[2026 summer seminar]

Efficient Agentic Systems through Latent Collaboration and Tool Orchestration

Pusan National University
Computer Vision & Signal Processing Lab

Presented By
윤상연

CVSP Lab
Computer Vision & Signal Processing

Index

1) Zou, Jiaru, et al. "Latent collaboration in multi-agent systems." In Forty-third International Conference on Machine Learning (2026).
2) Su, Hongjin, et al. "Toolorchestra: Elevating intelligence via efficient model and tool orchestration." In Forty-third International Conference on Machine Learning (2026).

- Background
- 논문 소개

  Latent Collaboration in Multi-Agent Systems [ICML 2026 Spotlight] [1)]

  ToolOrchestra: Elevating Intelligence via Efficient Model and Tool Orchestration [ICML 2026] [2)]

- Conclusion

CVSP Lab
Computer Vision & Signal Processing

1) Zou, Jiaru, et al. "Latent collaboration in multi-agent systems." In Forty-third International Conference on Machine Learning (2026).

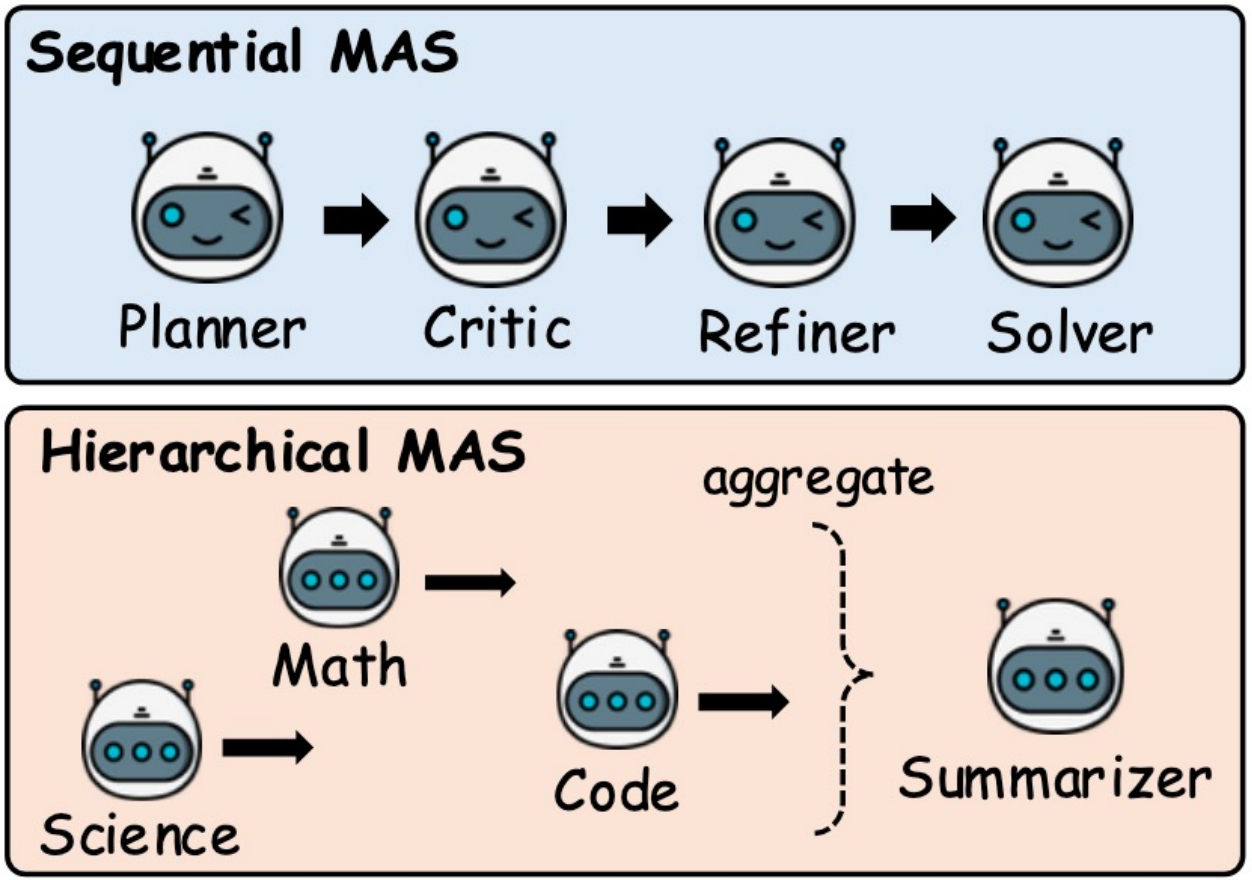
Latent Collaboration in Multi-Agent Systems

[ICML 2026 Spotlight]

Background

- Multi-Agent Systems(MAS)

  여러 에이전트가 역할을 분담하고 서로 상호작용하여 복잡한 문제를 해결



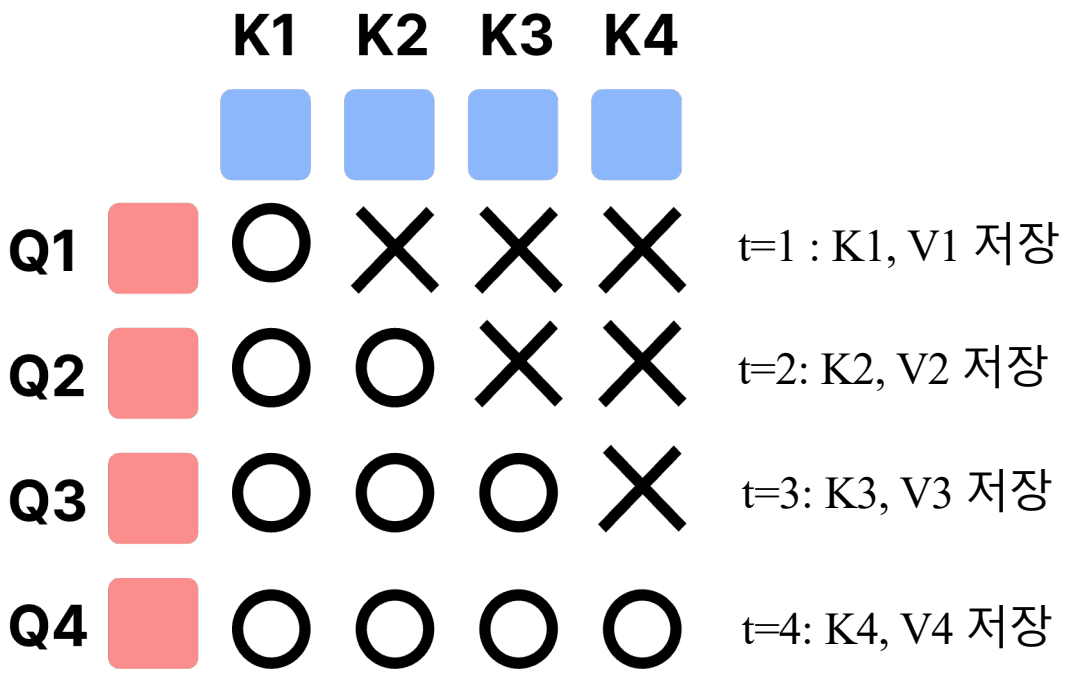
CVSP Lab
Computer Vision & Signal Processing

Background

- KV cache

  LLM 내 attention 연산 과정에서 생기는 KV를 저장, 이후 연산에 재활용

  - 연산 복잡도 $O(n^2)$ → $O(n)$



<Causal Attention>

CVSP Lab
Computer Vision & Signal Processing

Introduction

3) Hao, Shibo, et al. "Training large language models to reason in a continuous latent space." *arXiv preprint arXiv:2412.06769* (2024).
4) Zheng, Yujia, et al. "Thought communication in multiagent collaboration." Advances in Neural Information Processing Systems 38 (2025): 123389-123418.
5) Fu, Tianyu, et al. "Cache-to-cache: Direct semantic communication between large language models." International Conference on Learning Representations. Vol. 2026.

- Motivation

  기존 연구: 1) Latent CoT Reasoning[3,4], 2) KV cache를 이용한 모델 간 정보 전달[5]

  - Training Based method

  Latent Space 내에서 작동하는 MAS 설계

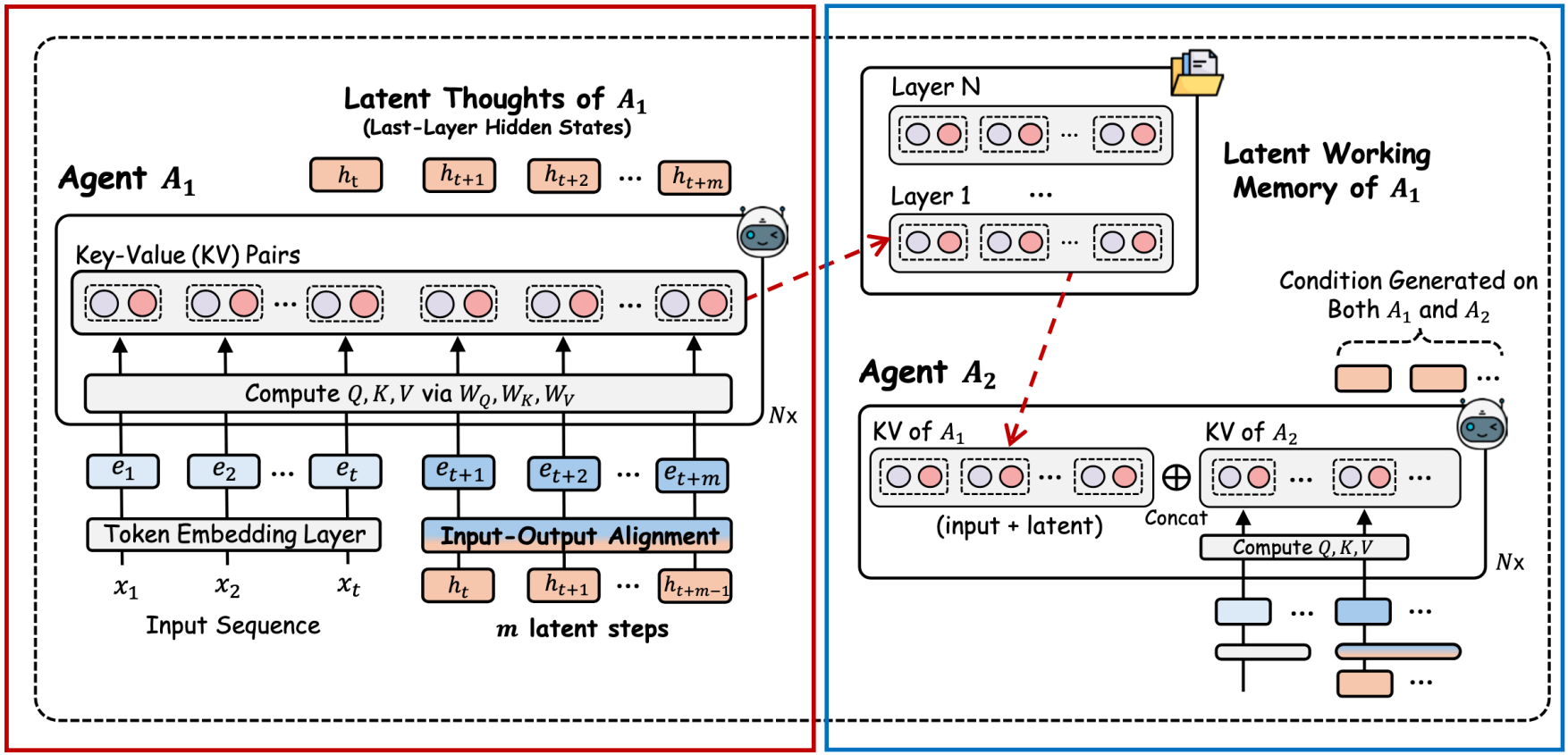
  - MAS 분야에서 1)과 2)를 통합한 연구는 진행되지 않음
  - 동일한 Latent space에서 reasoning 및 communication을 수행하는 Training-free MAS Framework 제안

    ⁂ Text-based 대비 더 좋은 표현력, 더 빠른 추론 속도 달성

CVSP Lab
Computer Vision & Signal Processing

Method

- Overview

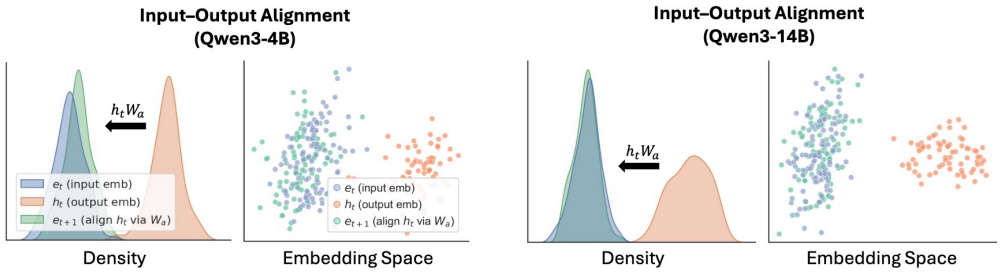


1. Latent Reasoning

2. Latent Communication

CVSP Lab
Computer Vision & Signal Processing

Method



- Latent Reasoning

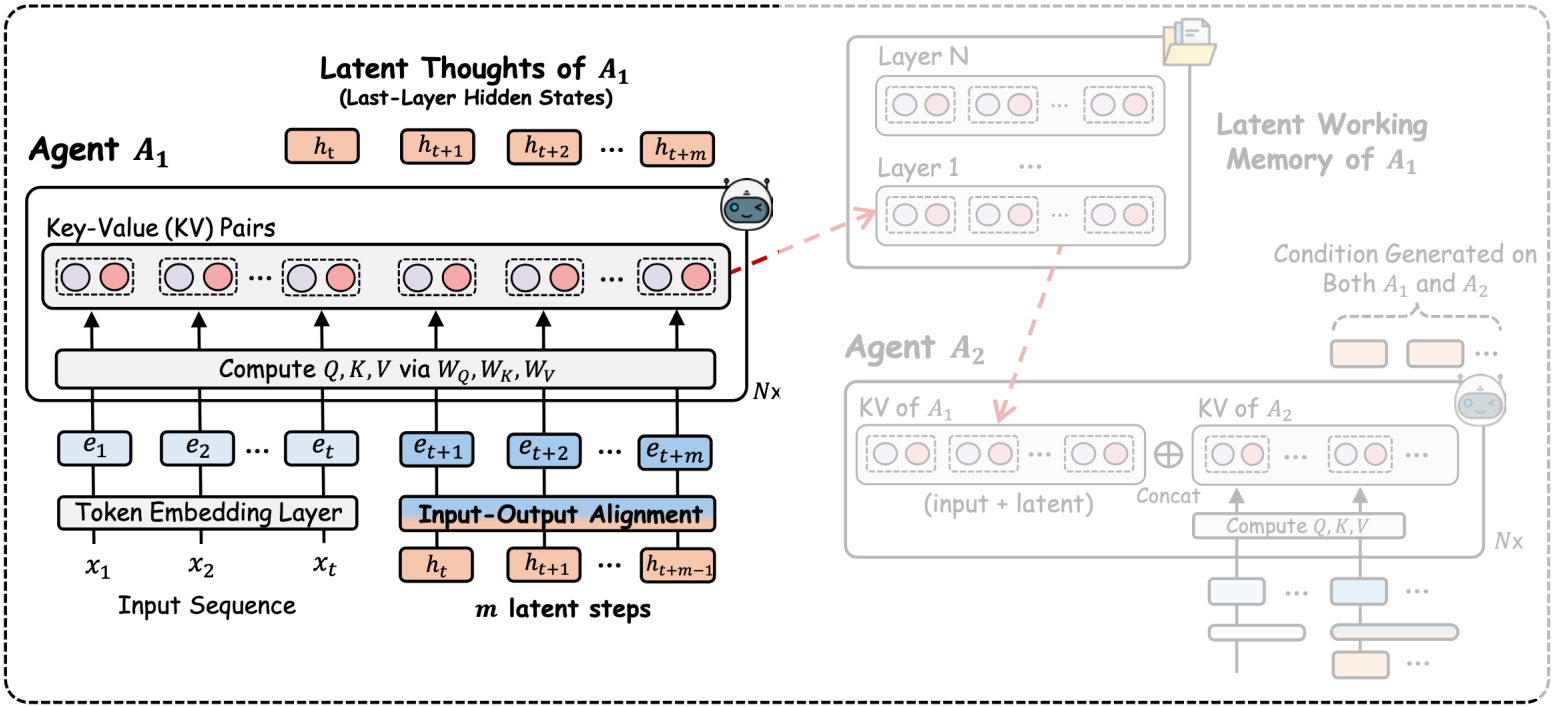
Step별 출력되는 마지막 레이어의 Hidden state를 다음 step의 입력으로 제공

- Hidden state를 입력 분포로 맞춰주기 위해 Alignment 수행 (Training-free)

$e_{t+1} = h_t \cdot W_a \,,\, W_a \approx W_{out}^{+} W_{in}$

✓ $h_t : 1 \times d_h \,,\, W_a : d_h \times d_h$

✓ W_{out}: LM head 가중치 / W_{in}: 토큰 임베딩 가중치



CVSP Lab
Computer Vision & Signal Processing

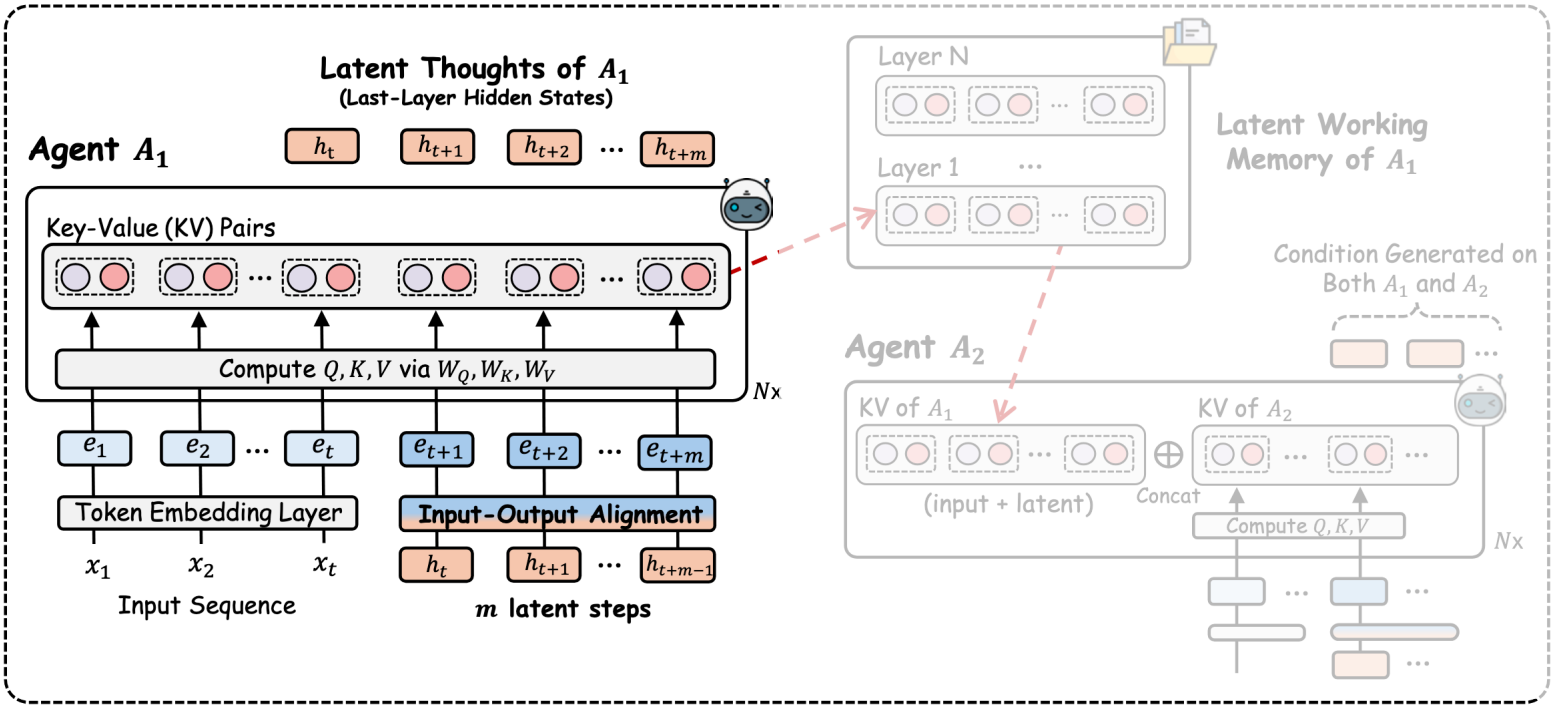
Method

- Latent Reasoning

  동일 수준의 표현을 Latent reasoning이 Text reasoning보다 더 효율적으로 수행

  - Latent reasoning step 대비 Text reasoning step이 최소 $\frac{d_h}{\log|\mathcal{V}|}$배 증가

    ⁘ Text reasoning은 text로 디코딩하는 과정에서 표현력 감소



CVSP Lab
Computer Vision & Signal Processing

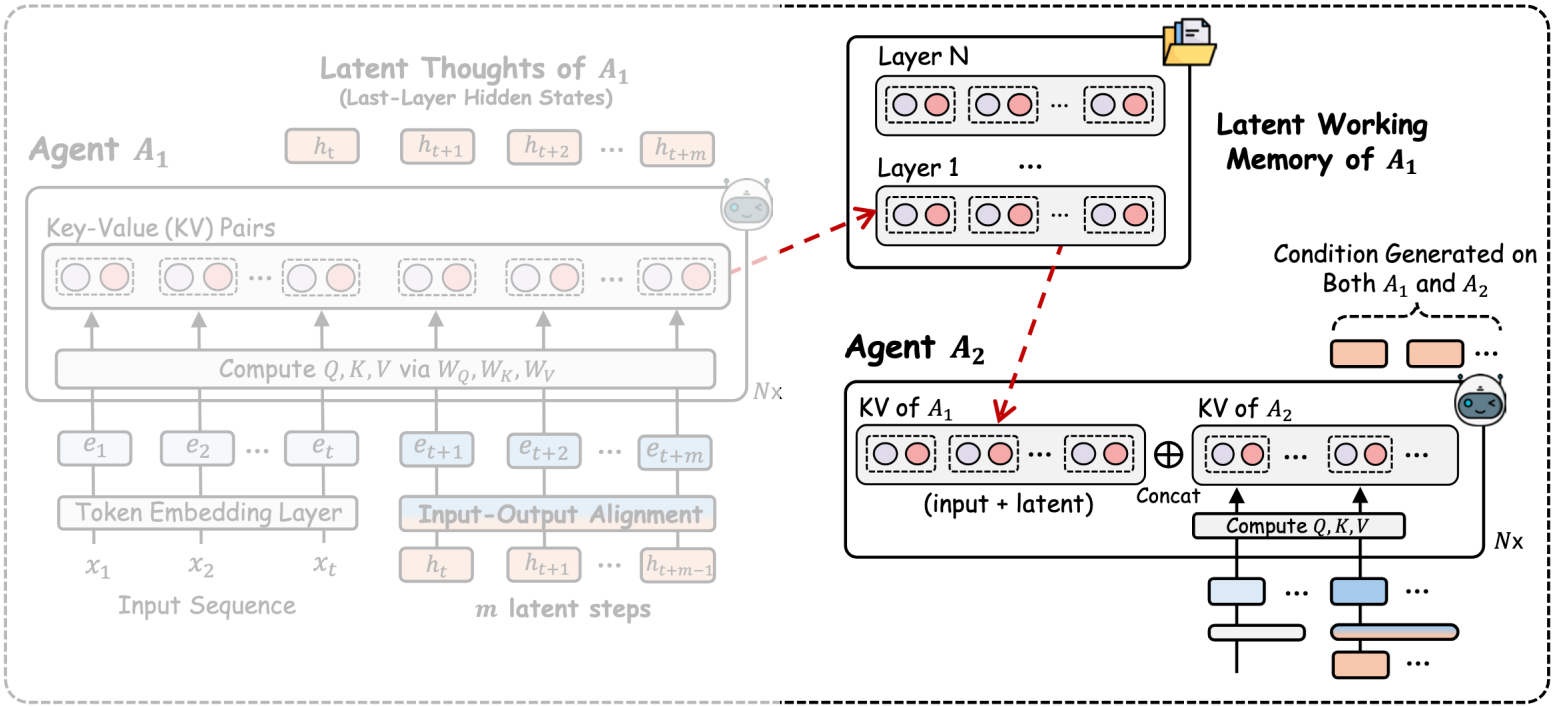
Method

- Latent Communication

  이전 에이전트의 각 step당 저장된 KV cache를 다음 에이전트에게 전달

  - $M_{A1} = \left\{\left(K_{A1,cache}^{(l)}, V_{A1,cache}^{(l)}\right)\right\}_{l=1}^{L}$ 를 다음 에이전트의 prefix로 제공

$$(K_{A1,cache}^{(l)}, V_{A1,cache}^{(l)}) = [\underbrace{\left(K_1^{(l)}, V_1^{(l)}\right), \ldots, \left(K_t^{(l)}, V_t^{(l)}\right)}_{\text{Prefill}}, \underbrace{\left(K_{t+1}^{(l)}, V_{t+1}^{(l)}\right), \ldots, \left(K_{t+m}^{(l)}, V_{t+m}^{(l)}\right)}_{\text{Reasoning}}]$$



CVSP Lab
Computer Vision & Signal Processing

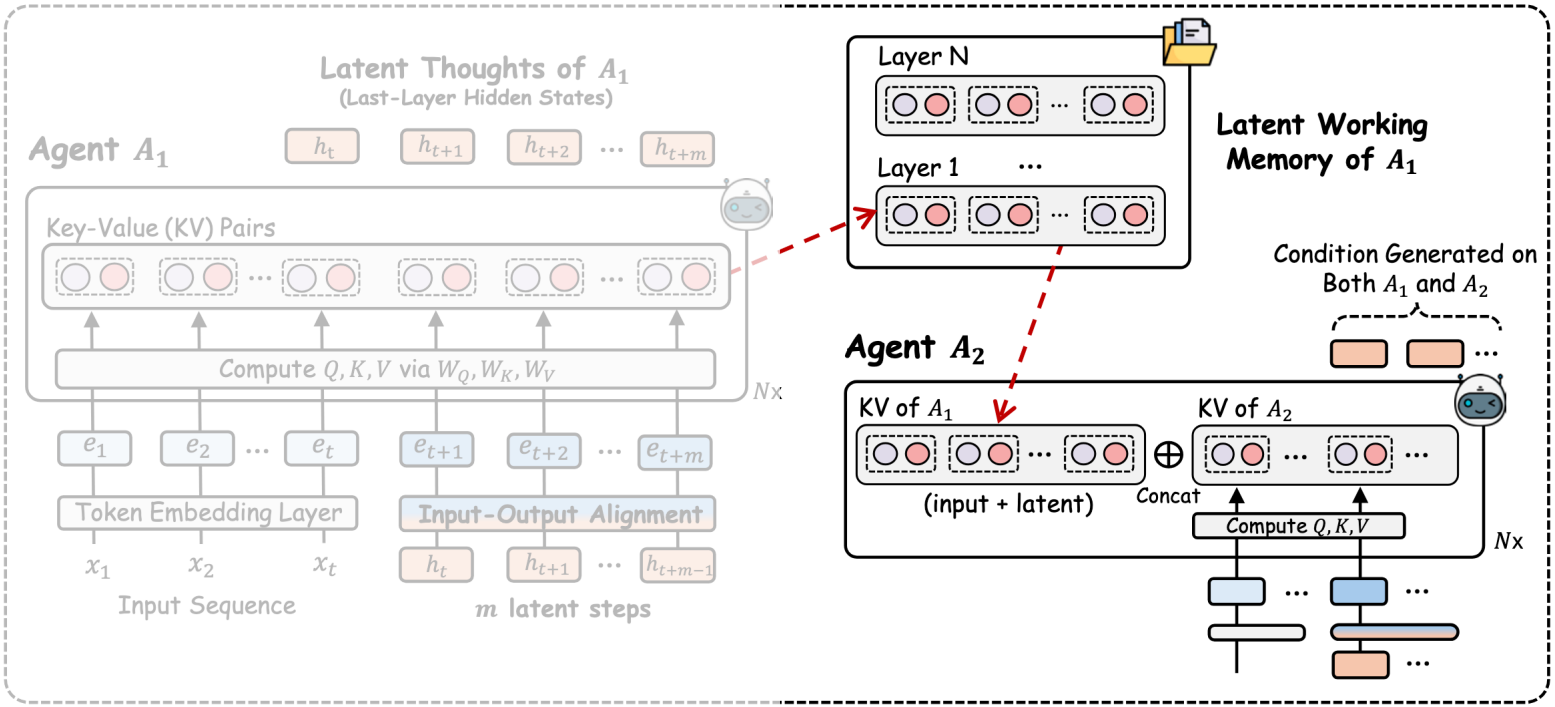
Method

- Latent Communication

  이전 에이전트의 각 step당 저장된 KV cache를 다음 에이전트에게 전달

  - 이전 에이전트가 전달한 정보를 다시 encoding하지 않음

    ⁂ 추론 속도 향상, 토큰 사용량 감소



CVSP Lab
Computer Vision & Signal Processing

Experiments

- Quantitative Results

다양한 Reasoning 벤치마크에서의 성능 및 효율성 평가

- Token: 최종 출력 토큰 사용량
- Speed: End-to-end 평균 추론 속도 (sec.)

Tasks	Metrics	Qwen3-4B			Improve	Qwen3-8B			Improve	Qwen3-14B			Improve
		Single	TextMAS	LatentMAS		Single	TextMAS	LatentMAS		Single	TextMAS	LatentMAS	
Sequential MAS Setting													
ARC-E	Acc.	95.4	96.4	**98.6**	↑ **2.2**	95.6	99.1	98.8	↓ 0.3	97.2	99.0	**99.4**	↑ **0.4**
	Token	724	2420	**581**	↓ **76.0%**	656	2085	**490**	↓ **76.5%**	608	1670	**224**	↓ **86.6%**
	Speed	369	2874	**512**	× **5.6**	404	3702	**1759**	× **2.1**	551	9171	**2124**	× **4.3**
ARC-C	Acc.	89.2	90.0	**92.3**	↑ **2.3**	91.0	94.6	94.4	↓ 0.2	92.6	95.9	95.6	↓ 0.3
	Token	913	2678	**718**	↓ **73.2%**	846	2252	**529**	↓ **76.5%**	773	2985	**426**	↓ **85.7%**
	Speed	97	1579	**260**	× **6.1**	266	2059	**703**	× **2.9**	338	5125	**1136**	× **4.5**
GSM8K	Acc.	82.4	89.8	88.2	↓ 1.6	81.1	92.3	**93.8**	↑ **1.5**	83.7	93.8	**95.2**	↑ **1.4**
	Token	1136	3172	**607**	↓ **80.9%**	1280	2324	**860**	↓ **63.0%**	1118	3324	**644**	↓ **80.6%**
	Speed	469	1970	**375**	× **5.3**	449	1739	**543**	× **3.2**	536	3729	**1952**	× **1.9**
MedQA	Acc.	47.7	65.3	**66.3**	↑ **1.0**	53.0	75.0	**75.3**	↑ **0.3**	64.7	80.3	**80.7**	↑ **0.4**
	Token	2134	3962	**1685**	↓ **57.5%**	2098	4260	**1555**	↓ **63.5%**	1746	3444	**1841**	↓ **46.5%**
	Speed	236	1267	**438**	× **2.9**	476	1923	**928**	× **2.1**	1360	4142	**1420**	× **2.9**
MBPP+	Acc.	63.5	69.8	**73.5**	↑ **3.7**	64.8	69.5	**74.6**	↑ **5.1**	68.5	72.8	**75.7**	↑ **2.9**
	Token	1634	4420	**1339**	↓ **69.7%**	2053	3695	**1164**	↓ **68.5%**	1858	4971	**1621**	↓ **67.4%**
	Speed	523	2148	**577**	× **3.7**	1064	3628	**1275**	× **2.8**	2410	8728	**2400**	× **3.6**
HumanEval+	Acc.	75.0	79.7	**79.9**	↑ **0.2**	74.4	80.5	**80.5**	↑ 0.0	76.8	81.1	**86.5**	↑ **5.4**
	Token	2380	5987	**1775**	↓ **70.4%**	2507	4593	**1866**	↓ **59.4%**	2366	5934	**2042**	↓ **65.6%**
	Speed	274	1044	**350**	× **3.0**	502	1619	**497**	× **3.3**	1084	4062	**1285**	× **3.2**

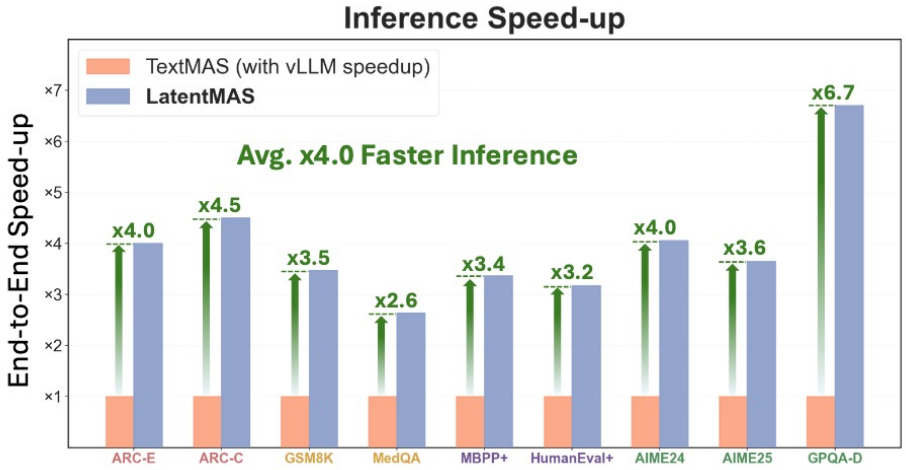
CVSP Lab
Computer Vision & Signal Processing

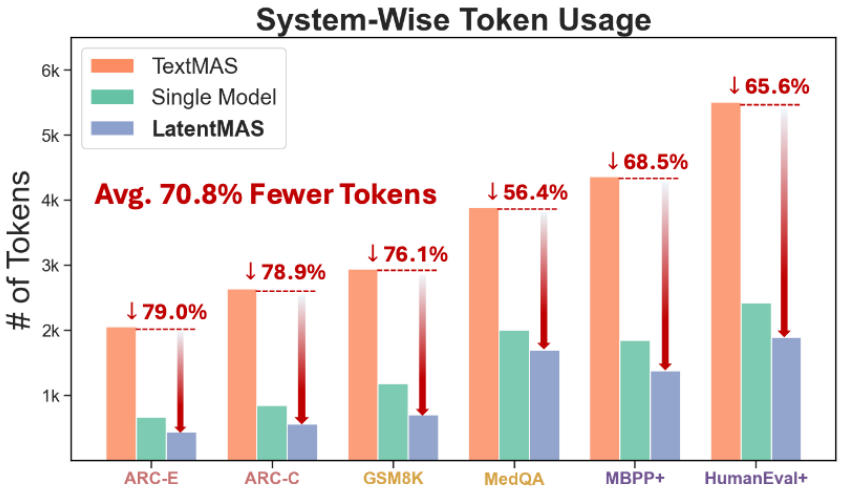
Experiments

- Quantitative Results

  다양한 Reasoning 벤치마크에서의 효율성 평가

  -추론 속도, 토큰 사용량 측면에서 일관된 향상





CVSP Lab
Computer Vision & Signal Processing

Experiments

- Ablation study

  Reasoning과 Communication 매개체 관련 실험 (Latent vs. Text)

  - Latent representation은 text보다 더 풍부한 정보를 보존 → 성능 향상으로 연결

Method	GSM8K	MBPP+	MedQA
Latent Reasoning + Text Communication	85.5	66.4	65.9
Text Reasoning + Latent Communication	90.1	68.0	71.2
LatentMAS	**93.8**	**74.6**	**75.3**

ToolOrchestra: Elevating Intelligence via Efficient Model and Tool Orchestration

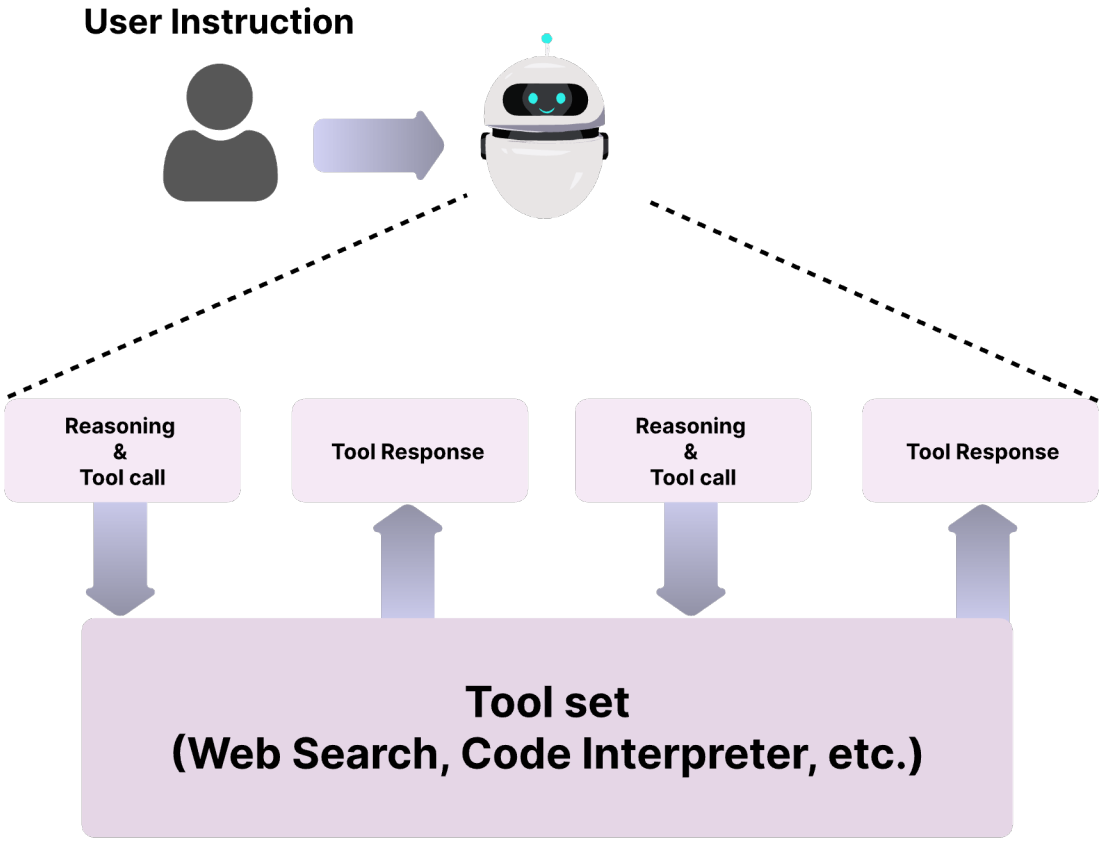
[ICML 2026]

Background

- Tool call framework

  매 step마다 어떤 tool을 사용할지 결정

  - 순차적 의사결정 과정



CVSP Lab
Computer Vision & Signal Processing

Introduction

- Motivation

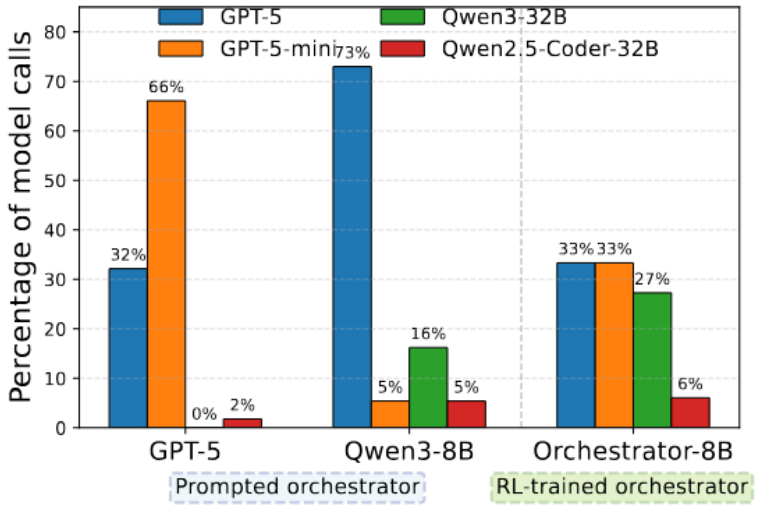
  Tool으로 다양한 크기의 LLM을 사용하기 시작

  - LLM의 크기에 비례하는 cost가 생김 → 비용을 고려한 tool call이 필요

  모델에 prompt를 제공하는 방법으로 tool call 개선 시도 → 특정 편향 존재

  - (1) Tool call을 하는 모델과 동일한 모델을 부르려는 편향
  - (2) 비용을 고려하지 않고 사용할 수 있는 가장 강력한 모델을 사용하려는 편향

  경량 모델에 RL 적용 → Tool call 편향을 줄이고 비용을 고려하는 Orchestrator 제안

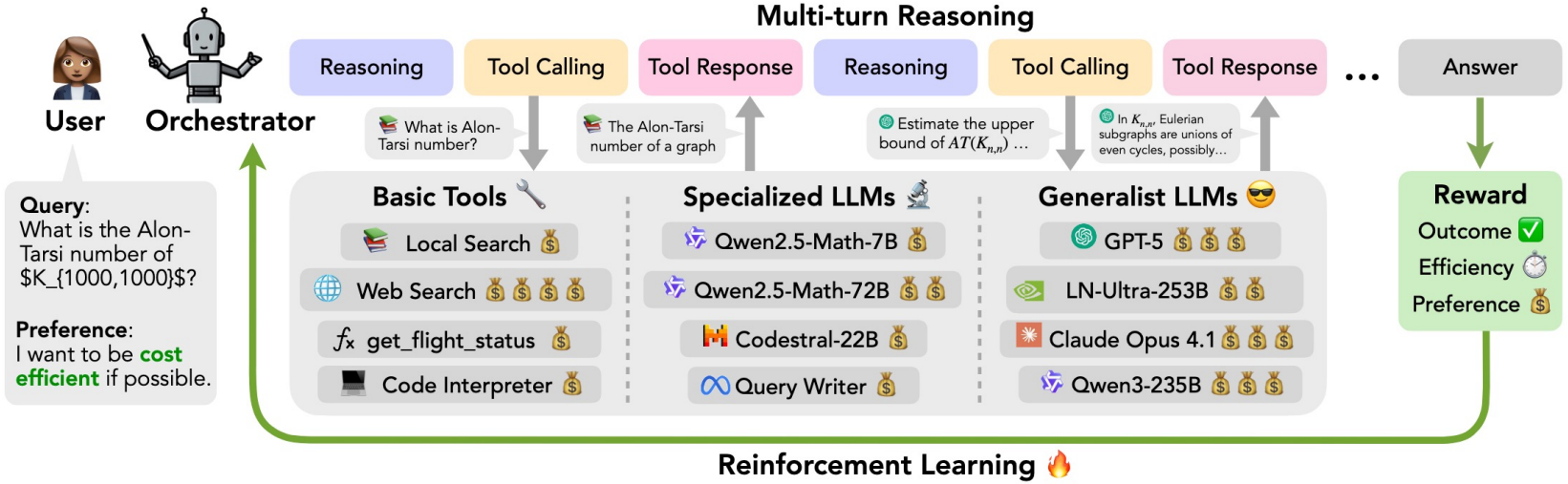


CVSP Lab
Computer Vision & Signal Processing

Method

- Overview

  정확도, 효율성, 사용자 선호도를 고려한 학습 진행



CVSP Lab
Computer Vision & Signal Processing

Method

- Multi-turn Rollout

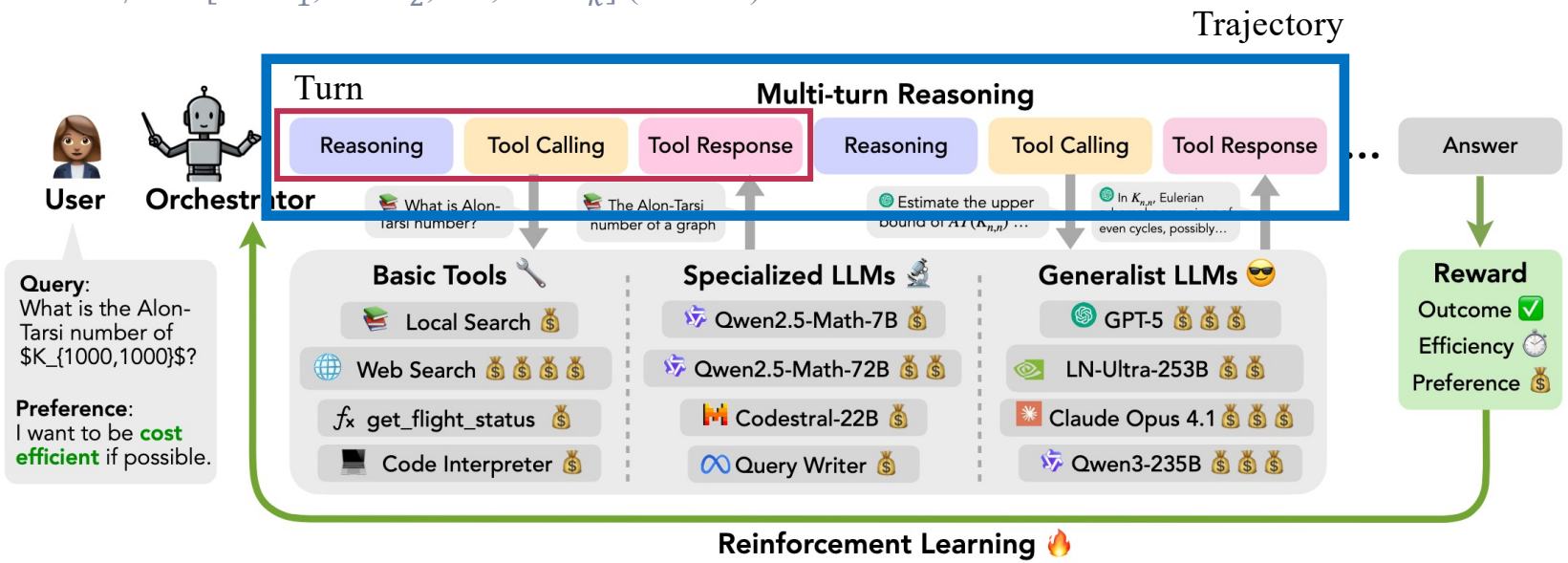
Turn

- 1) Reasoning → 2) Tool call (Action) → 3) Tool response (Observation) 순으로 구성

Rollout (Trajectory, τ)

- 여러 Turn으로 구성됨

$\tau = [\text{Turn}_1, \text{Turn}_2, \dots, \text{Turn}_k]\ (k \leq 50)$



CVSP Lab
Computer Vision & Signal Processing

Method

- Reward Design

  Outcome reward : $r_{outcome}(\tau)$

  - 아래 세 가지 조건을 모두 만족해야 1, 이외는 0
    - 1. GT trajectory와 생성 trajectory를 각각 실행 시 최종 결과가 같아야 한다.
    - 2. trajectory에 반드시 들어가야 하는 정보 o가 존재해야 한다.
    - 3. trajectory를 실행 시 GT trajectory가 조작한 대상들을 동일하게 조작해야 한다.

  Efficiency reward : $r_{compute}(\tau), r_{latency}(\tau)$

  - $r_{compute}(\tau) = -\ \$(\tau)$
    - $\$(\tau)$: trajectory 전체의 tool 및 model 호출 비용
  - $r_{latency}(\tau) = -\ \text{Clock}(\tau)$
    - $\text{Clock}(\tau)$: 전체 wall-clock time

CVSP Lab
Computer Vision & Signal Processing

Method

6) Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

- Reward Design

Final reward

- Trajectory batch 내 Trajectory 평가 벡터들을 항목별 min-max 정규화

  - Trajectory 평가 벡터 $M^{\tau_t} = [m_{t_1}^{\tau_t}, m_{t_2}^{\tau_t}, \dots, m_{t_n}^{\tau_t}, r_{outcome}(\tau_t), r_{compute}(\tau_t), r_{latency}(\tau_t)]$

    ✓ $M^{\tau_t}[k]$ $(1 \le k \le n)$: 각 tool의 사용 횟수

  - $1 \le k \le n+3$일 때, $M_{normalized}^{\tau}[k] = \frac{M^{\tau}[k] - M_{min}^{T}[k]}{M_{max}^{T}[k] - M_{min}^{T}[k]}$

- Preference vector

  - $\mathrm{P} = [p_{t_1}, \dots, p_{t_n}, p_{outcome}, p_{compute}, p_{latency}]$ $(0 \le p_* \le 1)$

    ✓학습 데이터 내 Preference Instruction과 mapping됨

- Final reward

  - $$\mathrm{R}(\tau) = \begin{cases} M_{normalized}^{\tau} \cdot P & if\ r_{outcome}(\tau) \\ 0 & otherwise. \end{cases}$$

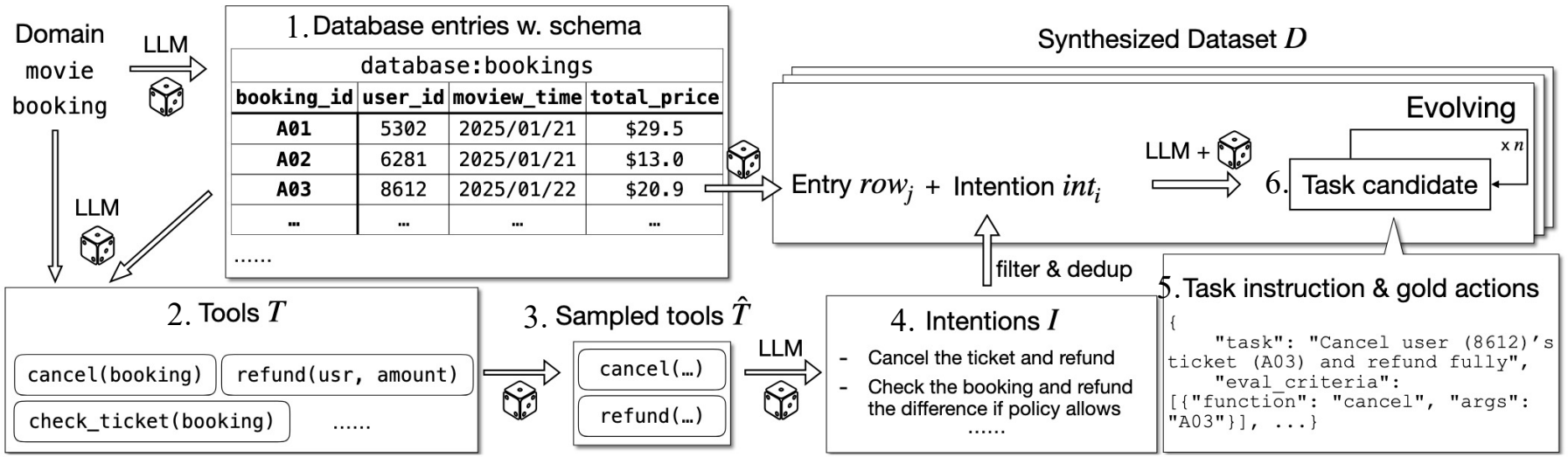
이후 GRPO[6]로 학습 진행

CVSP Lab
Computer Vision & Signal Processing

Method

- ToolScale

  논문의 RL reward 형식으로 데이터를 생성하기 위한 Data synthesis framework



CVSP Lab
Computer Vision & Signal Processing

Experiments

- Quantitative Results

  각 벤치마크 성능 및 Cost(cents)와 Latency(minutes)

  - 동일한 Tool pool 내에서, Orchestrator-8B가 성능, 비용, 속도 측면에서 가장 좋음

Tools	Model(s)	HLE (↑)	FRAMES (↑)	τ^2-Bench (↑)	Cost (↓)	Latency (↓)
Existing baselines	GPT-5	35.2	–	84.2‡	–	–
	o3	24.3	–	68.4	–	–
	GPT-4o	5.3	–	43.8	–	–
	ToRL	0.9	14.8	29.8	0.2	0.5
	RouteLLM	7.2	28.3	24.9	13.4	8.7
No tool	Qwen3-8B	3.2	24.2	–*	0.2	0.6
	Llama-Nemotron-49B	3.6	25.6	–*	0.4	1.1
	Llama-3.3-70B	3.8	32.4	–*	0.5	1.4
	Qwen3-235B-A22B	5.2	34.3	–*	2.6	3.3
	Claude Opus 4.1	11.7	58.2	–*	27.4	8.2
	GPT-5	23.4	66.3	–*	6.2	4.1
Basic tools	ToolLlaMA-2-7b-v2	1.3	12.6	11.4	1.0	1.6
	Qwen3-8B	4.7	26.5	40.7	1.3	2.2
	Llama-Nemotron-49B	6.8	28.2	23.2	2.5	3.5
	Llama-3.3-70B	4.6	42.3	17.6	2.8	4.3
	Qwen3-235B-A22B	14.0	39.5	52.9	12.3	10.2
	Claude Opus 4.1	19.8	63.5	46.0	76.2	32.5
	GPT-5	35.1	74.0	77.7	30.2	19.8
	RestGPT	35.6	74.3	78.2	47.4	29.7
Basic tools, Specialized LLMs Generalist LLMs	ToolLlaMA-2-7b-v2	18.8	48.6	52.4	16.2	11.3
	Qwen3-8B	30.6	68.9	72.3	27.6	18.3
	Llama-Nemotron-49B	25.8	57.9	66.7	25.6	17.1
	Llama-3.3-70B	19.7	52.4	55.8	19.7	13.4
	Qwen3-235B-A22B	32.8	74.2	75.6	29.7	21.2
	Claude Opus 4.1	34.6	72.8	76.8	52.5	25.6
	GPT-5	21.2	57.5	62.3	17.8	13.6
	RestGPT	21.6	57.7	62.8	27.2	18.9
	Orchestrator-8B	**37.1**	**76.3**	**80.2**	**9.2**	**8.2**

CVSP Lab
Computer Vision & Signal Processing

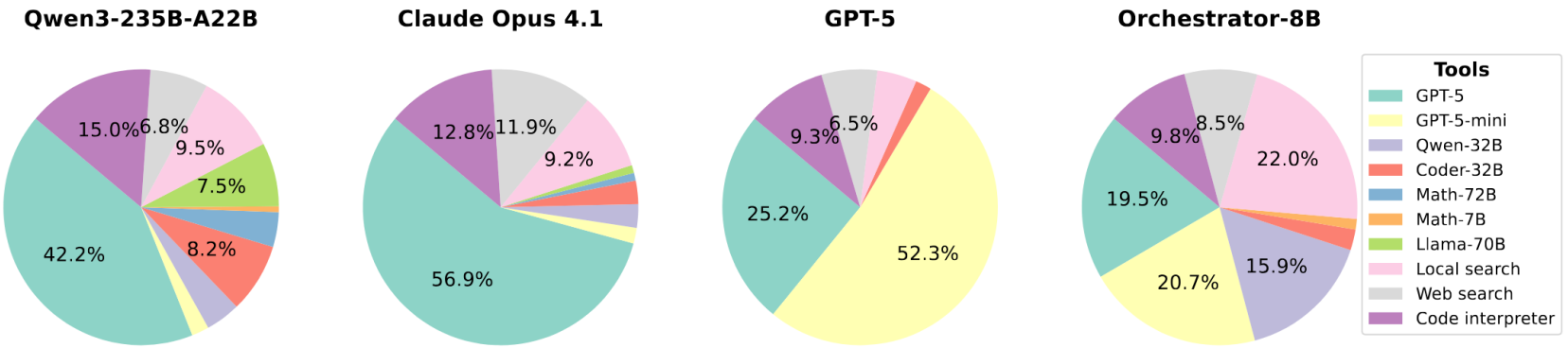
Experiments

- Quantitative Results

  각 모델 별 Tool call 비율

  - HLE, FRAMES, τ^2-bench 측정 시 발생한 평균 tool call 비율

    - 다른 모델을 orchestra로 사용 → 편향 존재
    - Orchestrator-8B → 편향 없이 균형 있는 tool call 수행



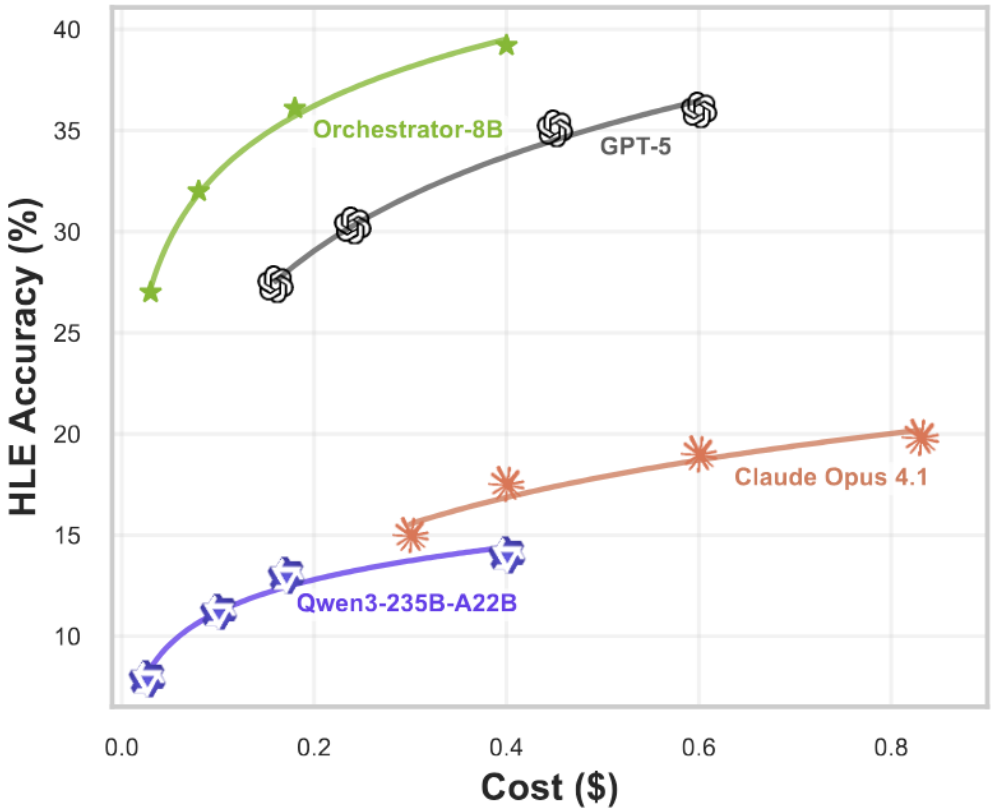
CVSP Lab
Computer Vision & Signal Processing

Experiments

- Quantitative Results

  비용 당 HLE 정확도 그래프

  - Orchestrator-8B가 저비용, 고성능 tool call 수행 확인



CVSP Lab
Computer Vision & Signal Processing

Appendix

- Latent Thoughts의 표현력

  동일 표현 시, Latent reasoning step 대비 Text reasoning step이 최소 $\frac{d_h}{\log|\mathcal{V}|}$배 증가

Theorem 3.1 (Expressiveness of Latent Thoughts). *Under the Linear Representation Hypothesis on h (stated in Assumption B.1), if the sequence of all latent thoughts with length m can be expressed losslessly through corresponding text-based reasoning, then the length of text (in tokens) needs to be at least $\Omega\left(d_h m / \log|\mathcal{V}|\right)$, where $|\mathcal{V}| > 1$ denotes the vocabulary size.*

CVSP Lab
Computer Vision & Signal Processing