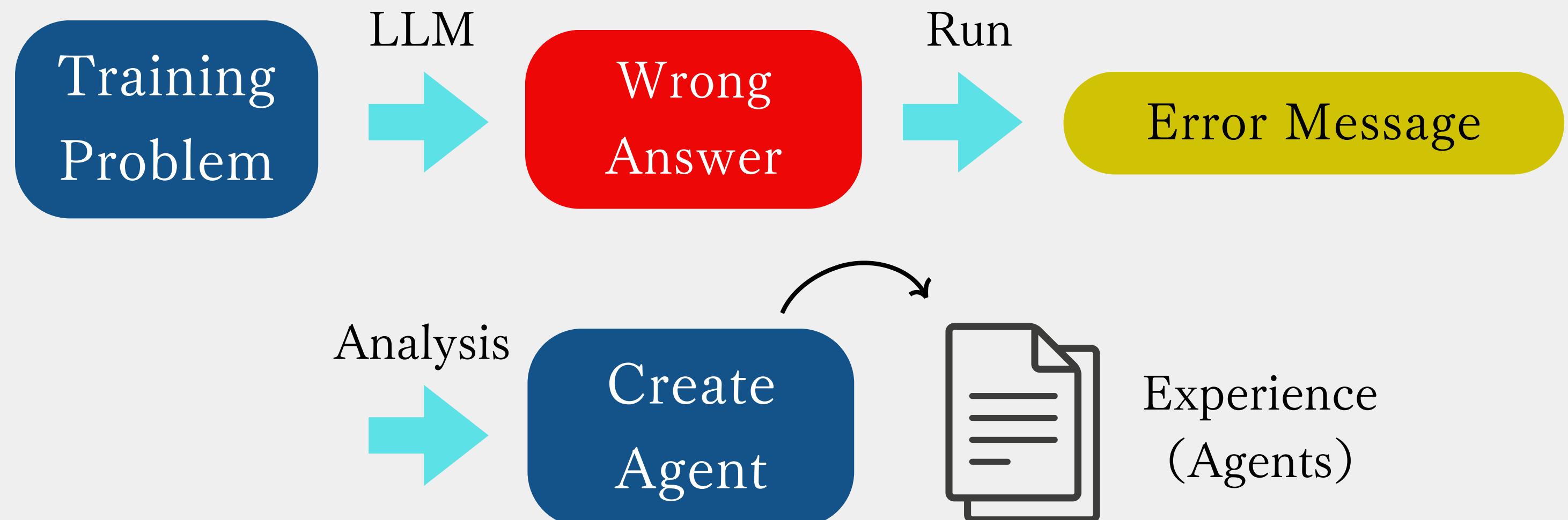
Contents

1. python code problems
2. correct answers
3. test code for checking
the answer

1000 problems are included

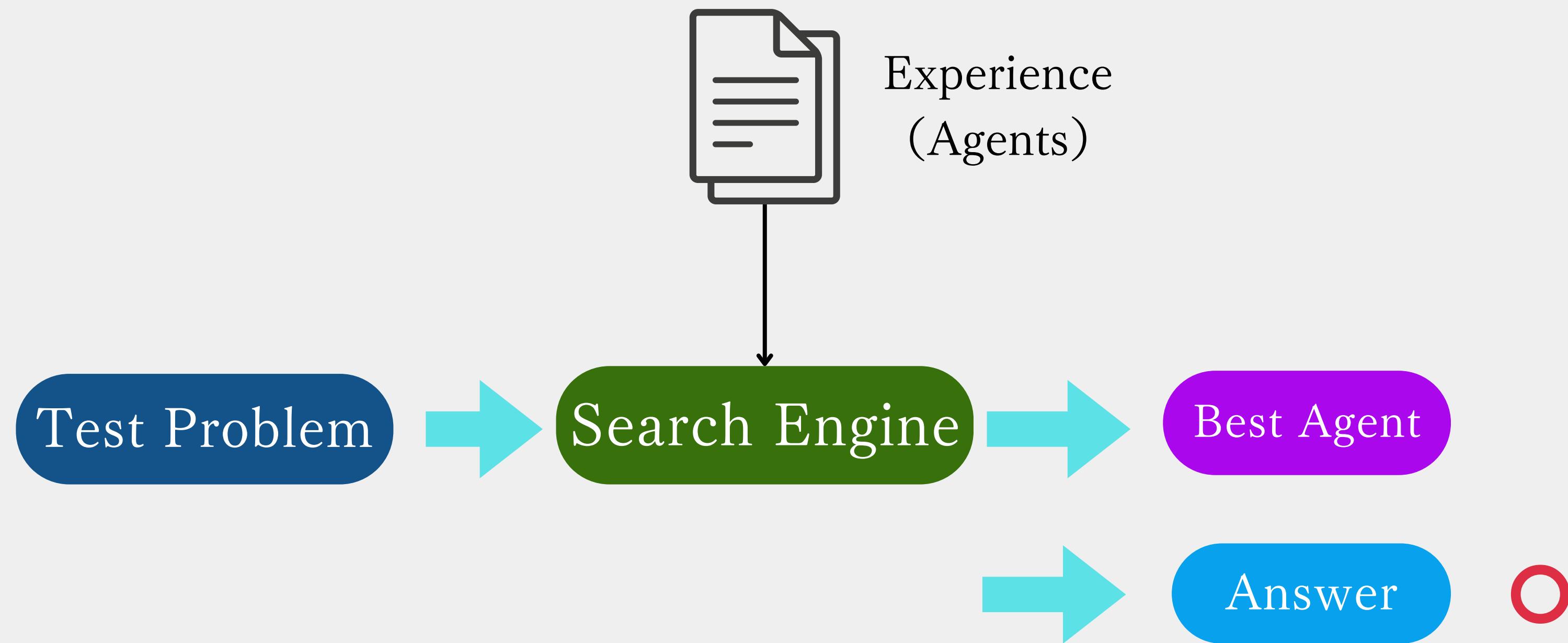
Method: Create Agent

- ① Error Message → Figure Out the Cause of Error
- ② Cause of Error → Create Agent which Leads to Answer



Method: Training Search Model

③ Train Search Model to Retrieve the Best Agent



Novelties

2 Novelties of KyotoAI Search Engine

1. We use different search model **other than semantic vector search** and **achieved stable training**
2. Our improved method of training shows **better evaluation accuracy** than current method

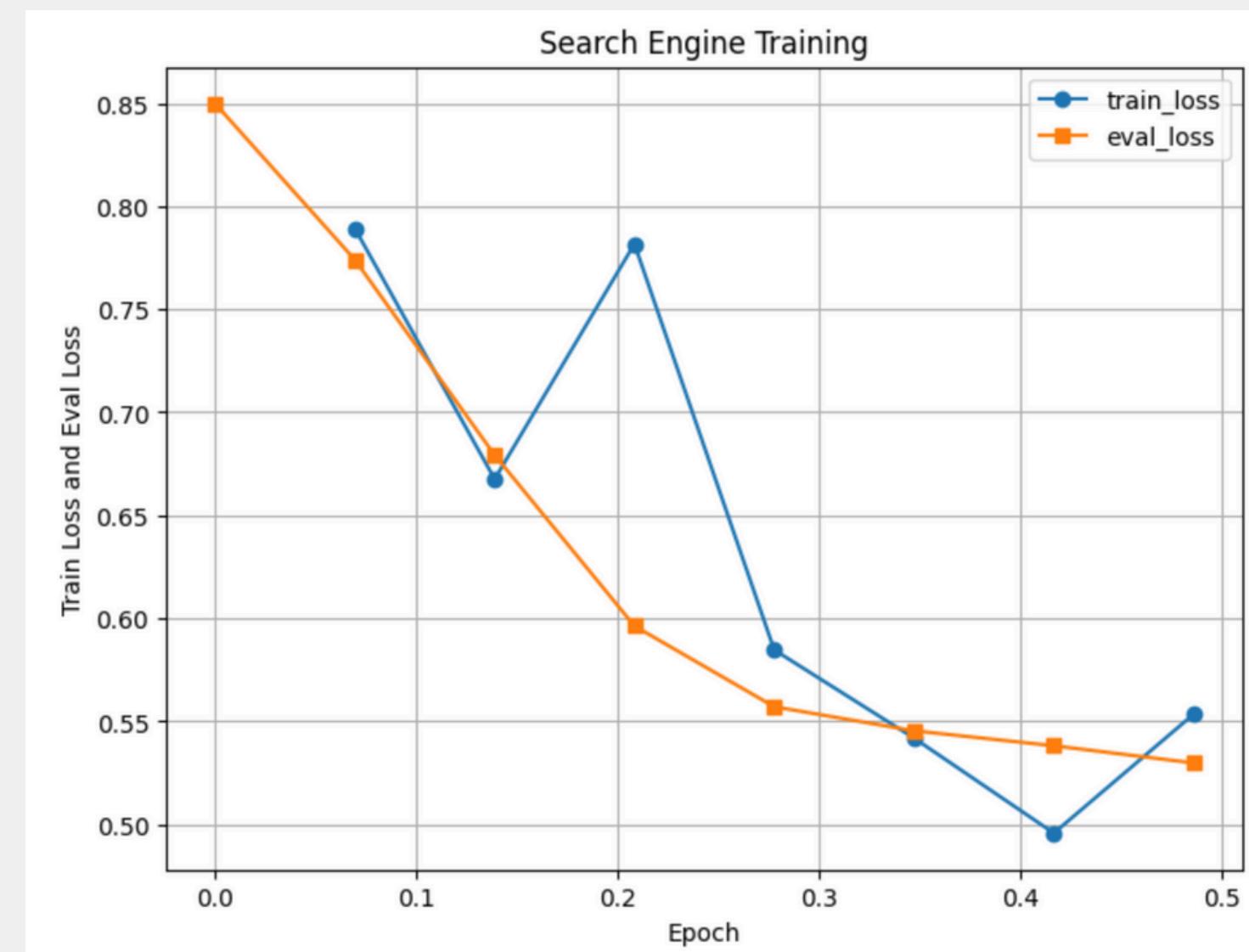
Result

1 . Stability of Our Proprietary Search Model

As the model learns from the training data (●), its performance on the evaluation data (■) also improves, demonstrating effective learning.



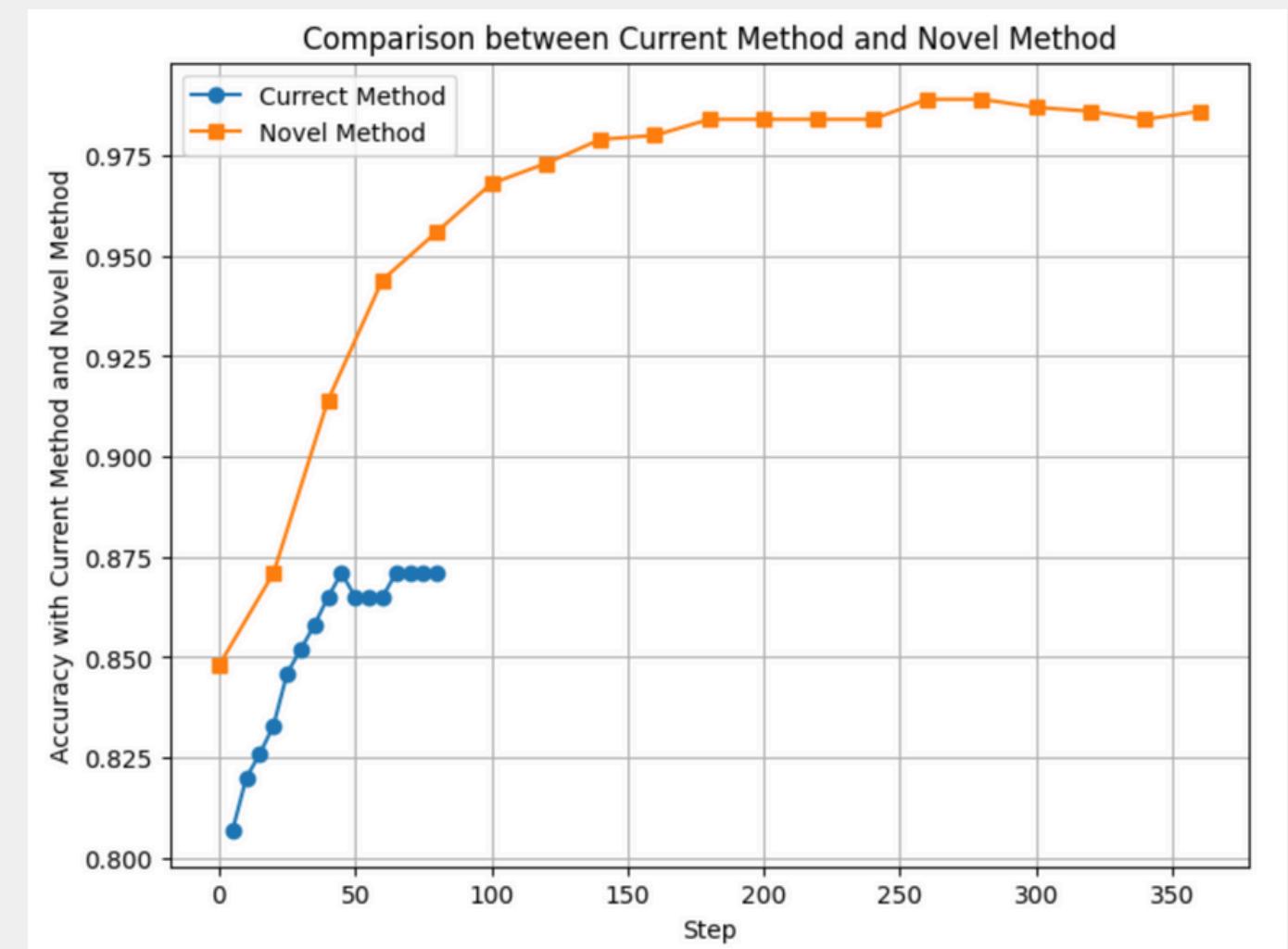
Stable Training



Result

2 . Improvement by Our Proprietary Model

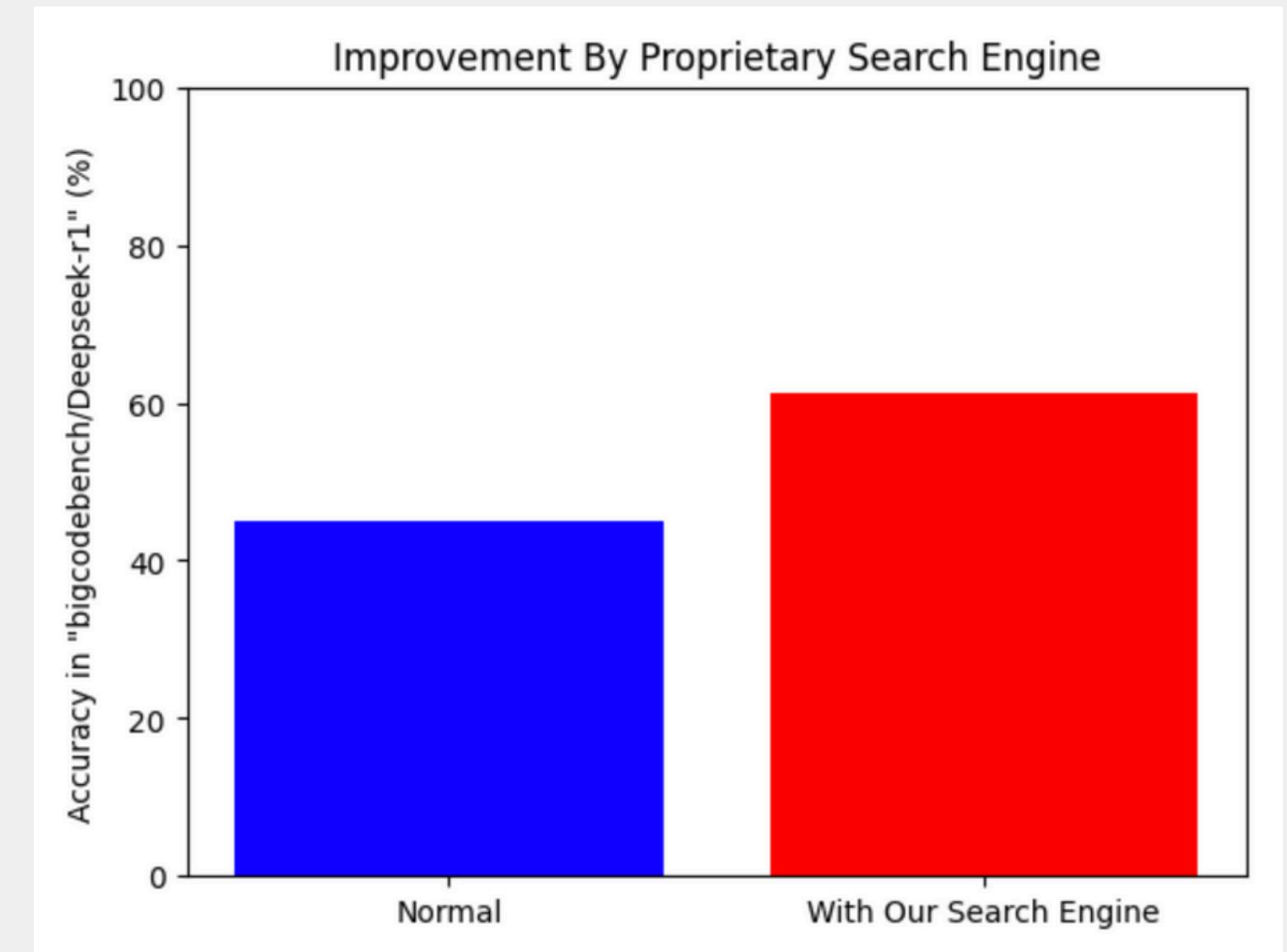
- The graph on the right compares the conventional method (●) with KyotoAI's proprietary efficient learning method (■) for search tasks.
- Using the same search model and data, **our proprietary method achieves a higher accuracy rate** compared to the conventional method.



Result

3 . Improve LLM by Trained Search Engine

- The graph on the right compares Deepseek-r1 without a search engine (■) to Deepseek-r1 enhanced with a trained search engine (■).
(Test data: bigcodebench)
- The AI agent enhanced with the search engine shows an improvement of approximately 16%.



05

Application

SEIMEI Application

SEIMEI covers AI Agent tasks that generally require more cognitive effort.

For example:

- Analyzing company documents
- Analyzing, auto-generating, and debugging code
- Conducting online surveys and proposing new business ideas
- Automatically creating presentations and writing papers

Debug

Currently in progress. Using datasets like SWE-benchmark to automatically generate AI Agents, we train the search model and evaluate the results, similar to previous research.

One of SEIMEI's advantages is that it can teach AI Agents and RAG search models to optimize based on human experience.

In the future, we aim to collaborate with debugging companies to achieve automated debugging.

Automatic Presentation Generation

To automatically generate presentation materials, it is necessary to convert HTML content into PPTX files using AI.

However, generating the XML content within PPTX files is challenging for language models. Therefore, we need to use an advanced AI agent (SEIMEI) to perform the complex conversion from HTML to XML.

Currently, we are exploring a joint venture with Company A to achieve this.

Company's Senior AI Advisor

We are considering implementing a system for Company B that analyzes all company documents to serve as a reliable senior AI advisor.

By leveraging SEIMEI's cognitive capabilities, we aim to uncover hidden connections within the documents and resolve uncertainties.

This research paves the way for a new era of company-specific trained AI.