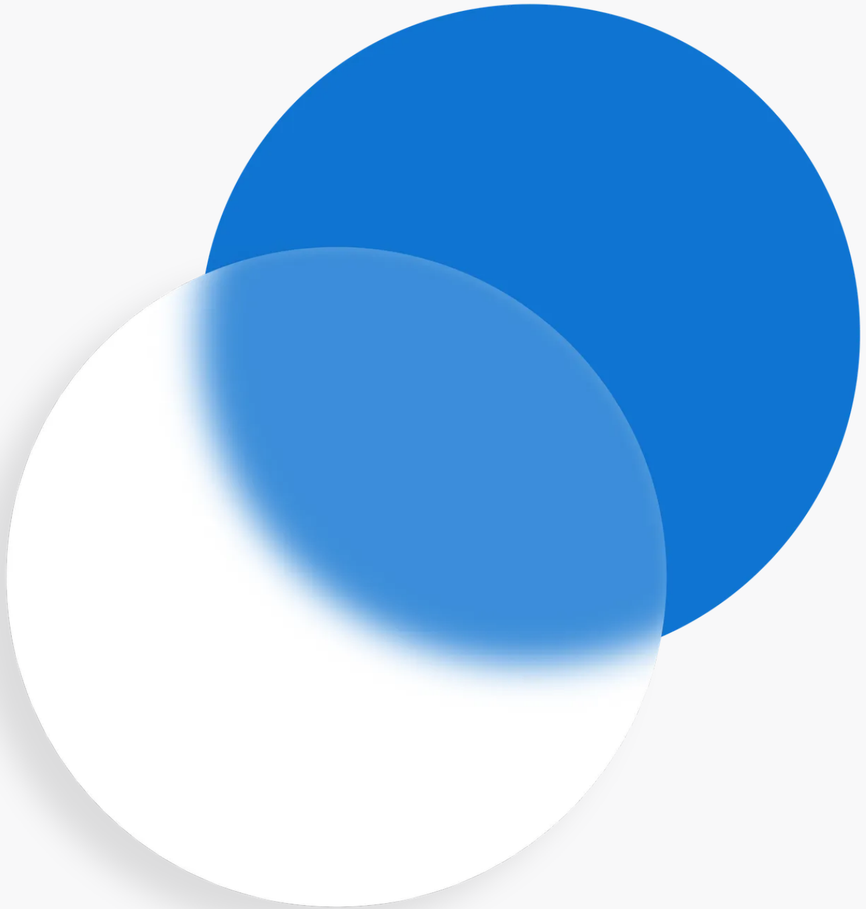
In-the-Flow Agentic System Optimization for Effective Planning and Tool Use



- Name Jiyun Kang

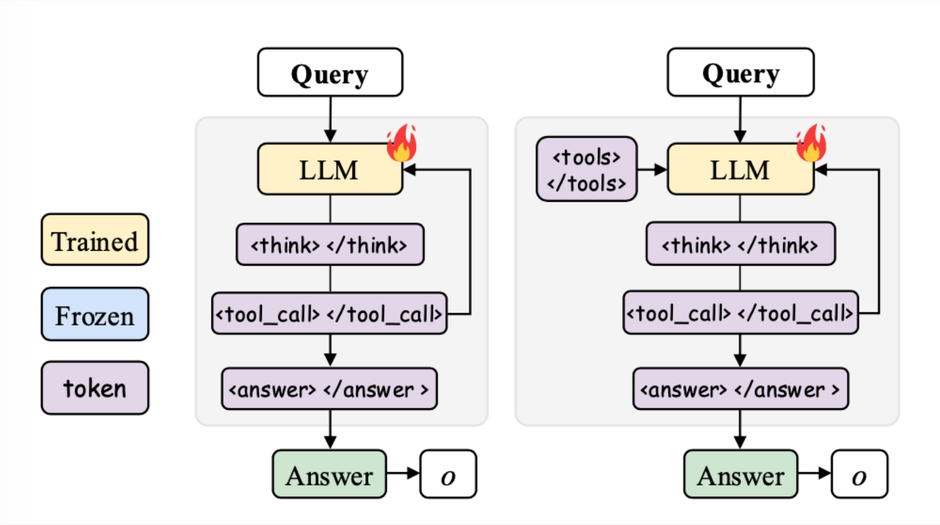
- Date 2025.11.26

1. Introduction

Backgrounds

: 기존 접근 방식의 한계

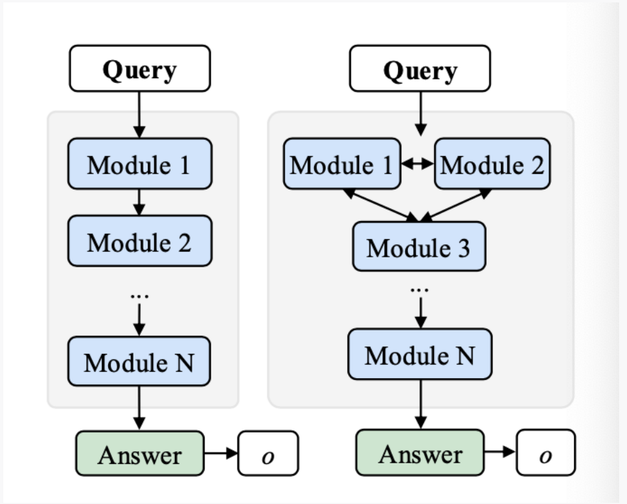
Tool-Integrated Reasoning, TIR



LLM 하나로 모든 추론·도구 호출을 처리하는 단일 구조

확장성 한계: 컨텍스트 길어지고 도구가 많아질수록 학습 불안정
일반화 취약: 새로운 작업·새로운 도구 등장 시 성능 급락

Agentic Systems



플래너–실행자–비평가 등 여러 모듈이 협업

Training-free : 대부분 프롬프트 휴리스틱·수작업 논리에 의존
오프라인 학습 한계 (SFT/Preference): 실제 도구 피드백 반영 불가

1. Introduction

Backgrounds

: The Main Problem of Tool-Integrated Agent

"Long-horizon tasks with sparse rewards"

Problem. 1

마지막에 주어지는 보상
(중간 단계 물음표...)

Problem. 2

어떤 turn이 도움이 되었는지
학습 불가능

Problem. 3

LLM이 실수하면 회복 불가
(compounding errors)

1. Introduction

Backgrounds

: The Main Problem of Tool-Integrated Agent

"Long-horizon tasks with sparse rewards"

Problem. 1

마지막에 주어진 보상
(중간 단계 물음표...)

Problem. 2

어떤 turn이 도움이 되었는지
학습 불가능

Problem. 3

LLM이 실수하면 회복 불가
(compounding errors)

Solution

AgentFlow

훈련 가능한(trainable) + 실시간(in-the-flow) + 장기 계획 가능한 에이전트 시스템

1. Introduction

Backgrounds

: Preview of Agentflow

Action Planner, P

모든 행동을 계획하는 플래너

학습 가능한 정책(πθ)으로, 다음 행동을 계획하고, 하위 목표를 설정하며, 도구를 선택하고, 메모리에서 관련 컨텍스트를 검색하는 역할

Tool Executor, E

도구를 실행하는 실행자

선택된 도구를 컨텍스트와 함께 호출하고 실행 관찰 결과(et)를 산출

Execution Verifier, V

검색한 결과가 유효한지 평가하는 검증자

실행 결과(et)가 유효한지, 누적된 메모리(Mt)가 쿼리를 해결하기에 충분한지 평가하여 이진 검증 신호(vt)를 생성

Solution Generator, G

솔루션 생성자

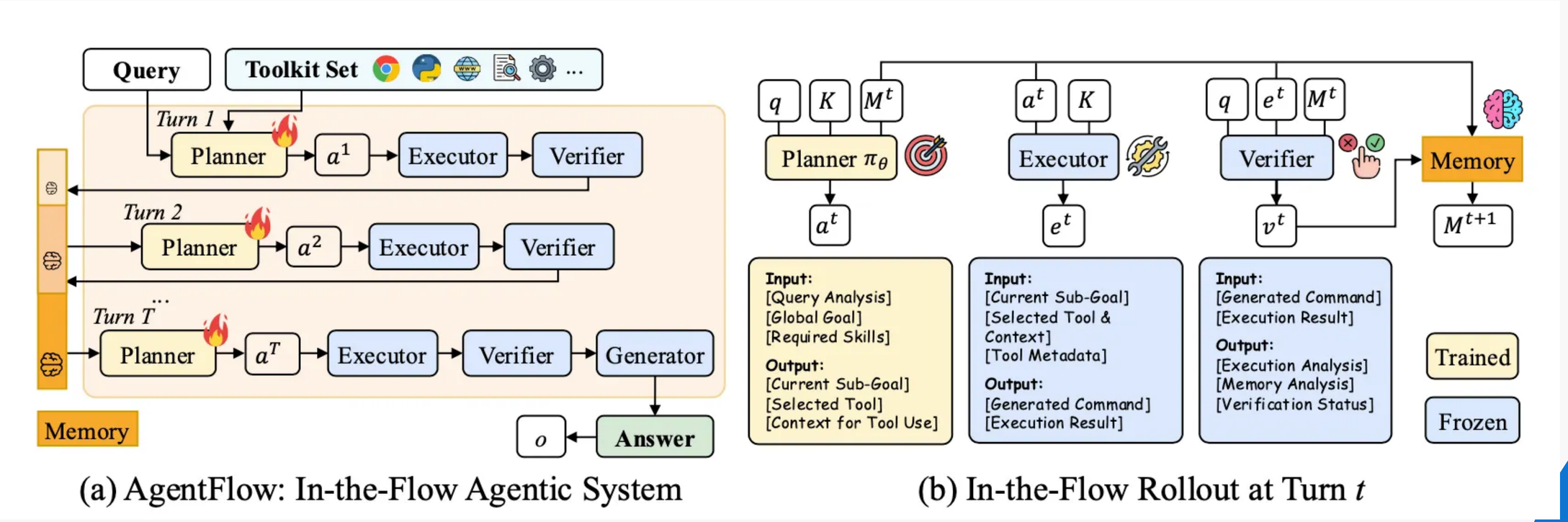
모든 턴이 종료되면 쿼리와 누적된 메모리를 기반으로 최종 솔루션(o)을 생성

with Flow-GRPO

Agentflow: An In-the-flow Agentic System

: Multi-turn Markov Decision Process(MDP)

"이전의 모든 history가 아닌 현재 Memory만 가지고 있으면, next action을 결정"

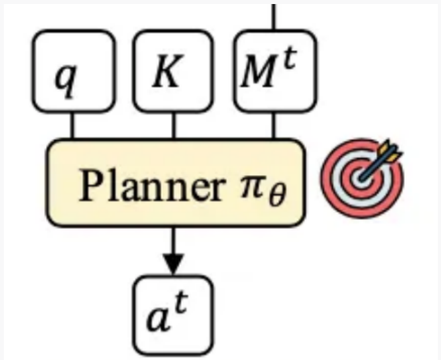


(a) AgentFlow: In-the-Flow Agentic System

(b) In-the-Flow Rollout at Turn t

Agentflow: An In-the-flow Agentic System

: Action Planner P



Input:
[Query Analysis]
[Global Goal]
[Required Skills]

Output:
[Current Sub-Goal]
[Selected Tool]
[Context for Tool Use]

Action Planner, P

Query | Tools | Memory

Instruction for Action Planner

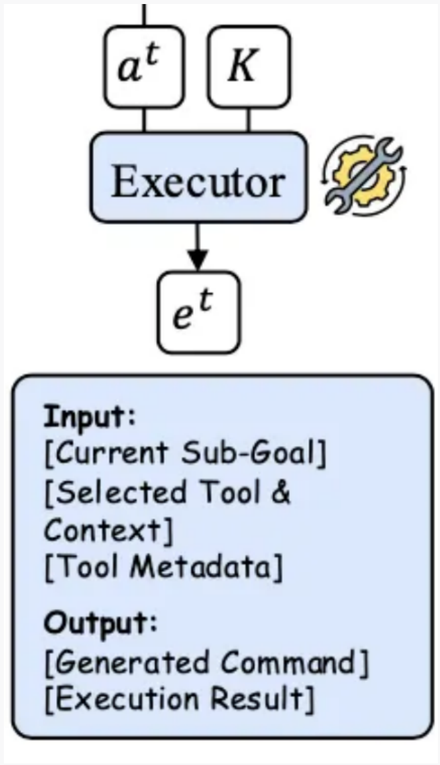
Task: Determine the optimal next step to address the query using available tools and previous context.

Context:

Query: {Question}
Available Tools: [Base Generator, Python Coder, Google Search, Wikipedia Search, Web Search]
Toolbox Metadata: [Tool Metadata1, Tool Metadata2, ...]
Previous Steps: {Actions from Memory}

Agentflow: An In-the-flow Agentic System

: Tool Executor, E



Tool Executor, E

Planner Output(Current Sub-Goal)

Selected Tool

Instruction for Tool Executor

Task: Generate a precise command to execute the selected tool.

Context:

Query: {Question}
Sub-Goal: {Sub Goal from Next Step Plan}
Tool Name: {Selected Tool from Next Step Plan}
Toolbox Metadata: {Selected Tool Metadata from Next Step Plan}
Relevant Data: {Context from Next Step Plan}

Output Format: Present your response in the following structured format. Do not include any extra text or explanations.

Example 1:

Generated Command:

```
execution = tool.execute(query="Summarize the following porblom:"Isaac has
100 toys, masa gets ...., how much are their together?")
```

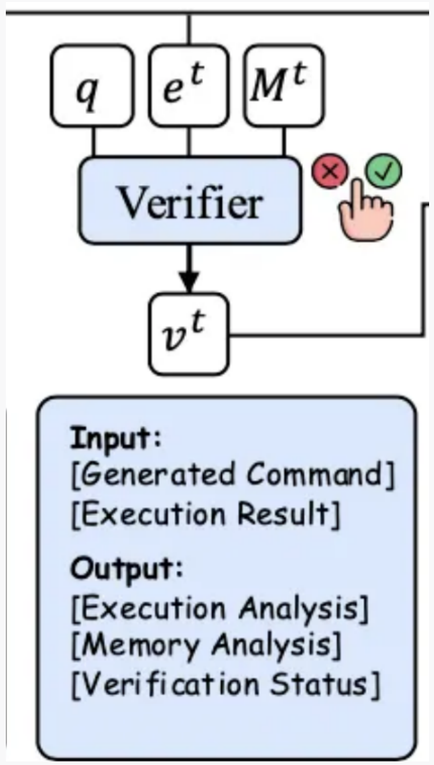
Example 2:

Generated Command:

```
execution = tool.execute(query=["Methanol", "function of hyperbola",
"Fermat's Last Theorem"])
```

Agentflow: An In-the-flow Agentic System

: Execution Verifier, V



Execution Verifier, V

Executor output | Memory | Query

Instruction for Execution Verifier

Task: Evaluate if the current memory is complete and accurate enough to answer the query, or if more tools are needed.

Context:

Query: {Question}
Available Tools: [Base Generator, Python Coder, Google Search, Wikipedia Search, Web Search]
Toolbox Metadata: [Tool Metadata1, Tool Metadata2, ...]
Memory (Tools Used & Results): {Actions from Memory}

- vt=0 (계속): 메모리가 새로운 증거를 통합하여 업데이트 $(M_{t+1} = f_{mem}(M^t, a^t, e^t, v^t))$
- vt=1 (종료): 최대 턴 예산에 도달하거나 검증자가 종료를 결정하면 프로세스가 종료

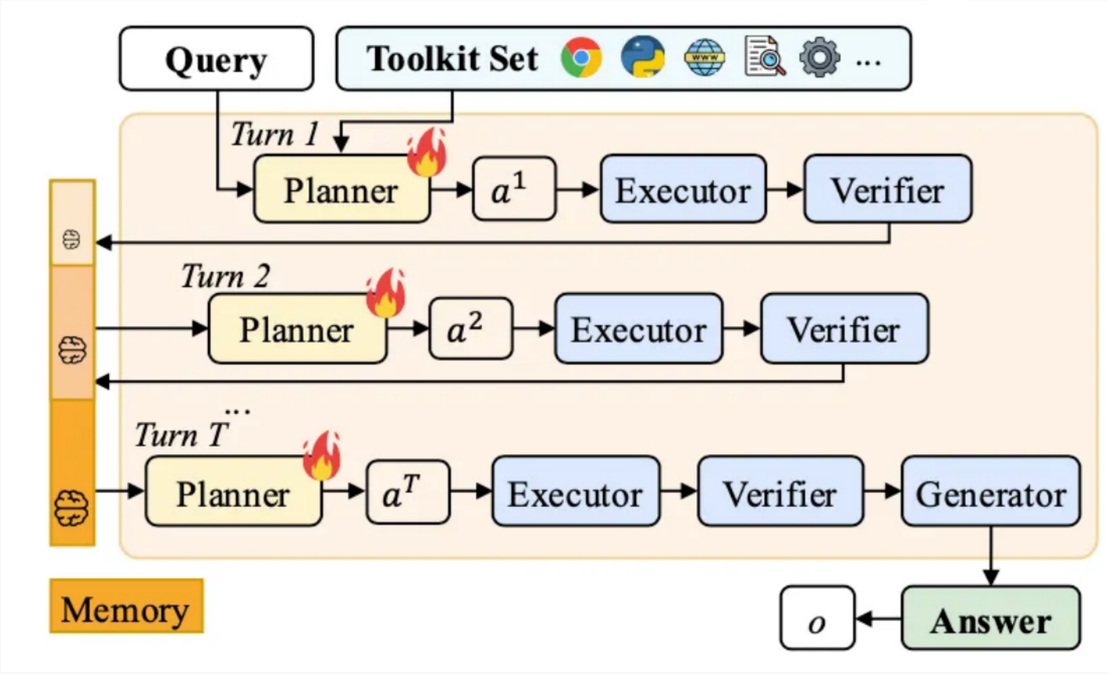
Agentflow: An In-the-flow Agentic System

: Solution Generator G

The joint generative process (output이 나오게될 확률)

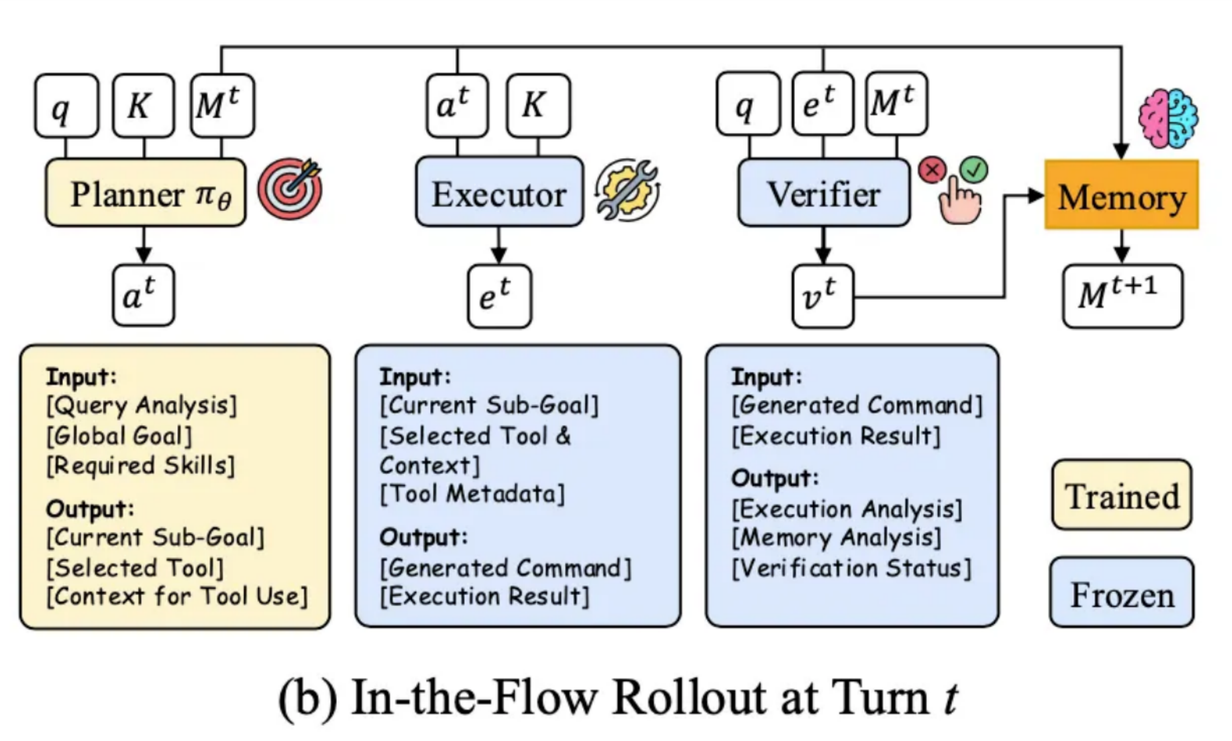
$$p_\theta\left(\{a^t, e^t, v^t\}_{t=1}^{T}, o \mid q\right) = \left[\prod_{t=1}^{T} \pi_\theta(a^t \mid q, K, M^t)\, \mathcal{E}(e^t \mid a^t, K)\, \mathcal{V}(v^t \mid q, e^t, M^t)\right] \mathcal{G}(o \mid q, M^T),$$

Solution Generator, G



In-the-flow Reinforcement Learning Optimization

: optimize “the planner” in the flow of execution

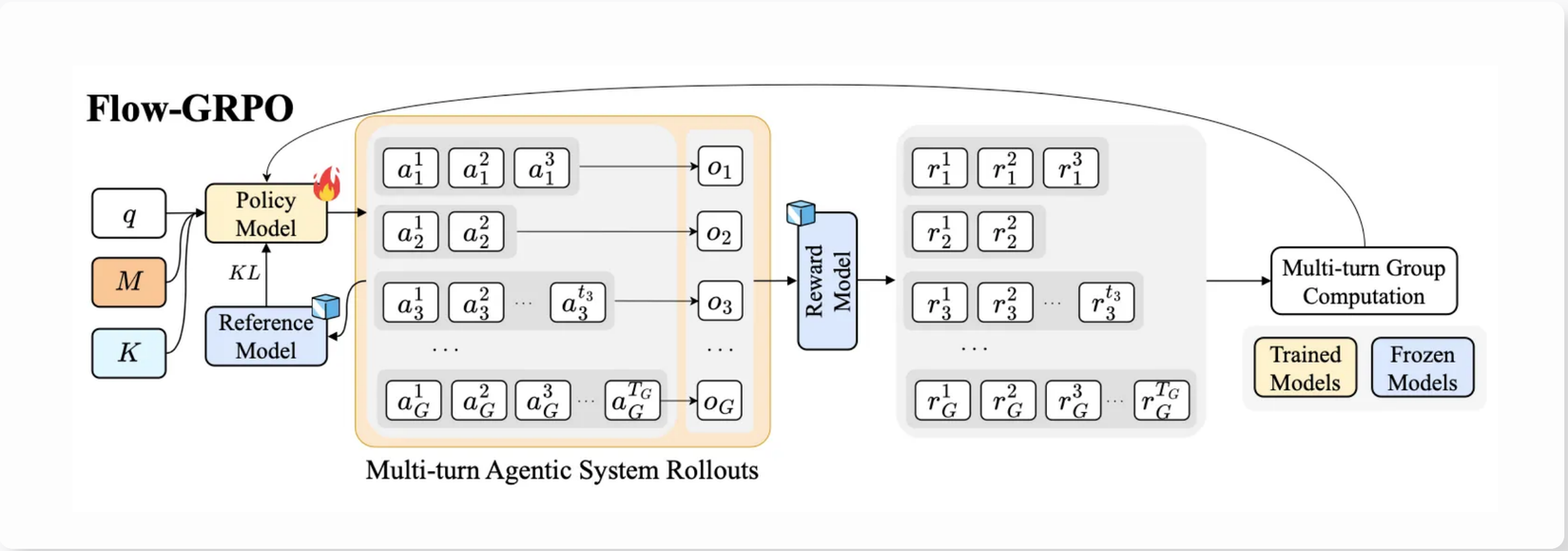


(b) In-the-Flow Rollout at Turn t

→ trajectory T를 collect해서, 최종 outcome signal인 vt로 다음 업데이트를 결정한 다음, Mt+1에 따라서 현재 정책(plan)을 다시 업데이트

In-the-flow Reinforcement Learning Optimization

: Flow-GRPO



Policy Optimization Objective

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\big[R(\tau)\big], \qquad \theta^\star = \arg\max_\theta \mathcal{J}(\theta),$$

Final-outcome reward(Sparse Reward)

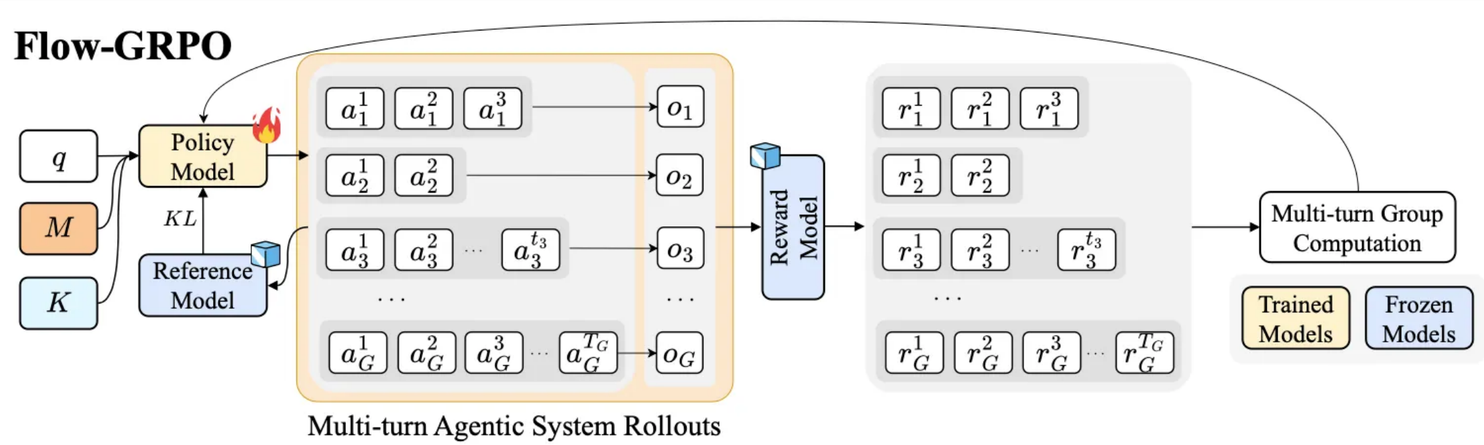
$$r = R(a^t) \quad \bar{R}(o, q, y^*), \quad \forall t = 1, \ldots, T$$

: 하나의 Rollut에 모든 action은 동일한 전역 보상(Global reward signal) 1 or 0

최종 보상을 모든 턴에 할당함으로써, 멀티 턴 강화 학습 문제를 다루기 쉬운
일련의 독립적인 싱글 턴 정책 업데이트로 효과적으로 변환
(“단일-step RL 문제처럼” 업데이트)

In-the-flow Reinforcement Learning Optimization

: Group-normalized advantages (상대적인 가치 부여)[1,1,0,0] → [+1,+1, -1, -1]



기존 GRPO : turn-level advantage

Value function 사용 : 앞으로의 기대 보상을 계산

$$A_t = Q(s_t, a_t) - V(s_t)$$

$$A'_t = \frac{A_t - mean(A)}{std(A)}$$

LLM

Multi-turn
Tool search

Value Function ✕

Flow-GRPO: rollout-level advantage

같은 문제(q)에 대해 샘플링한 G개의 rollout reward를 그룹으로 모아서 그 분포에서 정규화(평균 0, 표준편차 1)

$$A_i^t = \frac{\bar{R}(o_i, q, y^*) - \text{mean}\left(\{\bar{R}(o_k, q, y^*)\}_{k=1}^G\right)}{\text{std}\left(\{\bar{R}(o_k, q, y^*)\}_{k=1}^G\right)}.$$

In-the-flow Reinforcement Learning Optimization

: Objective function

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

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

rollout 평균: 1/G

turn(스텝) 간의 평균: $1/T_i$

token 간 평균: $1/|a^t_i|$

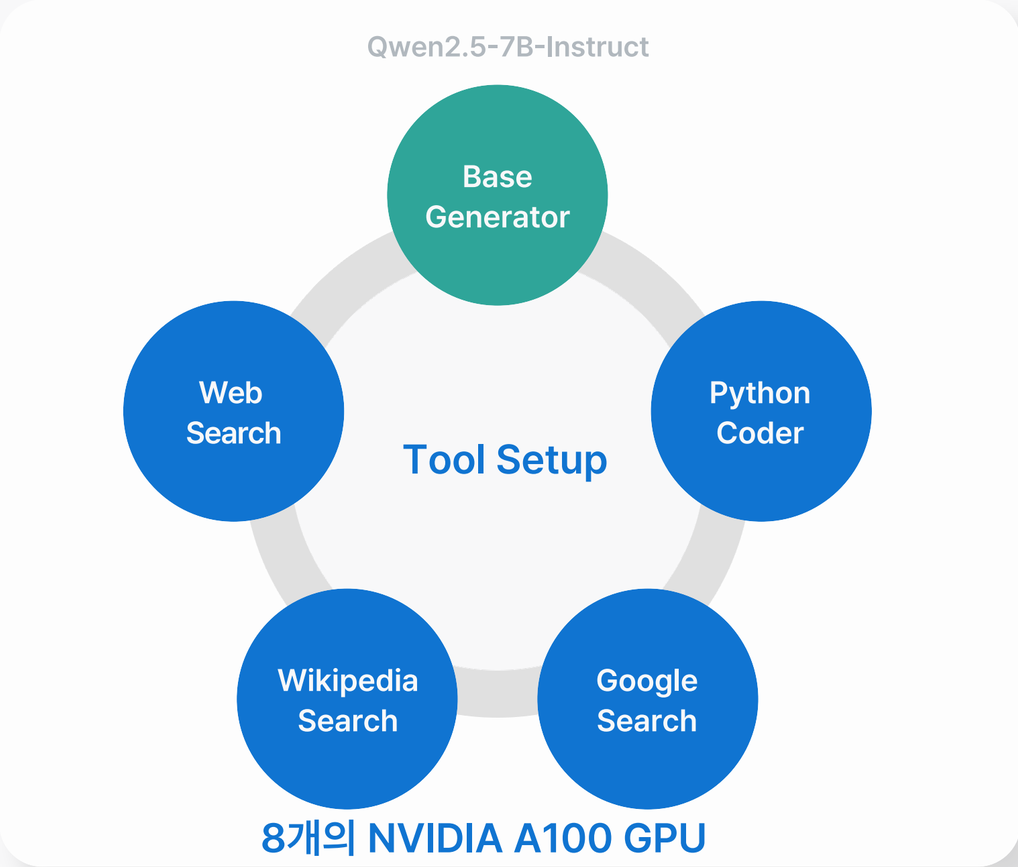
Token Level Importance Ratio

$$\rho_{i,j}^t = \frac{\pi_\theta\left(a_{i,j}^t \mid s_i^t, a_{i,1:j-1}^t\right)}{\pi_{\theta_{\text{old}}}\left(a_{i,j}^t \mid s_i^t, a_{i,1:j-1}^t\right)}$$

ex. ["search", "weather", "today"]
→ 각 토큰마다 p의 확률을 구하고, ρ1*A+ρ2*A+ρ3*A 를 계산

4. Experiments

Experimental Setup



Training Setup

Model Training Configuration

- Learning rate: 1×10^{-6}
- Action Planner sampling temperature: 0.5 (exploration–exploitation 균형)
- KL divergence penalty coefficient: $\beta = 0.001$
- Planner 최대 출력 길이: 2048 tokens
- Batch size: 32

Rollout & Turn Settings

- 샘플당 rollout 개수: 8
- 각 rollout의 최대 turn 수: 3 turns

Reward & Evaluation

- Final-outcome reward(정답 여부 0/1)는 GPT-4o 기반 LLM-as-judge가 평가

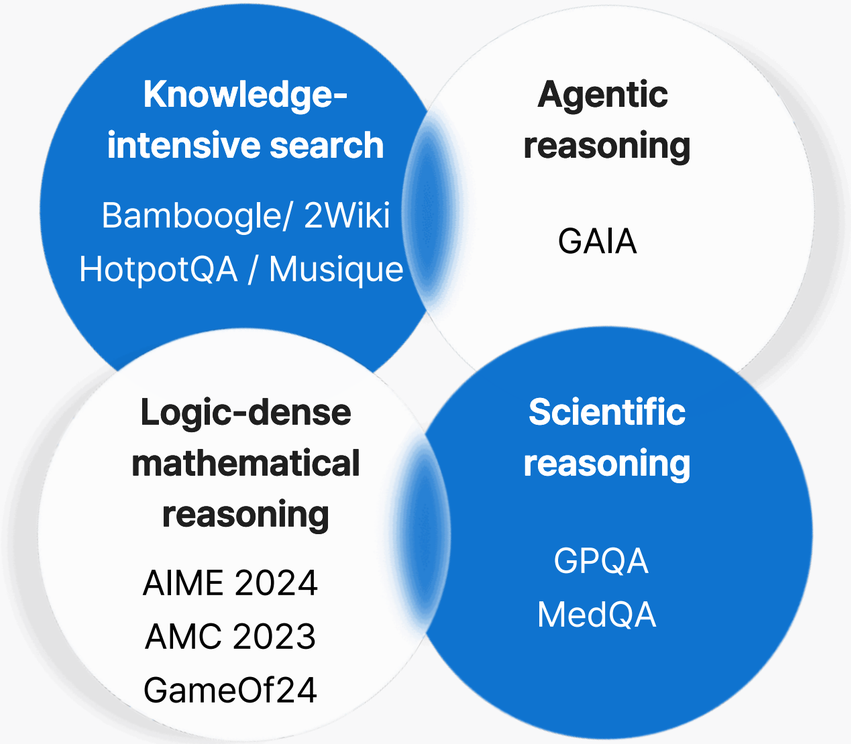
Tool Execution Environment

- 모든 tool call은 동기식(synchronous) 처리
- Tool call timeout: 500 seconds
- Tool 내부 LLM 엔진 temperature: 0.0 (determiniic)

4. Experiments

Experimental Setup

Evaluation Setup



Rollout & Planner Settings

- Evaluation 시 rollout 최대 turn 수: T = 10
  → 더 깊은 multi-step reasoning 수행 가능
- Planner sampling temperature: 0.7
  → 다양한 reasoning 경로 탐색 촉진

Evaluation Protocol: GPT-4o 기반 LLM-as-Judge

Task: Determine if the Model Response is equivalent to the Ground Truth.

Instructions:

1. **Extract:** Isolate the final answer from the Model Response, ignoring all reasoning steps. Look specifically for content within $\boxed{\ldots}$ or the concluding statement.
2. **Normalize & Compare:** Assess equivalence after normalization:
3. **Mathematical Answers:** Must be mathematically identical (e.g., $\frac{1}{2}$ is equivalent to 0.5).
4. **Numerical/Textual Answers:** Ignore formatting (commas, spaces), case sensitivity, and extraneous units/currency (e.g., “1,000” == “1000”, “Paris” == “PARIS”).
5. **Multiple Choice Questions (MCQ):** The answer must match either the correct option’s content (e.g., “Paris”) or its identifier (e.g., “A” or “1st”).
6. **Verdict:** Return “True” only if the normalized answers are semantically or mathematically equivalent.

Inputs:

Question: {Question}
Model Response: {Final Response from Solution Generator}
Ground Truth: {GT}

Output Format: Present your response in the following structured format. Do not include any extra text or explanations.
<analysis>: Brief analysis of the comparison.
<true_false>: “True” or “False”.

4. Experiments

Main Results: Accuracy

		Search Intensive						Agentic	
Model	**Size**	**Bamboogle**	**2Wiki**	**HotpotQA**	**Musique**	**Avg.**	**Δ**	**GAIA**	**Δ**
Qwen-2.5-7B-Instruct	7B-Inst	12.0	23.0	21.0	6.0	15.5	↑ 41.8	3.2	↑ 29.9
Qwen-2.5-14B-Instruct	14B-Inst	21.6	26.7	20.0	8.0	19.1	↑ 38.2	5.5	↑ 27.6
Qwen-2.5-32B-Instruct	32B-Inst	24.0	26.7	27.0	6.0	20.9	↑ 36.4	9.5	↑ 23.6
Llama-3.3-70B-Instruct	70B-Inst	18.4	22.7	52.0	16.0	27.3	↑ 30.0	3.2	↑ 29.9
GPT-4o-mini (Hurst et al., 2024)	∼8B	40.8	35.6	41.0	15.0	33.1	↑ 24.2	7.1	↑ 26.0
GPT-4o (Hurst et al., 2024)	∼200B	68.8	49.5	54.0	24.0	49.1	↑ 8.2	17.3	↑ 15.8
Supervised Fine-Tuning (SFT)	7B-Inst	12.0	25.9	22.0	6.6	16.6	↑ 40.7	3.2	↑ 29.9
Iter-RetGen (Shao et al., 2023)	7B-Inst	36.8	33.6	37.4	17.8	31.4	↑ 25.9	3.9	↑ 29.2
Search-R1 (Jin et al., 2025)	7B-Inst	43.2	38.2	37.0	14.6	33.3	↑ 24.0	19.1	↑ 14.0
ZeroSearch (Sun et al., 2025)	7B-Base	27.8	35.2	34.6	18.0	28.9	↑ 28.4	16.5	↑ 16.6
ReSearch (Chen et al., 2025)	7B-Base	42.4	47.6	43.5	22.3	39.0	↑ 18.3	17.3	↑ 15.8
StepSearch (Wang et al., 2025d)	7B-Base	40.0	36.6	38.6	22.6	34.5	↑ 22.8	–	–
VerlTool (Jiang et al., 2025)	7B-Base	46.4	45.3	44.8	19.3	39.0	↑ 18.3	11.2	↑ 21.9
AutoGen (Wu et al., 2024)	7B-Inst	59.6	44.0	50.0	15.9	42.4	↑ 14.9	6.3	↑ 26.8
AGENTFLOW	7B-Inst	58.4	60.0	51.3	19.2	47.2	↑ 12.1	17.2	↑ 15.9
AGENTFLOW (w/ Flow-GRPO)	7B-Inst	**69.6**	**77.2**	**57.0**	**25.3**	**57.3**	–	**33.1**	–

Proprietary LLMs

Search-Integrated Reasoning LLMs

Training-free Agentic System

4. Experiments

Main Results: Accuracy

Reasoning LLMs

Code-Integrated Reasoning LLMs

		Math Reasoning					Scientific Reasoning			
Model	**Size**	**AIME24**	**AMC23**	**GameOf24**	**Avg.**	**Δ**	**GPQA**	**MedQA**	**Avg.**	**Δ**
Qwen-2.5-7B-Instruct	7B-Inst	6.7	47.5	33.0	29.1	↑ 22.5	34.0	66.0	50.0	↑ 13.5
Qwen-2.5-14B-Instruct	14B-Inst	6.7	60.0	25.0	30.6	↑ 21.0	31.0	75.0	53.0	↑ 10.5
Llama-3.3-70B-Instruct	70B-Inst	6.7	47.5	31.0	28.4	↑ 23.1	35.0	67.0	51.0	↑ 12.5
Llama-3.1-405B-Instruct	405B-Inst	26.7	47.5	23.0	32.4	↑ 19.1	30.0	62.0	46.0	↑ 17.5
GPT-4o-mini (Hurst et al., 2024)	∼8B	13.3	57.5	16.0	28.9	↑ 22.6	27.0	66.0	46.5	↑ 17.0
GPT-4o (Hurst et al., 2024)	∼200B	13.3	60.0	32.0	35.1	↑ 16.4	31.0	60.0	45.5	↑ 18.0
Supervised Fine-Tuning (SFT)	7B-Inst	6.7	47.5	33.0	29.1	↑ 22.5	34.0	66.0	50.0	↑ 13.5
SimpleRL-reason (Zeng et al., 2025b)	7B-Base	16.7	60.0	33.0	36.6	↑ 15.0	45.0	65.0	50.0	↑ 13.5
Open-Reasoner-Zero (Hu et al., 2025a)	7B-Base	16.7	54.9	32.0	34.5	↑ 17.0	34.0	54.0	44.0	↑ 19.5
General-Reasoner (Ma et al., 2025)	7B-Base	13.3	55.0	33.0	33.8	↑ 17.7	35.5	61.0	48.3	↑ 15.2
Luffy (Yan et al., 2025)	7B-Inst	30.7	44.8	33.0	36.2	↑ 15.3	34.0	77.0	55.5	↑ 8.0
TIR (Yang et al., 2024b)	7B-Inst	10.0	50.0	33.0	31.0	↑ 20.5	42.0	76.8	59.4	↑ 4.1
ToRL (Li et al., 2025b)	7B-Inst	20.0	60.0	31.0	37.0	↑ 14.5	35.0	76.5	55.8	↑ 7.7
AutoGen (Wu et al., 2024)	7B-Inst	13.3	57.5	24.0	31.6	↑ 19.9	42.0	72.0	57.0	↑ 6.5
AGENTFLOW	7B-Inst	16.7	47.4	31.0	31.7	↑ 19.8	37.0	76.0	56.5	↑ 7.0
AGENTFLOW (w/ Flow-GRPO)	7B-Inst	**40.0**	**61.5**	**53.0**	**51.5**	–	**47.0**	**80.0**	**63.5**	–

4. Experiments

In-depth Anaysis of Optimized Planning

1. 도구 사용률 변화 양상

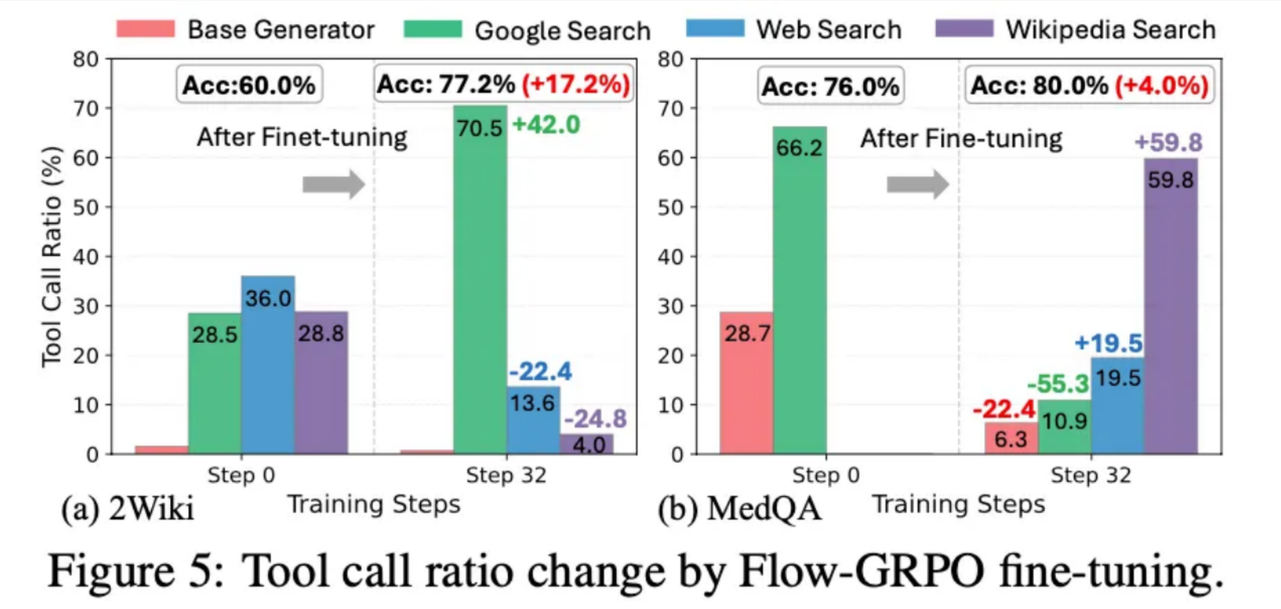


Figure 5: Tool call ratio change by Flow-GRPO fine-tuning.

광범위한 지식
(Google Search)

전문 지식
(Wikipedia)

2. 도구 호출 오류 감소

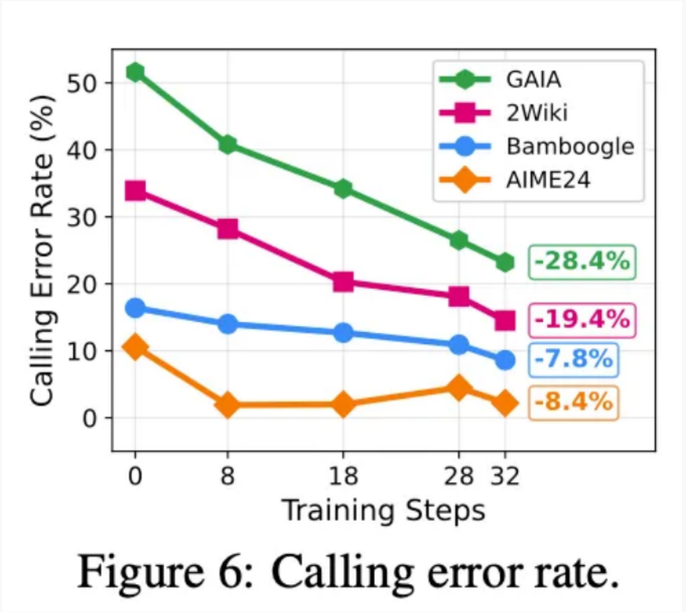
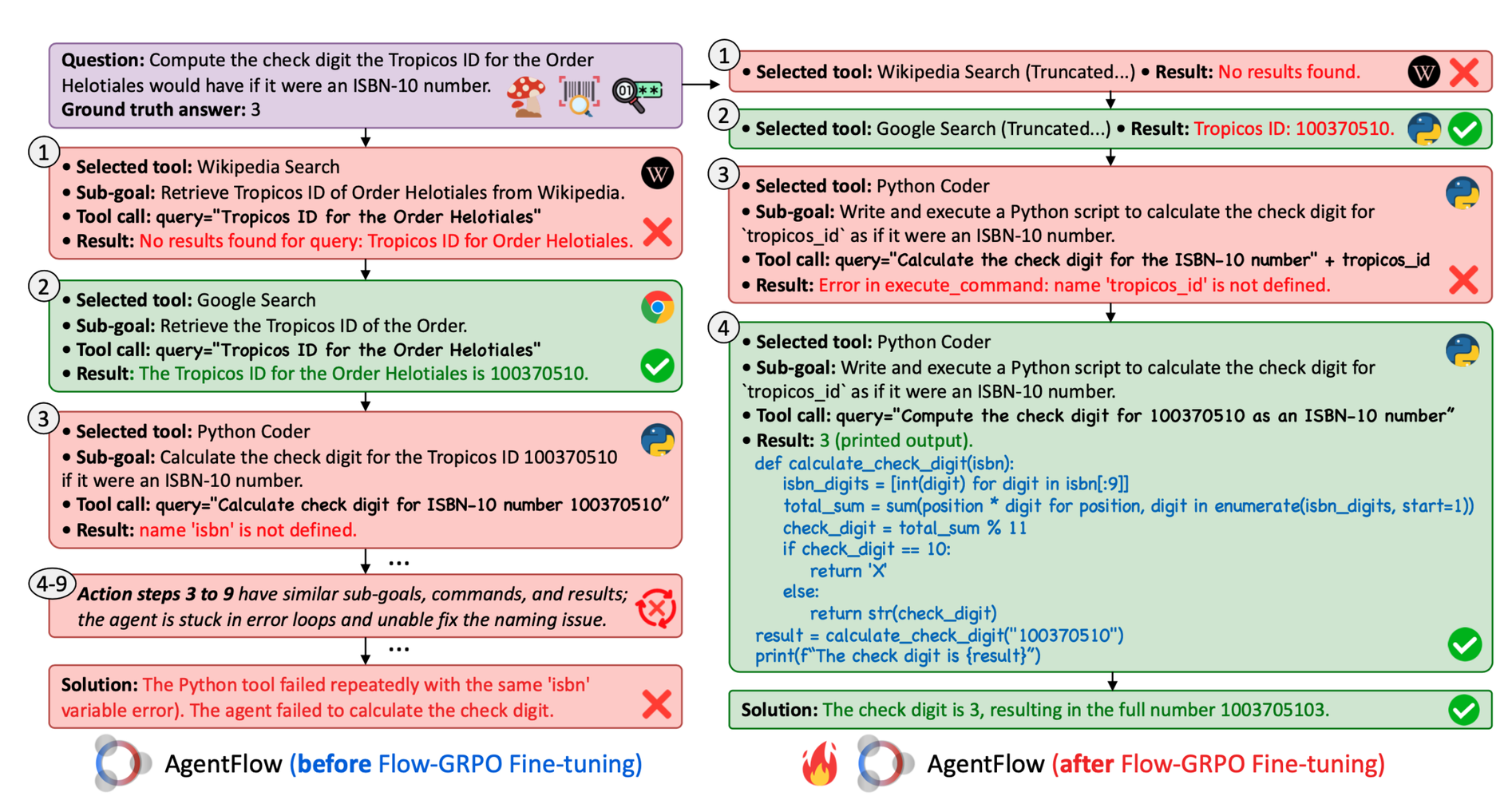


Figure 6: Calling error rate.

4. Experiments

In-depth Anaysis of Optimized Planning

3. 새로운 솔루션 경로를 다시 찾는 능력



4. Experiments

Training Strategies On The Planner

Planner Model	Training	Bamboogle	2Wiki	GAIA	AIME24	AMC23	GameOf24	Avg.
Qwen-2.5-7B	Frozen	58.4	60.0	17.2	16.7	47.4	31.0	38.5
GPT-4o	Frozen	65.0 ↑ 6.6	70.0 ↑ 10.0	23.6 ↑ 6.4	16.7 ↑ 0.0	48.7 ↑ 1.3	42.0 ↑ 11.0	44.3 ↑ 5.8
Qwen-2.5-7B	SFT	30.4 ↓ 28.0	32.7 ↓ 27.3	6.3 ↓ 10.9	3.3 ↓ 13.4	37.5 ↓ 9.9	7.0 ↓ 24.0	19.5 ↓ 19.0
Qwen-2.5-7B	Flow-GRPO	69.6 ↑ 11.2	77.2 ↑ 17.2	33.1 ↑ 15.9	40.0 ↑ 23.3	61.5 ↑ 14.1	53.0 ↑ 22.0	55.7 ↑ 17.2

Table 3: Performance comparison of AGENTFLOW across different training methods.

SFT 의 성능저하

SFT(지도학습)는 토큰 수준 모방(token-level imitation) 목적

- 실제로 이 도구를 왜 써야하는지는 학습 되지 않음 !
- 도구의 error가 발생하면 복구할 수도 없음 !

4. Experiments

Training Efficiency Analysis

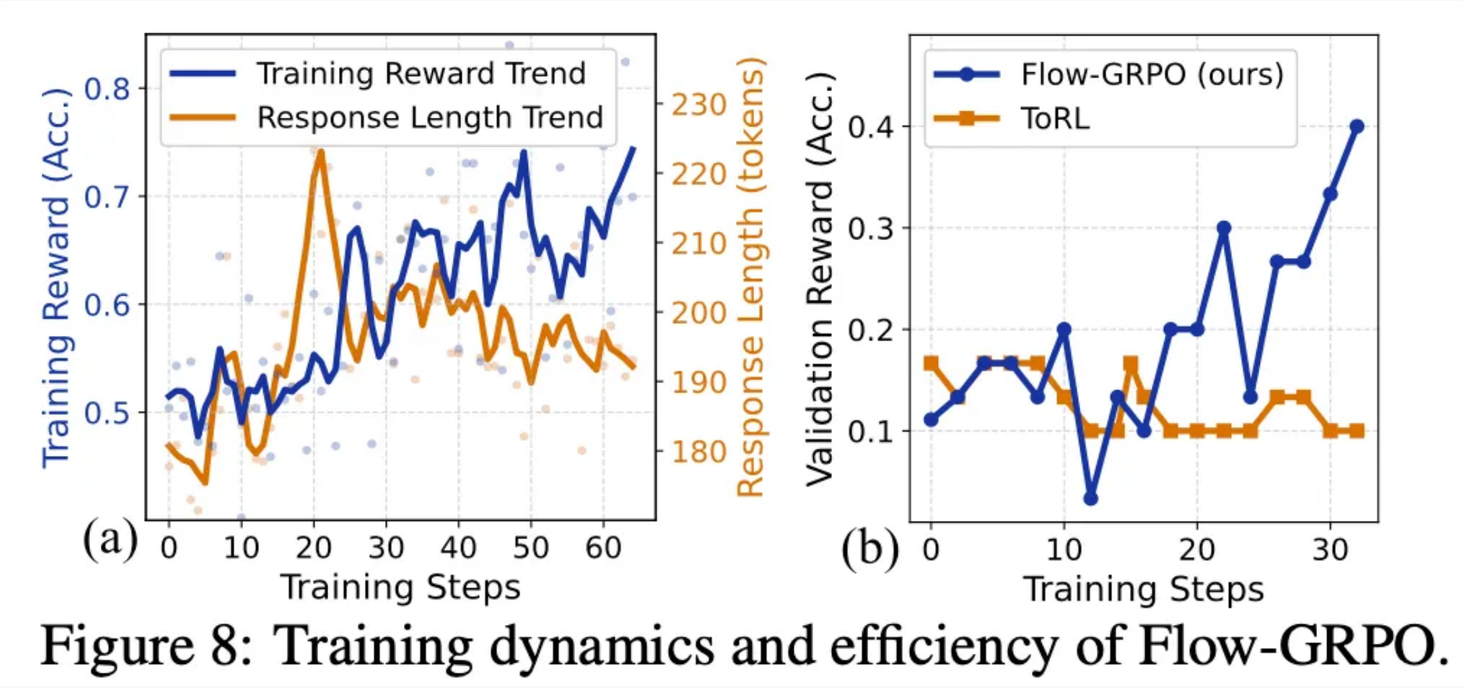


Figure 8: Training dynamics and efficiency of Flow-GRPO.

보상 증가 및 응답 간결화

Flow-GRPO를 통한 훈련 보상은 꾸준히 증가 동시에, 응답 길이(Response Length)는 초기 탐색적 증가 이후 점진적으로 단축되고 안정화

ToRL 대비 효율성

단일 정책 기반의 도구 통합 추론 베이스라인(ToRL)과 비교했을 때, 검증 정확도에서 꾸준한 성능 향상을 보인 반면, ToRL은 빠르게 정체되거나 하락하는 경향

4. Experiments

Scaling Trends In Agentflow

1. Backbone 모델 크기에 따른 성능

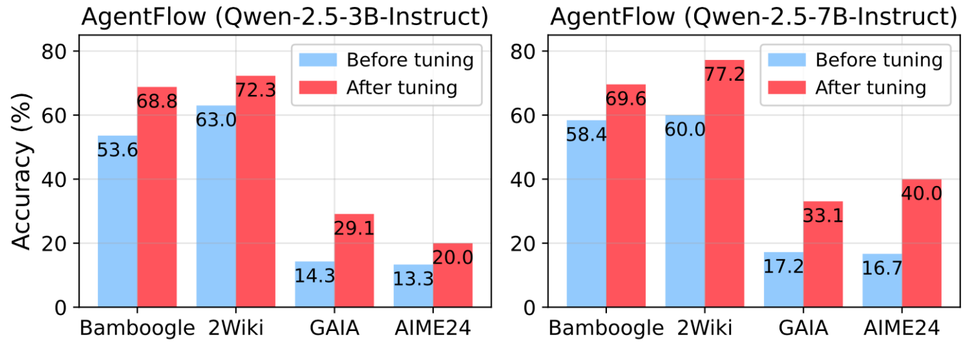


Figure 9: Flow-GRPO fine-tuning offers consistent gains on AGENTFLOW as the backbone model size scales from 3B to 7B.

2. Turn 크기 스케일링 (3 to 10)

Turns (T_{max})	3	5	7	10
2Wiki	2.22	3.18	3.81	4.44
GameOf24	1.63	2.12	2.36	2.67
AIME24	1.63	1.63	1.86	1.90
GAIA	2.43	3.46	4.28	5.42

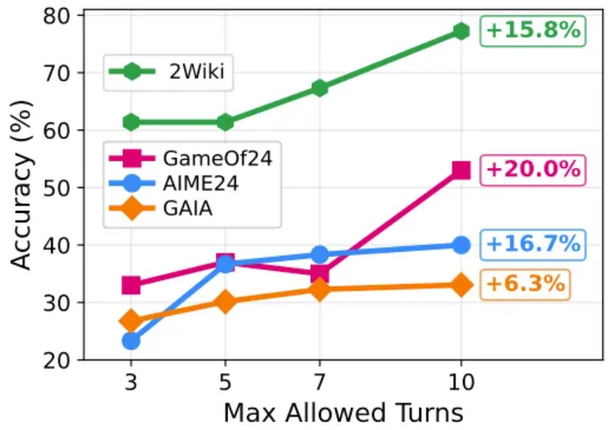


Figure 10: Average turns and accuracy with increased T_{max}.