Stanford TEXAS A&M UNIVERSITY UC San Diego Lambda

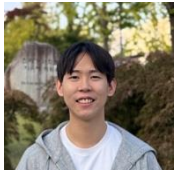
ICLR

AgentFlow

In-the-Flow Agentic System Optimization for Effective Planning and Tool Use

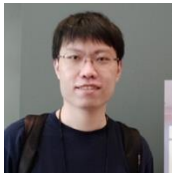

*Zhuofeng Li**


*Haoxiang Zhang**

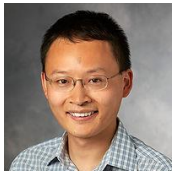

Seungju Han

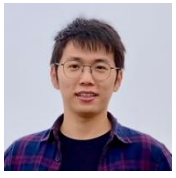

Sheng Liu


Jianwen Xie


Yu Zhang

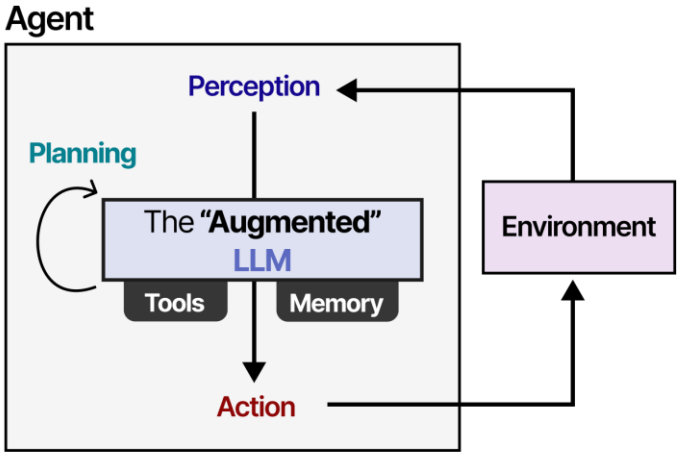

Yejin Choi


James Zou+


Pan Lu+*

Stanford Artificial Intelligence HAI Stanford University Human-Centered Artificial Intelligence Renaissance Philanthropy Office of Naval Research ONR Science & Technology KAIST

LLM Agents and Agentic Systems

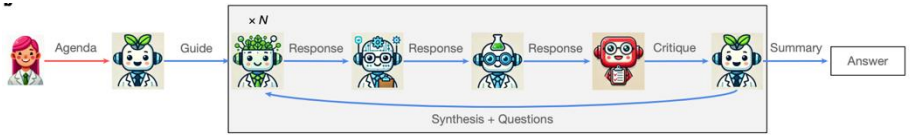


nature

Article | Published: 29 July 2025

The Virtual Lab of AI agents designs new SARS-CoV-2 nanobodies

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak & James Zou

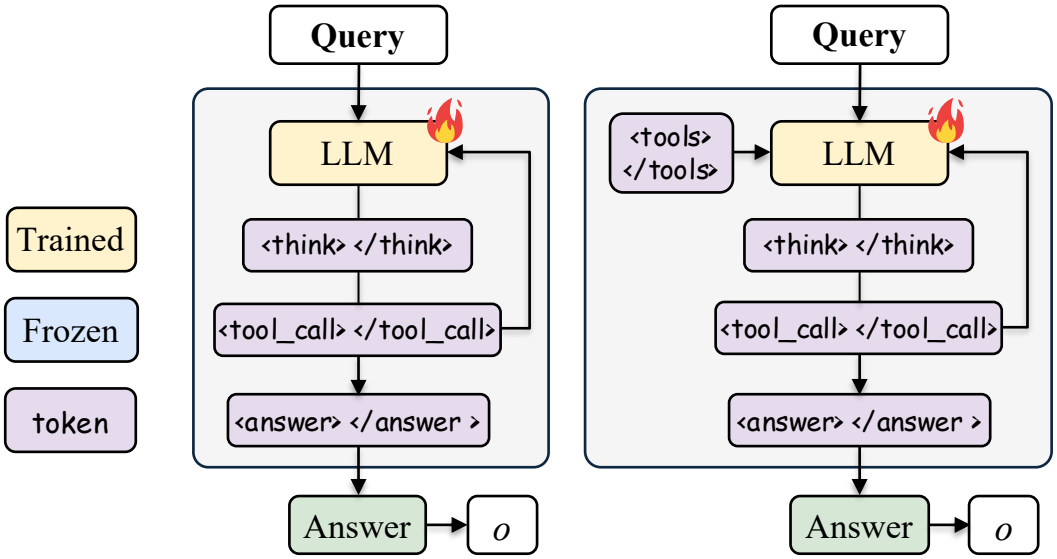


Biomni

A General-Purpose Biomedical AI Agent

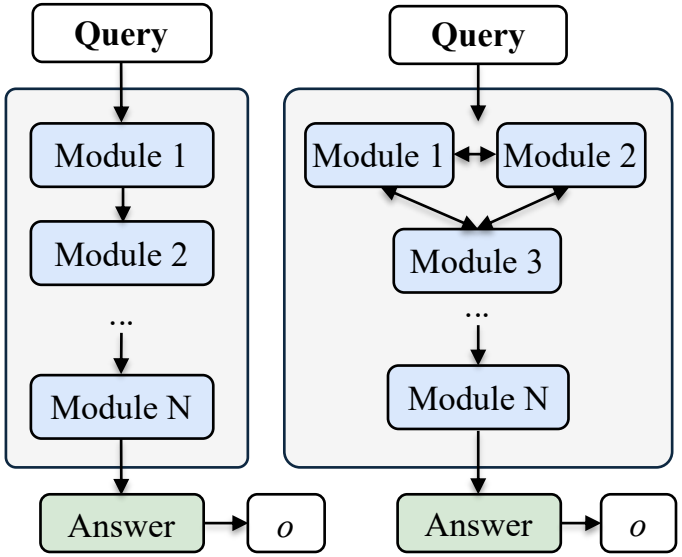
... with 150 specialized tools, 59 databases, and 105 software to automate

Two Paradigms of Tool-use LLMs



(a) Tool-Integrated Reasoning Models (LLM Agents)

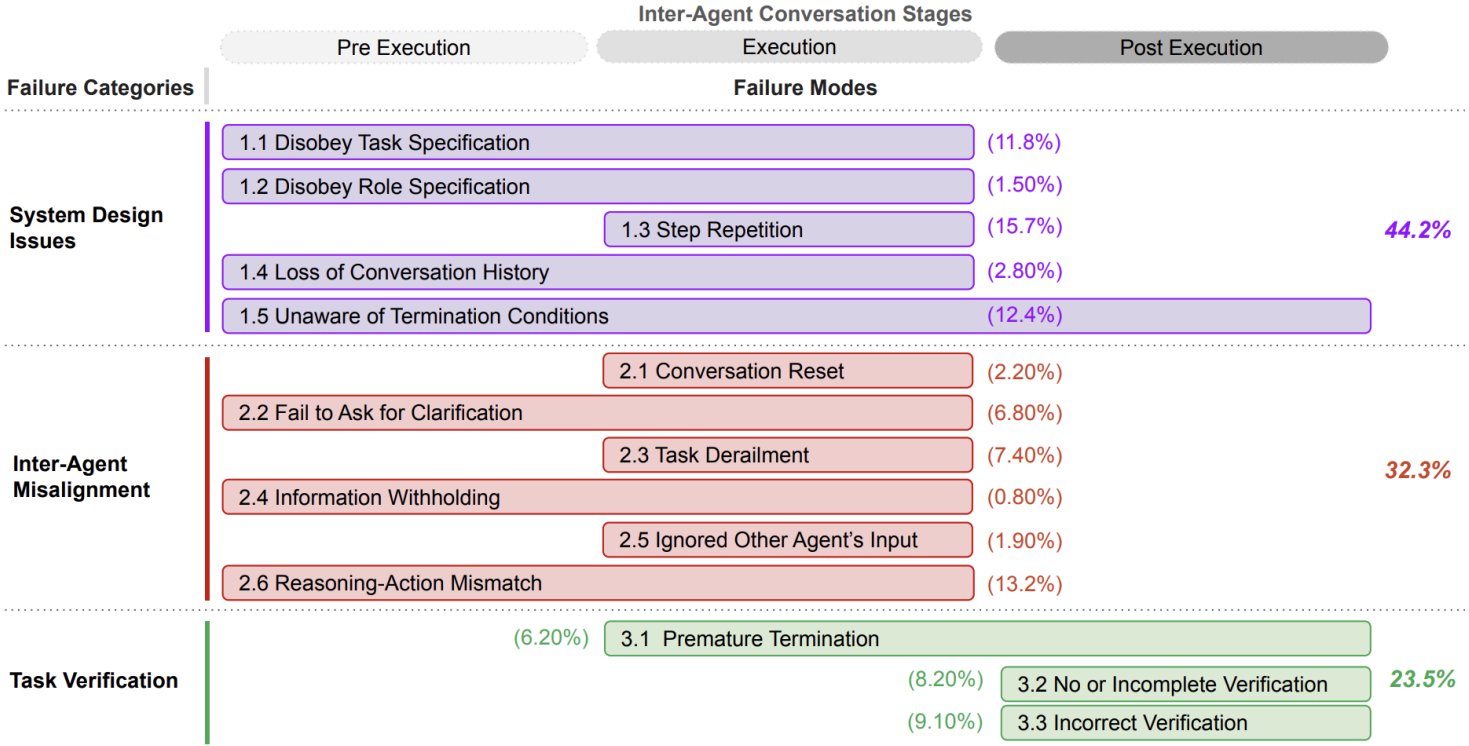
Two Paradigms of Tool-use LLMs



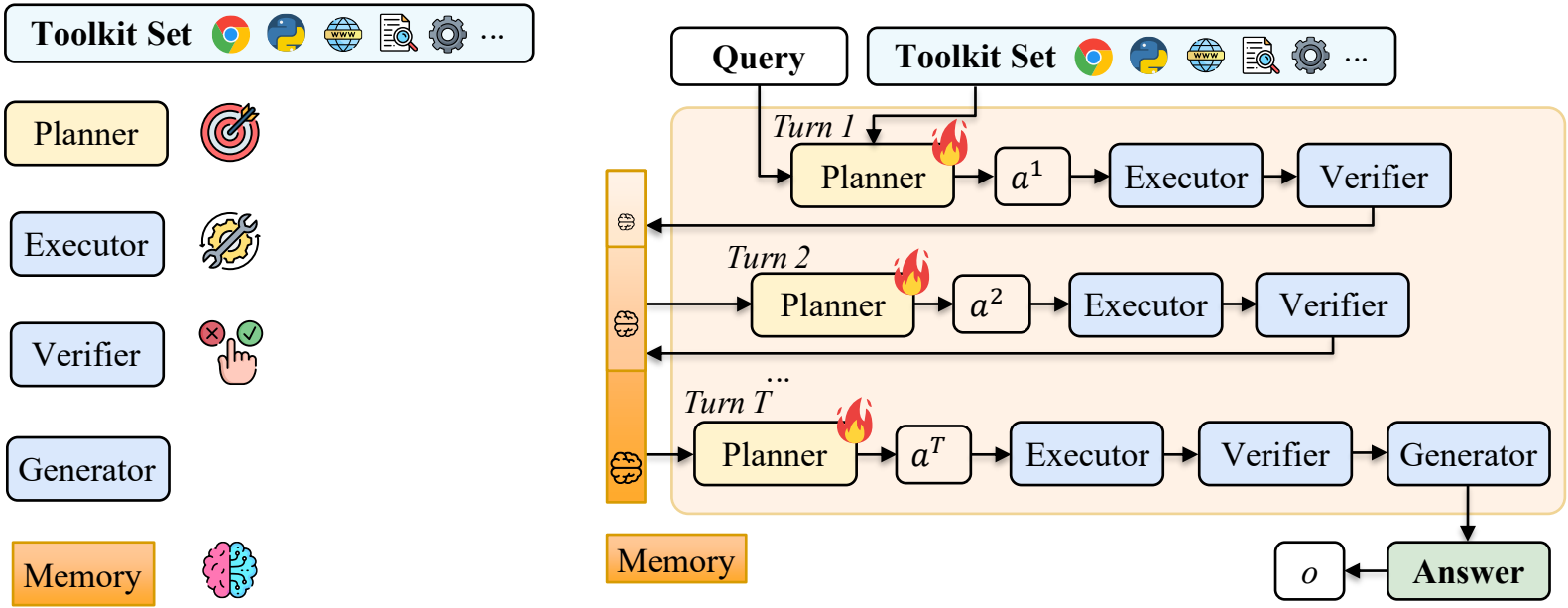
(b) Training-Free Agentic Systems

Why Do Multi-Agent LLM Systems Fail?

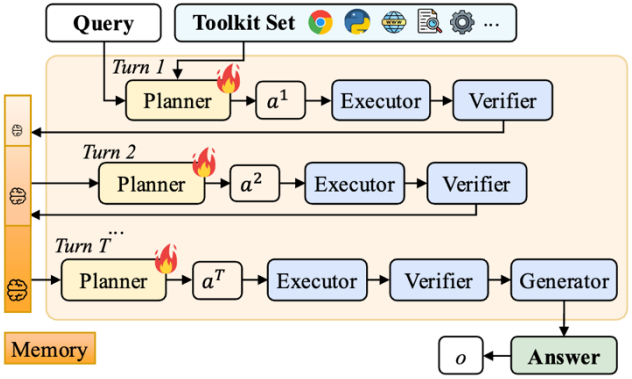
Mert Cemri[1*] Melissa Z. Pan[1*] Shuyi Yang[2*] Lakshya A Agrawal[1] Bhavya Chopra[1]
Rishabh Tiwari[1] Kurt Keutzer[1] Aditya Parameswaran[1] Dan Klein[1]
Kannan Ramchandran[1] Matei Zaharia[1] Joseph E. Gonzalez[1] Ion Stoica[1]
[1]UC Berkeley [2]Intesa Sanpaolo *Equal Contribution

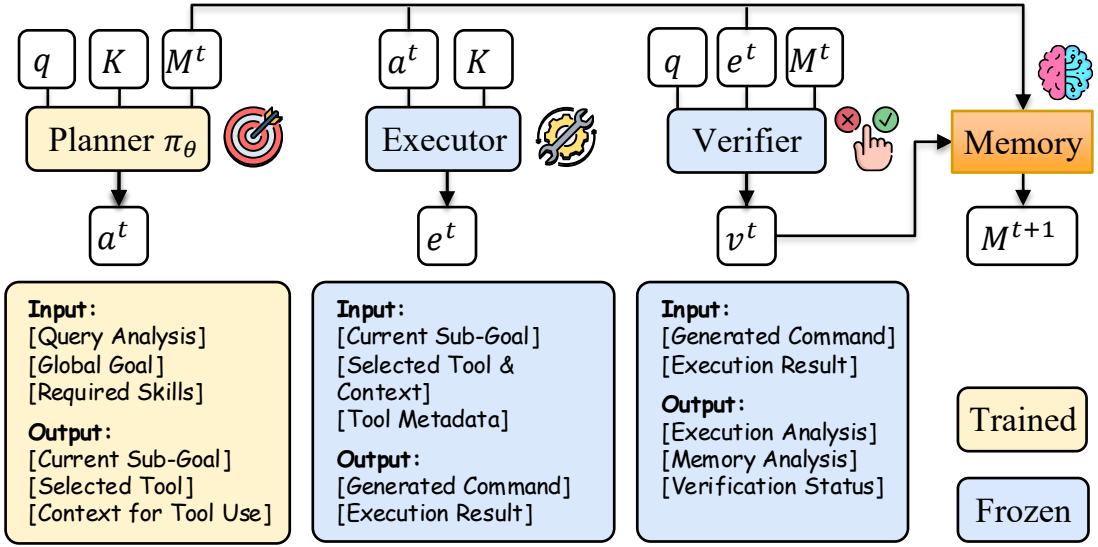


AgentFlow: An In-the-Flow Agentic System



AgentFlow: An In-the-Flow Agentic System





In-the-Flow Rollout at Turn t

Featured Tools

- Base Generator
- Python Coder
- Google Search
- Wikipedia Search
- Web Search

Base Generator

A generalized tool that takes query from the user, and answers the question step by step to the best of its ability. It can also accept an image.

INPUTS

`query` *str* - The query that includes query from the user to guide the agent to generate response.

OUTPUT

str - The generated response to the original query

EXAMPLE USAGE

```
execution = tool.execute(query="Summarize the following text in a few lines")
```

Generate a short summary given the query from the user.

Metadata | Code

Featured Tools

- Base Generator
- Python Coder
- Google Search
- Wikipedia Search
- Web Search

Python Coder

A tool that generates and executes simple Python code snippets for basic arithmetical calculations and math-related problems. The generated code runs in a highly restricted environment with only basic mathematical operations available.

INPUTS

`query` *str* - A clear, specific description of the arithmetic calculation or math problem to be solved, including any necessary numerical inputs.

OUTPUT

dict - A dictionary containing the generated code, calculation result, and any error messages.

EXAMPLE USAGE

```
execution = tool.execute(query="Find the sum of prime numbers up to 50")
```

Generate a Python code snippet to find the sum of prime numbers up to 50.

Metadata | Code

Featured Tools

- 💡 Base Generator
- 🐍 Python Coder
- 🔍 Google Search
- 📚 Wikipedia Search
- 🌐 Web Search

🔍 Google Search

A web search tool powered by Google Search that provides real-time information from the internet with citation support.

INPUTS

`query` *str* - The search query to find information on the web.

`add_citations` *bool* - Whether to add citations to the results. If True, the results will be formatted with citations. By default, it is True.

OUTPUT

str - The search results of the query.

EXAMPLE USAGE

```
execution = tool.execute(query="What is the capital of France?")
```

Search for general information about the capital of France with default citations enabled.

Metadata | Code

Featured Tools

- Base Generator
- Python Coder
- Google Search
- Wikipedia Search
- Web Search

Wikipedia Search

A tool that searches Wikipedia and returns relevant pages with their page titles, URLs, abstract, and retrieved information based on a given query.

INPUTS

`query` *str* - The search query for Wikipedia.

OUTPUT

dict - A dictionary containing search results, all matching pages with their content, URLs, and metadata.

EXAMPLE USAGE

```
execution = tool.execute(query="What is the exact mass in kg of the moon")
```

Search Wikipedia and get the information about the mass of the moon.

Metadata | Code

Featured Tools

- Base Generator
- Python Coder
- Google Search
- Wikipedia Search
- Web Search

Web Search

A specialized tool for answering questions by retrieving relevant information from a given website using RAG (Retrieval-Augmented Generation).

INPUTS

`query` *str* - The search query for the website.

`url` *str* - The URL of the website to retrieve information from.

OUTPUT

str - The answer to the user's query based on the information gathered from the website.

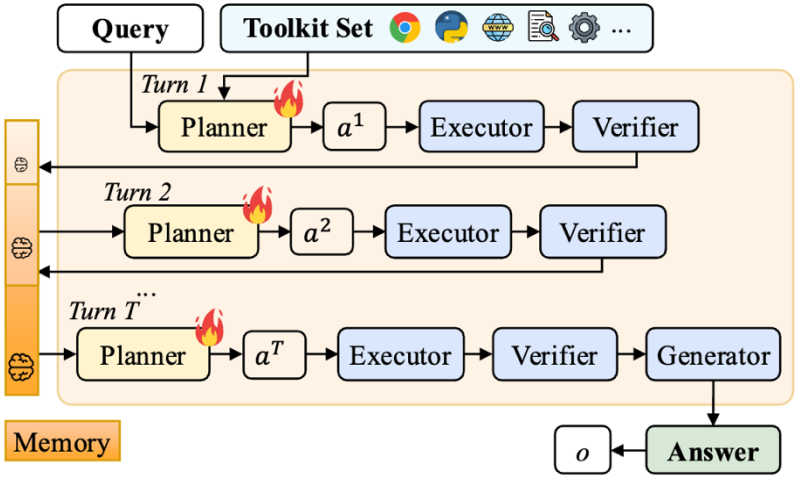
EXAMPLE USAGE

```
execution = tool.execute(query="What is the exact mass in kg of the moon?", url="https://en.wikipedia.org/wiki/Moon")
```

Retrieve information about the moon's mass from Wikipedia.

Metadata | Code

AgentFlow: A Glance (w/ Flow-GRPO Tuning)



Question: Compute the check digit the Tropicos ID for the Order Helotiales would have if it were an ISBN-10 number.
Ground truth answer: 3

1
- **Selected tool:** Wikipedia Search
- **Sub-goal:** Retrieve Tropicos ID of Order Helotiales from Wikipedia.
- **Tool call:** query="Tropicos ID for the Order Helotiales"
- **Result:** No results found for query: Tropicos ID for Order Helotiales.

2
- **Selected tool:** Google Search
- **Sub-goal:** Retrieve the Tropicos ID of the Order.
- **Tool call:** query="Tropicos ID for the Order Helotiales"
- **Result:** The Tropicos ID for the Order Helotiales is 100370510.

3
- **Selected tool:** Python Coder
- **Sub-goal:** Calculate the check digit for the Tropicos ID 100370510 if it were an ISBN-10 number.
- **Tool call:** query="Calculate check digit for ISBN-10 number 100370510"
- **Result:** name 'isbn' is not defined.

...

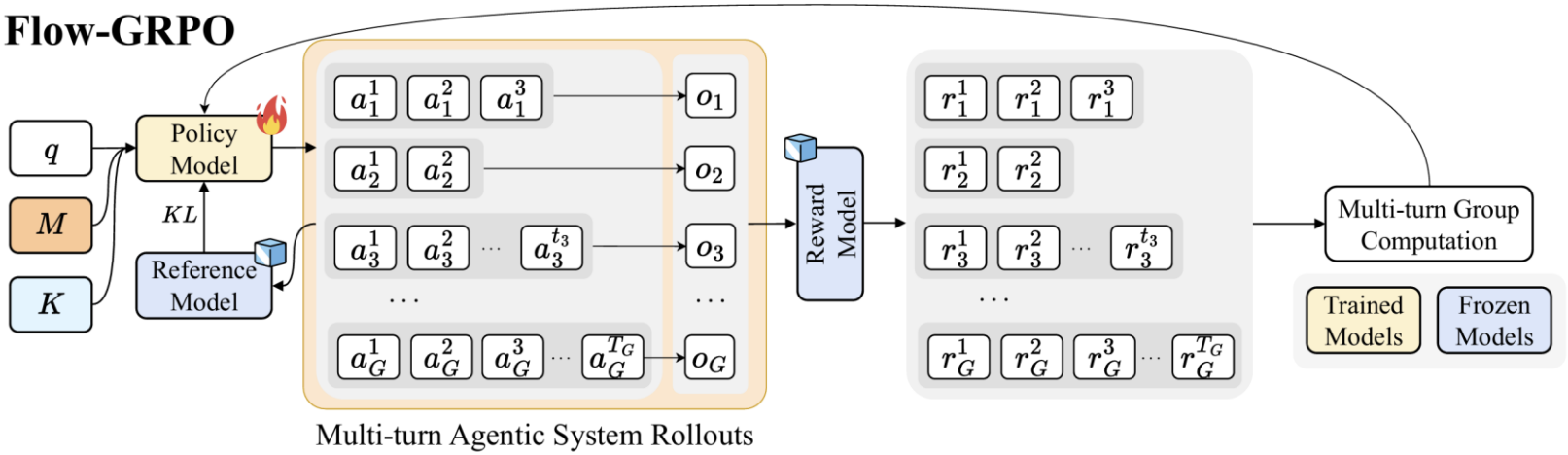
4-9
Action steps 3 to 9 *have similar sub-goals, commands, and results; the agent is stuck in error loops and unable fix the naming issue.*

...

Solution: The Python tool failed repeatedly with the same 'isbn' variable error). The agent failed to calculate the check digit.

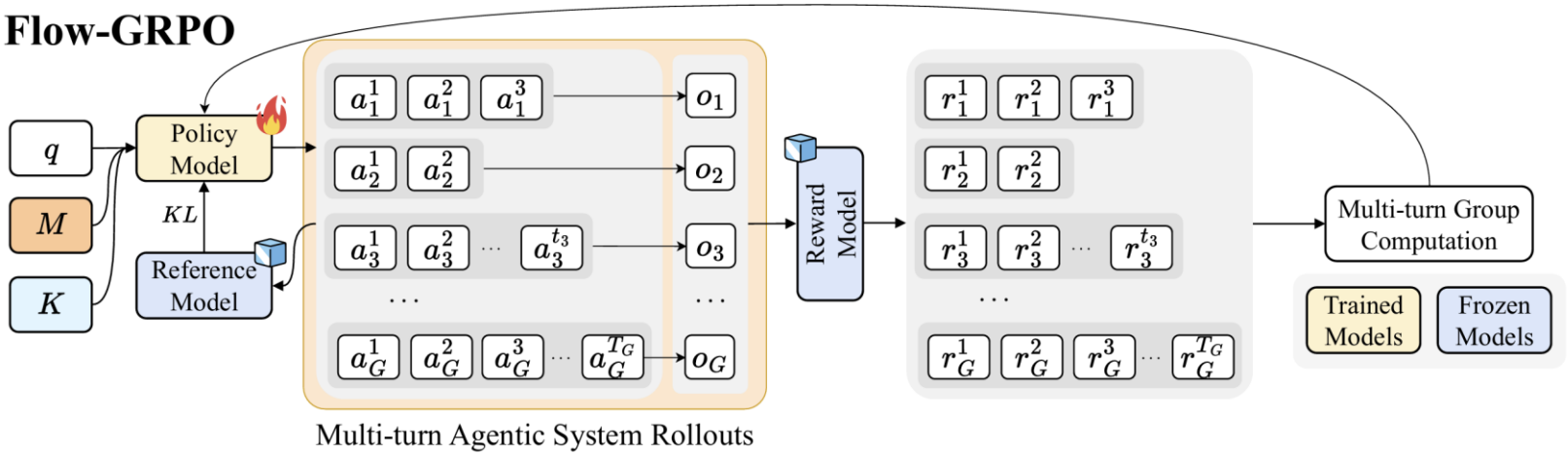
AgentFlow (**before** Flow-GRPO Fine-tuning)

MDP Process Form



$$p_\theta\left(\{a^t, e^t, v^t\}_{t=1}^{T},\ o \mid q\right) = \left[\prod_{t=1}^{T} \pi_\theta(a^t \mid q, K, M^t)\ \mathcal{E}(e^t \mid a^t, K)\ \mathcal{V}(v^t \mid q, e^t, M^t)\right] \mathcal{G}(o \mid q, M^T)$$

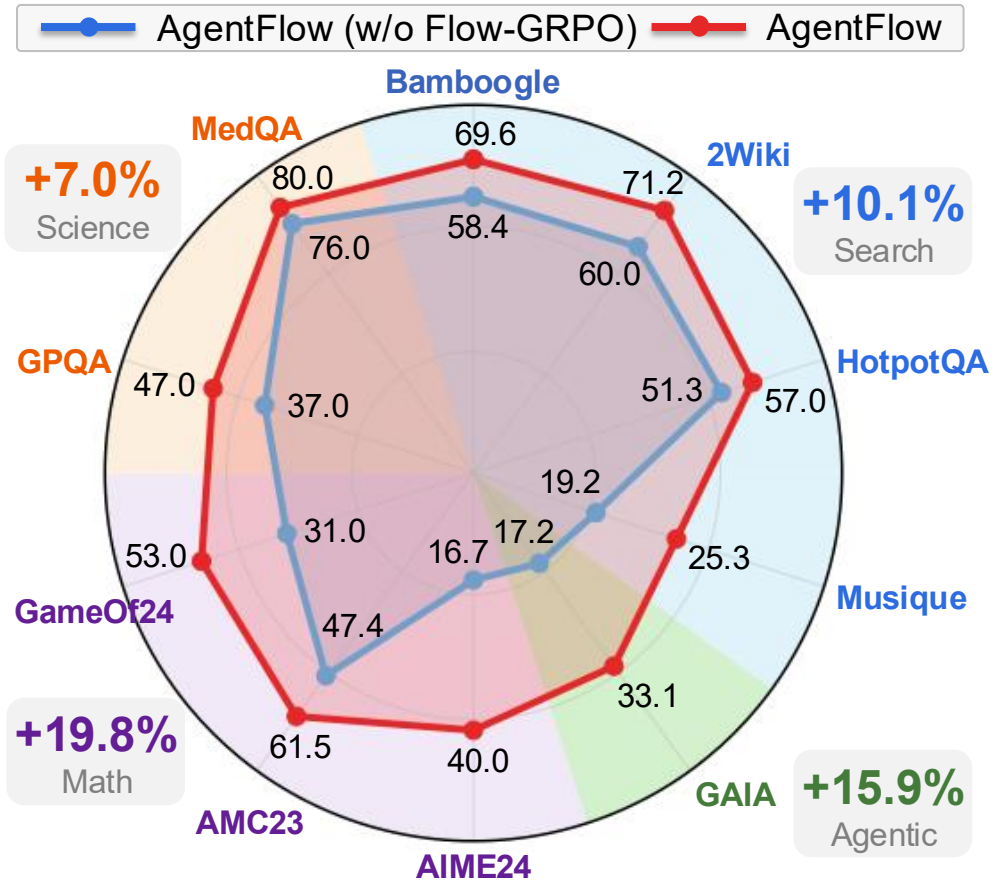
In-the-Flow Group Relative Policy Optimisation



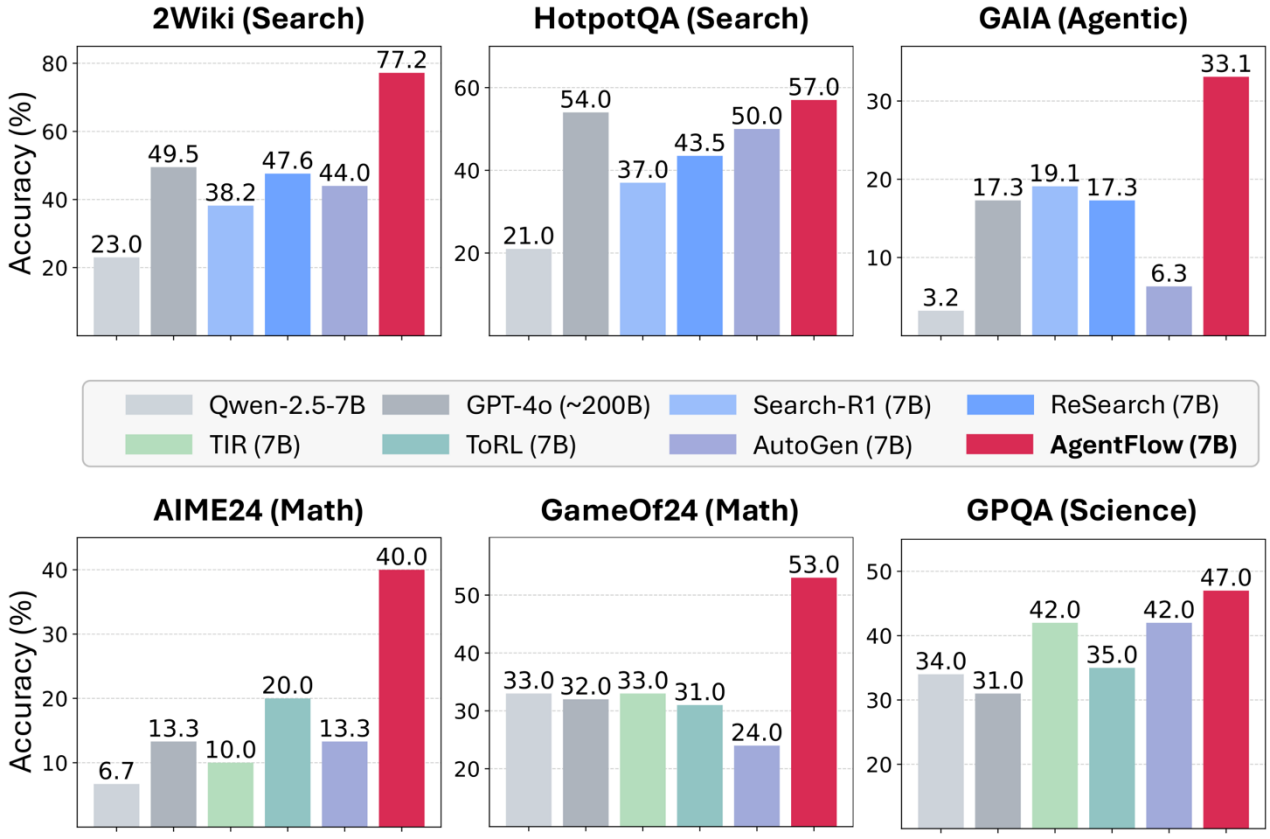
$$\mathcal{J}_{\text{Flow-GRPO}}(\theta) = \mathbb{E}_{(q,y^*)\sim\mathcal{D},\ \{\tau_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}}$$

$$\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\frac{1}{|a_i^t|}\sum_{j=1}^{|a_i^t|}\min\left\{\rho_{i,j}^t A_i^t,\ \text{clip}(\rho_{i,j}^t,\ 1-\epsilon,\ 1+\epsilon)\, A_i^t\right\} \ -\ \beta\,\mathbb{D}_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right)\right]$$

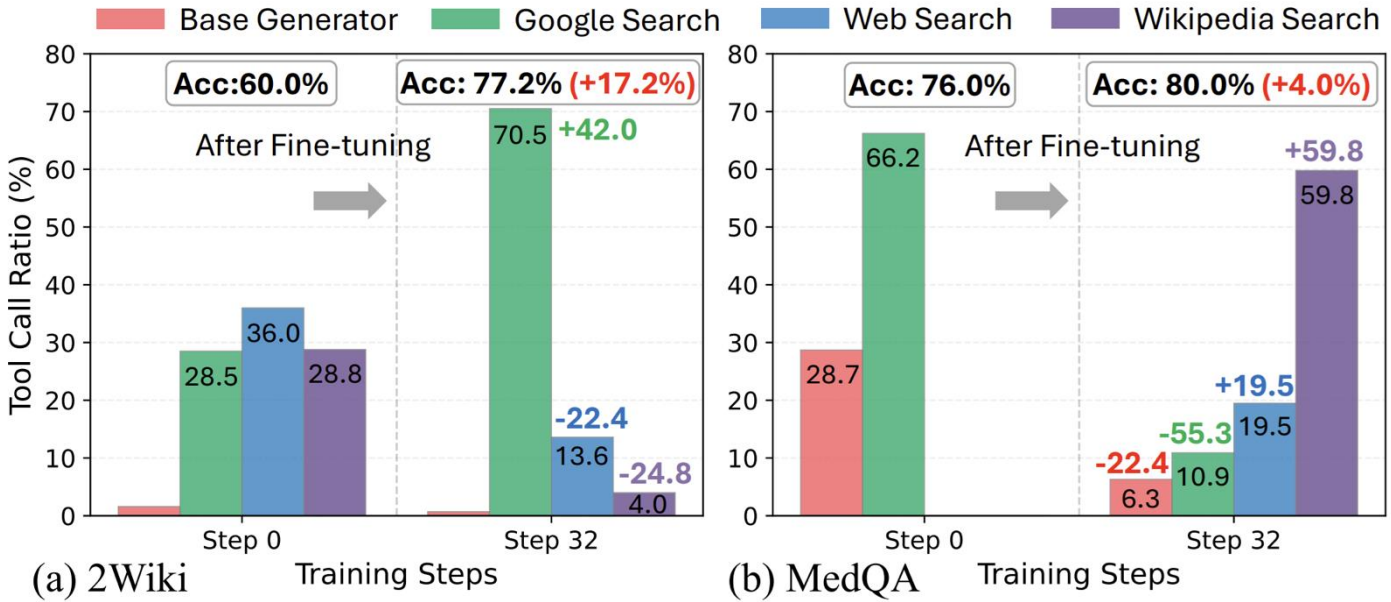
Highlight Results



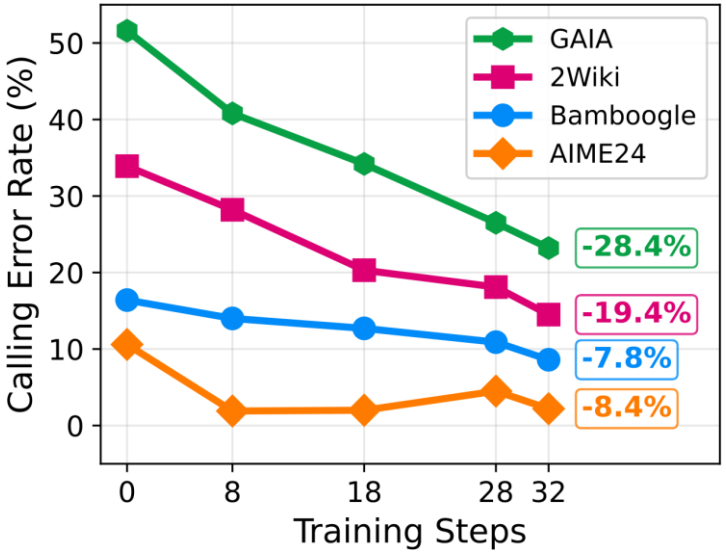
Highlighted Main Results

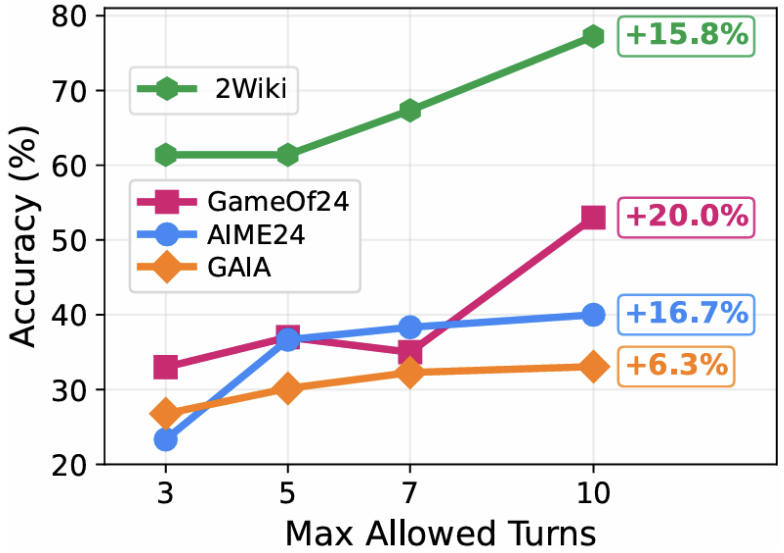


Autonomy new tool usage distribution

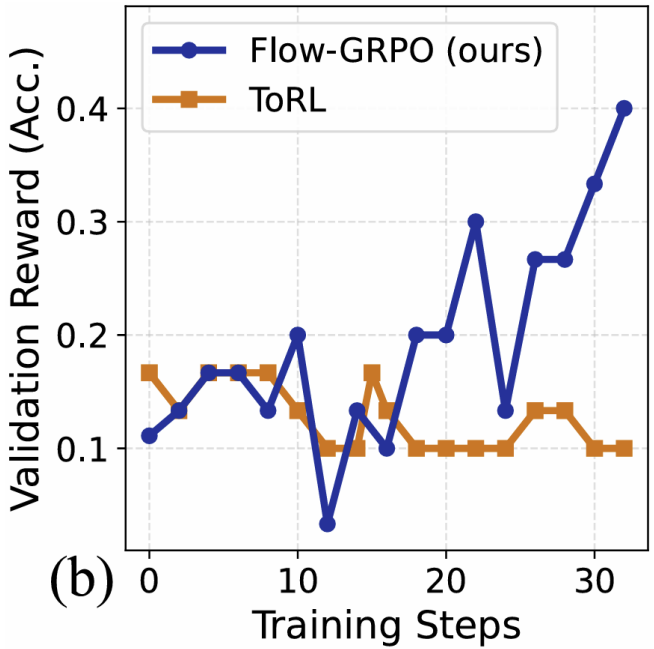


Scale with turns, refine with tuning

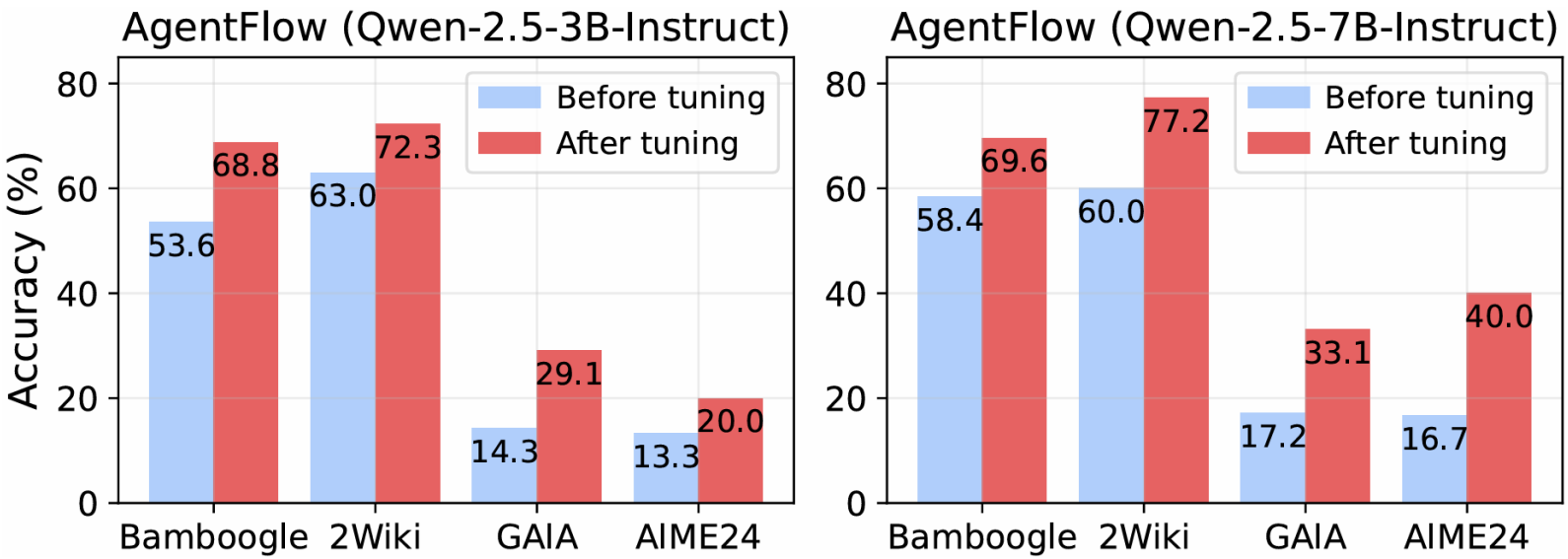




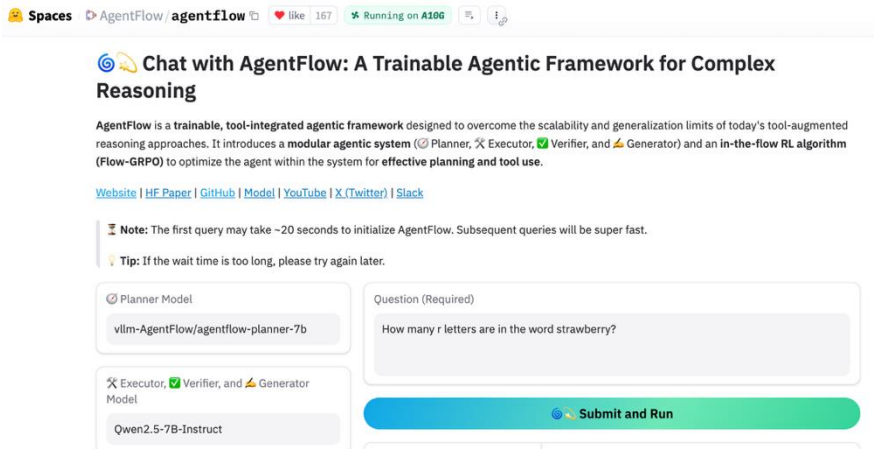
Controllable Planner Output & Training efficiency

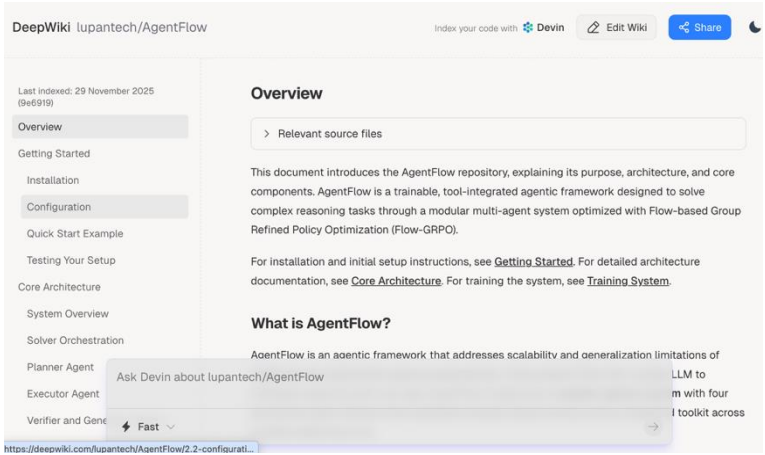


Scale with planner size

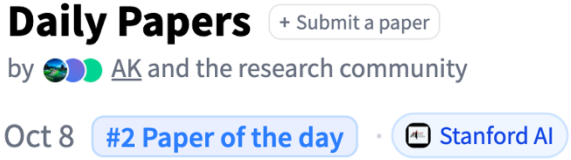


AgentFlow: Accessible Resources



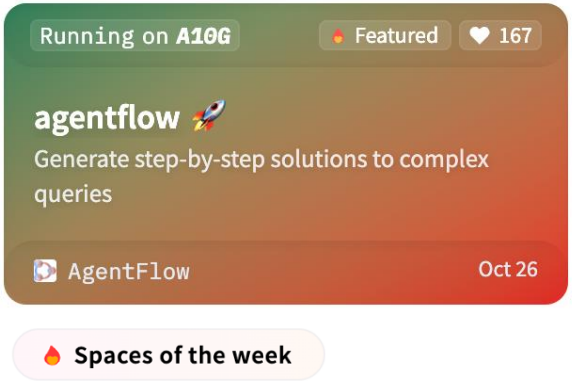


AgentFlow: Broad Impacts



#2 Paper of the day

YouTube (14K views)



HuggingFace Demo (47K visits)

AgentFlow: Takeaways

- **AgentFlow:** In-the-flow agentic system, coordinated by four specialized modules, diverse toolset, and evolving memory.
- Flow-GRPO: in-the-flow RL tuning, which cover

Website

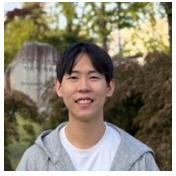
Slack

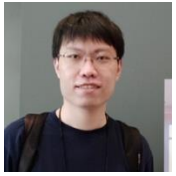
Website: https://agentflow.stanford.edu
Paper: https://arxiv.org/abs/2510.05592
Code: https://github.com/lupantech/AgentFlow
Model: https://huggingface.co/AgentFlow
Demo: https://huggingface.co/spaces/AgentFlow/agentflow
YouTube: https://www.youtube.com/watch?v=kIQbCQIH1SI

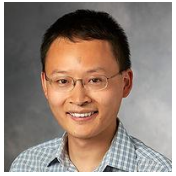
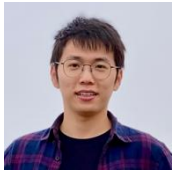
Thanks!

Q/A





*Zhuofeng Li** *Haoxiang Zhang** Seungju Han Sheng Liu Jianwen Xie Yu Zhang Yejin Choi *James Zou+* *Pan Lu*+*

AgentFlow Stanford TEXAS A&M UNIVERSITY UC San Diego Lambda