LLMs for Multi-turn Dialogue

손수현

REINFORCING MULTI-TURN REASONING IN LLM AGENTS VIA TURN-LEVEL REWARD DESIGN

Quan Wei[1*] **Siliang Zeng**[1*] **Chenliang Li**[2] **William Brown**[3] **Oana Frunza**[4]
Wei Deng[4] **Anderson Schneider**[4] **Yuriy Nevmyvaka**[4] **Yang Katie Zhao**[1]
Alfredo Garcia[2] **Mingyi Hong**[1]

[1]University of Minnesota [2]Texas A&M University [3]Prime Intellect [4]Morgan Stanley

StructFlowBench: A Structured Flow Benchmark for Multi-turn Instruction Following

Jinnan Li[1,5] **Jinzhe Li**[2] **Yue Wang**[3] **Yi Chang**[1,4,5*] **Yuan Wu**[1*]

[1]School of Artificial Intelligence, Jilin University
[2]College of Computer Science and Technology, Jilin University
[3]School of Information and Library Science, University of North Carolina at Chapel Hill
[4]Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, MOE, China
[5]International Center of Future Science, Jilin University
{jnli23, lijz2121}@mails.jlu.edu.cn, wangyue@email.unc.edu,
yichang@jlu.edu.cn, yuanwu@jlu.edu.cn

Motivation

- Multi-turn LLM agents를 위해서 GRPO나 PPO같은 RL algorithms이 적용되고 있음
- 하지만 보통 마지막 결과(정답 맞았는지) 같은 outcome-only 보상만 주는 경우가 많음

  → 보상이 sparse해서 어느 턴의 행동이 성공/실패에 기여했는지를 가르치는 credit assignment가 잘 안 됨
- 멀티턴 에이전트 RL에서 성능이 제한되는 이유는 결국 턴 단위 촘촘한 reward를 주지 않기 때문에

➔ 멀티턴에서 turn-level reward를 설계해 중간중간 피드백을 주면, RL이 어떤 단계가 기여했나를 더 잘 알 수 있다!

➔ GRPO와 PPO를 멀티턴 버전으로 확장하고, 턴 단위 보상을 통합하는 방식을 제안함

Method

1. GRPO WITH TURN-LEVEL REWARDS FOR MULTI-TURN AGENTIC TASKS

- 기존의 multi-turn agentic LLM의 수식을 보면

1. single turn problem으로 formulate하는 경우

→ multi-turn(검색→추론→검색→답)이어도, 이 식에서는 그 과정을 쪼개지 않고 "한 번의 선택"처럼 취급함

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} [R(x,y)]$$

2. 턴별로 쪼개서 formulate하는 경우

→ 만약 turn별 중간 reward가 0이고 γ 가 1이면 결국 이것도 단일턴과 다를게 없다.

$$\max_{\pi_\theta} \mathbb{E}_{s_k,\, a_k\sim\pi_\theta(\cdot|s_k)} \left[\sum_{k=1}^{K} \gamma^k R(s_k, a_k)\right]$$

그렇다고 GRPO 수식을 그대로 적용하면?

trajectory는 "질문 x 하나에 대해, 모델이 끝까지 만든 전체 응답 y" → 멀티턴이면 전체 history

하지만, trajectory-level reward니까 전체 history y에 대해서 주는 reward는 1개 → 어떤 턴이 문제였는지 구분X

➔ 그니까 "turn-level reward 설계 + 멀티턴용 GRPO"가 필요하다!

Method

2. MT-GPRO: TURN-LEVEL CREDIT ASSIGNMENT FOR GRPO

- MT-GRPO

simple two-turn agent setting (K = 2), 첫번째 turn reward $\{R_i^I\}_{i=1}^G$, 두번째 turn reward $\{R_i^O\}_{i=1}^G$

turn-level advantages

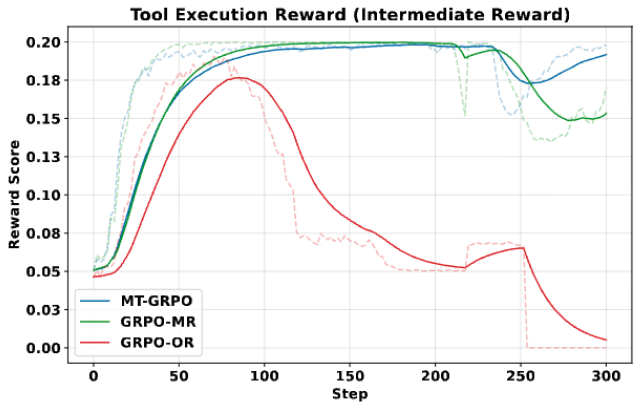
$$A_{i,1}^{\text{MT-GPRO}} = A_i^I + \alpha A_i^O, \quad A_{i,2}^{\text{MT-GPRO}} = A_i^O$$

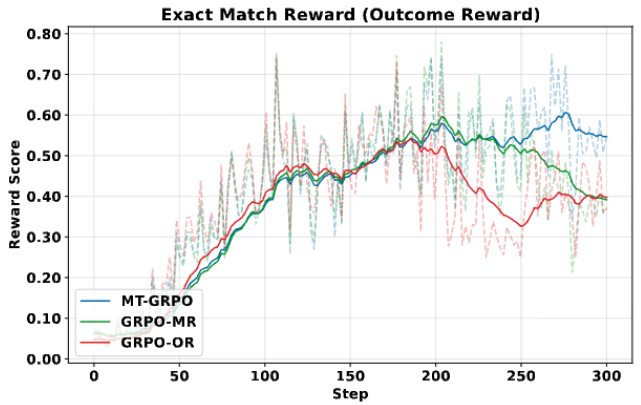
$$A_i^I = \frac{R_i^I - \text{mean}(\{R_i^I\}_{i=1}^G)}{\text{std}(\{R_i^I\}_{i=1}^G)}, \quad A_i^O = \frac{R_i^O - \text{mean}(\{R_i^O\}_{i=1}^G)}{\text{std}(\{R_i^O\}_{i=1}^G)}$$

[Case Study for MT-GRPO on a Two-Turn Agentic Task]

(1턴) reasoning + search tool 호출 → 검색 결과 반환

(2턴) retrieved result 기반 reasoning → final answer 출력





GRPO-OR: outcome reward만 사용

GRPO-MR: intermediate + outcome을 합쳐서(merged) trajectory-level 보상 1개로 만듦

MT-GRPO(제안하는): intermediate/outcome을 턴 단위 advantage로 분리해서 credit assignment를 더 세밀하게

→ MT-GRPO가 툴을 더 안정적으로, 꾸준히 제대로 호출

→ MT-GRPO가 exact match가 더 높게

Method

2. MT-GPRO: TURN-LEVEL CREDIT ASSIGNMENT FOR GRPO

- MT-GRPO

하지만 MT-GRPO는 한계가 존재함

1. 턴이 증가할수록 롤 아웃수가 폭증함

각 턴별로 G개의 롤아웃 * 그거에따라 달라지는 다음 턴 상태에 맞게 또 G번

⇒ horizon이 긴 멀티턴 작업에는 계산적으로 너무 비싸다

2. 그룹 안의 모든 rollout이 같은 턴 수를 갖도록 강제해야 함

같은 턴 k에서 나온 intermediate reward들끼리 비교해서 평균보다 낮고 높음을 해야하는데

G번 안에서 어떤 답변을 생성하는냐에 따라서 어떤 샘플은 1턴만에 끝나고, 어떤 샘플은 3턴 → 비교에 공정X

⇒ 같은 턴 수를 강제하면 유연성이 떨어짐

→ 지수적 롤아웃 비용 + fixed-turn 제약 때문에 일반 에이전트에 쓰기 어렵다

Method

3. PPO WITH TURN-LEVEL REWARDS FOR MULTI-TURN AGENTIC TASKS

- MT-PPO

그럼 좀 더 현실적으로 쓸 수 있는 방법은? → PPO

PPO objective (clipped surrogate)

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{\text{old}}(\cdot|x)} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} \min\left(w_t(\theta)A_t,\ \text{clip}\left(w_t(\theta),\ 1-\epsilon,\ 1+\epsilon\right)A_t\right) \right]$$

- 차이: reward 설계

Turn-level rewards: 중간 턴 R^I, 최종 결과 R^O

$$r_t = \begin{cases} R^O & \text{if } t \text{ is the last token of the entire trajectory} \\ R^I & \text{if } t \text{ is the last token of the intermediate turn} \\ 0 & \text{otherwise} \end{cases}$$

reward은 턴 경계에만 주지만, GAE(Generalized Advantage Estimation)가 그 신호를 앞 토큰들로 분배

이를 시간방향으로 누적해서 A_t

$$A_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l}, \quad \delta_t = r_t + \gamma V_{t+1} - V_t$$

Case Study

MULTI-TURN REASONING-AUGMENTED SEARCH AGENT

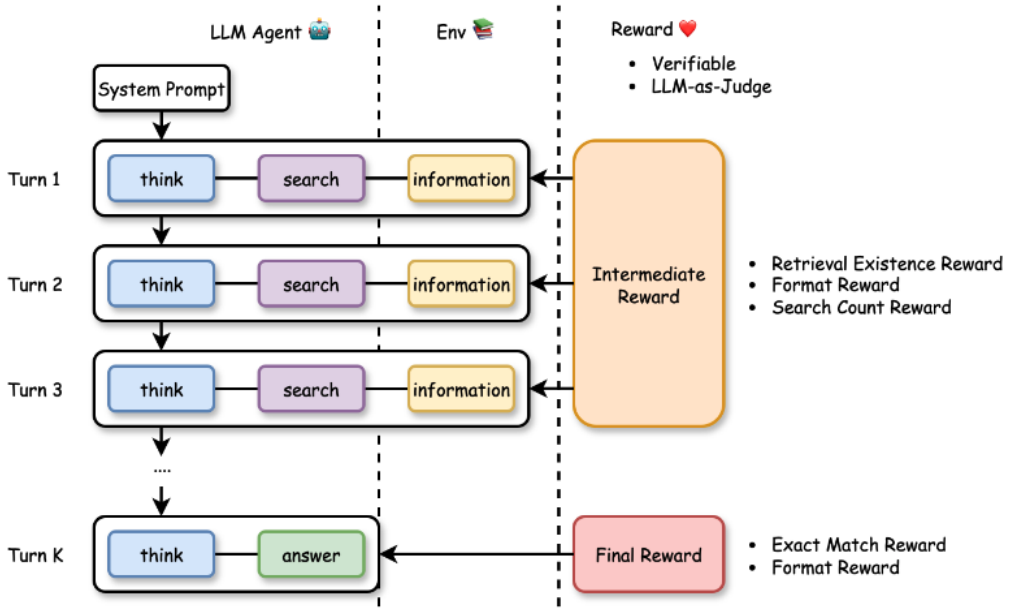
- 매 iteration에서

1) reasoning으로 현재 컨텍스트에서 부족한 정보를 판단

2) 검색 쿼리 생성

3) 외부 DB(Wikipedia search)에서 결과를 받아 컨텍스트에

→ 충분하다고 판단되면 마지막에 최종 답변 생성



- TURN-LEVEL VERIFIABLE REWARD DESIGN

기존 연구들처럼 final-answer correctness만 보지 않겠다. 중간 reward도 고려하겠다

Outcome Verifiable Rewards

: Outcome Exact Match Reward, Outcome Format Reward

Intermediate Verifiable Rewards

: Intermediate Retrieval Existence Reward (검색 결과에 정답 문자열이 포함되었는지)

: Intermediate Format Reward (중간 턴 출력이 포맷에 맞고, 규칙대로 썼는지)

: Intermediate Search Count Reward (검색을 너무 많이 하지 않게 누적 검색 횟수에 비례해 깎는 항)

Experiments

- Training Details

Base model: Qwen2.5-7B

Retrieval: E5

Corpus: 2018 Wikipedia dump

Datasets: NQ, HotpotQA

Metric: (1) answer correctness (EM) reward

(2) format correctness reward

(3) retrieval correctness reward

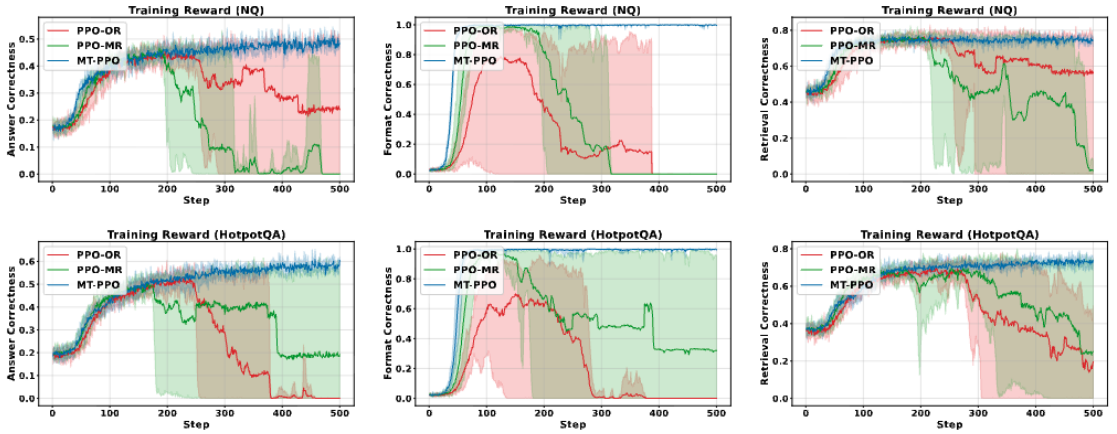


Figure 3: Training reward curves recorded during training for PPO baselines and MT-PPO on the NQ and HotpotQA datasets. The rewards include answer correctness, format correctness, and retrieval correctness. Solid lines show mean reward values, while shaded regions indicate variability across five independent runs.

- Training Dynamics

- MT-PPO가 초반 수렴이 더 빠름
- Step이 지나면 PPO에서 분산이 더 커지고 성능 저하, 하지만 MT-PPO는 일관되게 성능을 유지함
- 특히 format reward에서 차이가 큼

Experiments

- Benchmark Performance

- 정답률 향상
- + format 을 맞추지 못해서 평가/학습이 깨지는 문제를 거의 제거해서 학습의 안정성과 실전 출력 품질이 같이 좋아졌음을 강조함

Methods	General QA			Multi-Hop QA			Avg.
	NQ†	TriviaQA*	PopQA*	HotpotQA†	2wiki*	Musique*	
Answer Correctness (Exact Match)							
Qwen2.5-7B-Base	0.177	0.319	0.181	0.160	0.167	0.040	0.174
Qwen2.5-7B-Instruct	0.320	0.563	0.349	0.292	0.277	0.118	0.320
GRPO-OR (Search-R1)	0.391	0.560	0.388	0.331	0.306	0.129	0.351
GRPO-MR (Search-R1)‡	0.453	0.628	0.450	0.416	0.375	0.164	0.414
PPO-OR (Search-R1)	0.483	0.639	0.456	0.435	0.382	0.199	0.432
PPO-MR (Search-R1)‡	0.472	0.629	0.452	0.436	0.402	0.180	0.429
GRPO (OTC)‡	0.444	0.597	0.431	0.366	0.311	0.130	0.380
PPO (OTC)‡	0.446	0.623	0.425	0.383	0.363	0.152	0.399
PPO (StepSearch)	0.355	0.570	0.385	0.351	0.396	0.179	0.373
MT-PPO (ours)	**0.490**	**0.647**	**0.459**	**0.453**	**0.424**	**0.209**	**0.447**
Format Correctness							
Qwen2.5-7B-Base	0.118	0.118	0.105	0.098	0.084	0.082	0.101
Qwen2.5-7B-Instruct	0.183	0.267	0.067	0.109	0.037	0.071	0.122
GRPO-OR (Search-R1)	0.706	0.685	0.597	0.513	0.376	0.328	0.534
PPO-OR (Search-R1)	0.909	0.954	0.952	0.916	0.806	0.834	0.895
PPO (StepSearch)	0.521	0.614	0.668	0.560	0.396	0.571	0.555
MT-PPO (ours)	**0.999**	**0.997**	**0.999**	**0.998**	**0.999**	**0.999**	**0.999**

Experiments

- Ablation Study

- (1) reward design, (2) max turn

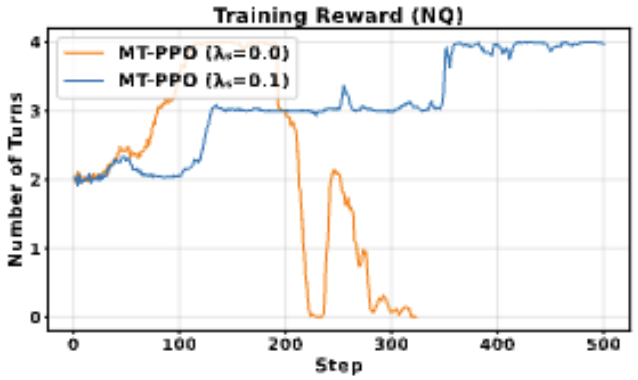
1) Reward Design

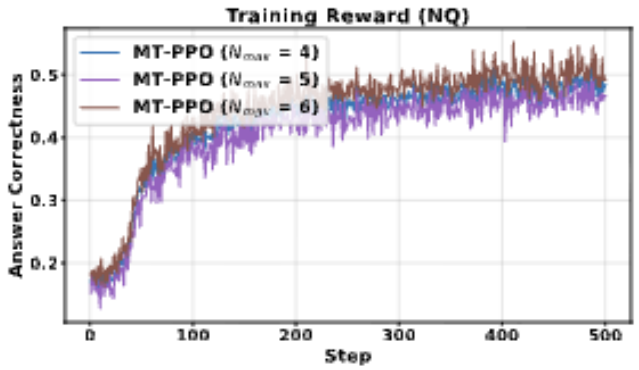
Search count reward가 없을 때 성능 변화

→ 패널티를 주지 않으면 과도한 검색/불안정으로 이해 성능이 특정 step 이후로 크게 하락함

2) max turn

턴을 늘려도 PPO의 성능 상승 트렌드의 유사함 -> 더 긴 multi-turn에서도 MT-PPO는 안정적으로 적용가능





Conclusion

- Multi-turn agentic 작업에서는 turn-level reward가 핵심임
- 이를 위해서 intermediate reward를 설계하고 GRPO/PPO를 multi-turn 으로 확장

  → 각 턴에서 더 세밀한 피드백을 받게
- reasoning-augmented search agent 실험에서, turn-level reward를 넣으면 여러 RL 알고리즘에서 학습 안정성과 정확도가 크게 개선

StructFlowBench: A Structured Flow Benchmark for Multi-turn Instruction Following

Jinnan Li[1,5] **Jinzhe Li**[2] **Yue Wang**[3] **Yi Chang**[1,4,5*] **Yuan Wu**[1*]
[1]School of Artificial Intelligence, Jilin University
[2]College of Computer Science and Technology, Jilin University
[3]School of Information and Library Science, University of North Carolina at Chapel Hill
[4]Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, MOE, China
[5]International Center of Future Science, Jilin University
{jnli23, lijz2121}@mails.jlu.edu.cn, wangyue@email.unc.edu,
yichang@jlu.edu.cn, yuanwu@jlu.edu.cn

Motivation

- LLM의 Multi-turn instruction following capability은 real-world application에서 core competency
- 하지만 기존 benchmark들은 주로
  - 형식/키워드/스타일과 같은 fine-grained constraint satisfaction
  - domain-specific capability

→ Turn들간의 structural dependency를 제대로 평가하지 못함

- 이러한 structural dependency는 user intent를 반영하고 있음
- 그러니까 if 평가는 constraint 만족만이 아니라 구조를 유지하는 능력도 고려해야한다!

- 그럼 구조를 평가하기 위해 구조를 정의하고 이를 기반으로 benchmark
- 기존 LLM들 분석 – fail case

Motivation

- 왜 structure가 중요한가

  기존 multi-turn 평가는 대화를 “single-turn들의 단순 연결”로 취급함

  → Failure to model complex scenarios

  → Methodological bias - inter-turn structural constraints을 놓치고, intra-turn constraints만 과대평가

  → Analytical deficiency

➔ 그래서 이 논문에서는 StructFlowBench 제안

  - structural flow modeling을 통해 turn 간 관계를 모델링

  - 여섯 가지 Structural Flow Taxonomy 정의

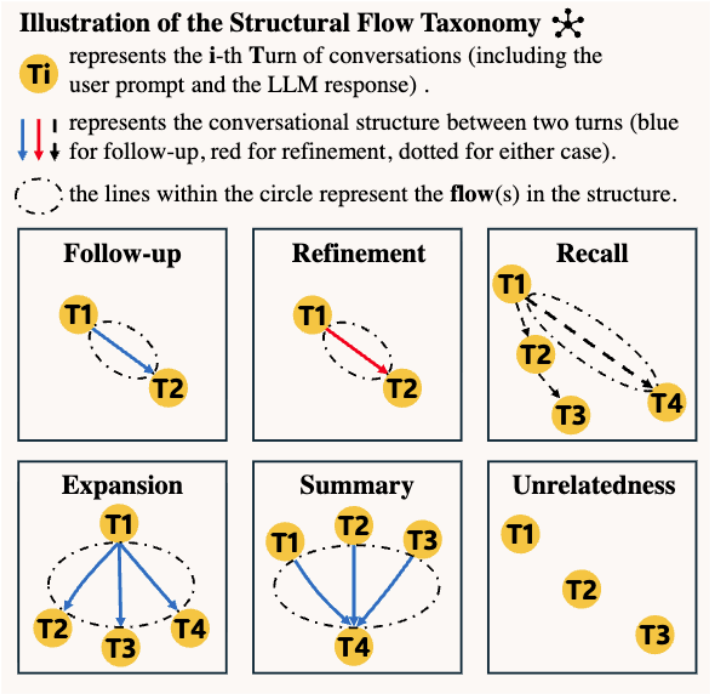
StructFlowBench
1. Structural Flow Taxonomy

- Structural Flow란

  각 user turn(또는 turn 쌍)을 "이 turn이 **이전** 어떤 turn과 어떤 방식으로 연결되는가"

  → 모델이 단순히 각 turn의 지시사항을 따르는지(intra-turn) 뿐 아니라, 이전 대화 맥락을 어떤 구조로 이어가며 유지/변형하는지도 고려하겠다

- Taxonomy



- Follow-up
  : 바로 직전 turn을 자연스럽게 이어 묻는 형태.
  : 직전 답변의 특정 부분을 더 자세히, 더 깊게 파고드는 추가 질문/추가 요구에 해당
- Refinement
  : 직전 user instruction을 수정해서 다시 요구하는 형태
  : 기존 요구를 더 잘 만족시키도록 constraints를 업데이트하는 것
- Recall
  : 2 turn 이상의 이전 정보를 다시 참조하는 형태
- Expansion
  : 하나의 주제에 대해서 말하다가 여러 subtopic으로 fan-out되는 형태
- Summary
  : 여러 이전 turn에서 나온 내용을 fan-in으로 합쳐서 요약/정리하는 형태
- Unrelatedness
  : 이전 맥락과 주제가 끊기고 갑자기 새 토픽으로 전환되는

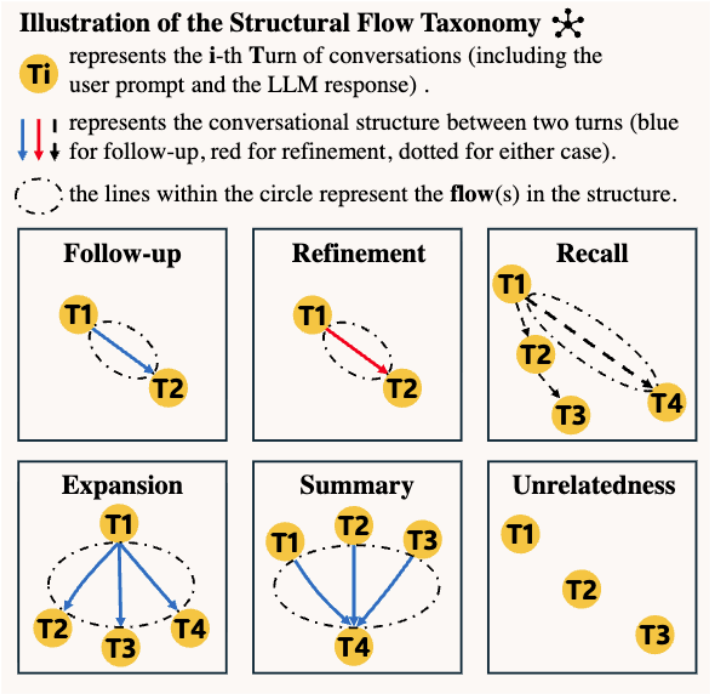
StructFlowBench
1. Structural Flow Taxonomy

- Structural Flow란

  각 user turn(또는 turn 쌍)을 "이 turn이 **이전** 어떤 turn과 어떤 방식으로 연결되는가"

  → 모델이 단순히 각 turn의 지시사항을 따르는지(intra-turn) 뿐 아니라, 이전 대화 맥락을 어떤 구조로 이어가며 유지/변형하는지도 고려하겠다

- Taxonomy



- Follow-up → 기존 맥락을 그대로 두고 더 묻는 느낌
  : 바로 직전 turn을 자연스럽게 이어 묻는 형태.
  : 직전 답변의 특정 부분을 더 자세히, 더 깊게 파고드는 추가 질문/추가 요구에 해당
- Refinement → 기존 요청을 고쳐서 다시 요구하는
  : 직전 user instruction을 수정해서 다시 요구하는 형태
  : 기존 요구를 더 잘 만족시키도록 constraints를 업데이트하는 것
- Recall
  : 2 turn 이상의 이전 정보를 다시 참조하는 형태
- Expansion
  : 하나의 주제에 대해서 말하다가 여러 subtopic으로 fan-out되는 형태
- Summary
  : 여러 이전 turn에서 나온 내용을 fan-in으로 합쳐서 요약/정리하는 형태
- Unrelatedness
  : 이전 맥락과 주제가 끊기고 갑자기 새 토픽으로 전환되는

StructFlowBench
2. Constraint Categories

- 정의된 관계들은 평가를 위한 새로운 structural constraints가 됨
- StructFlowBench의 평가 기준을 두 종류의 constraints로

(1) intra-turn constraints

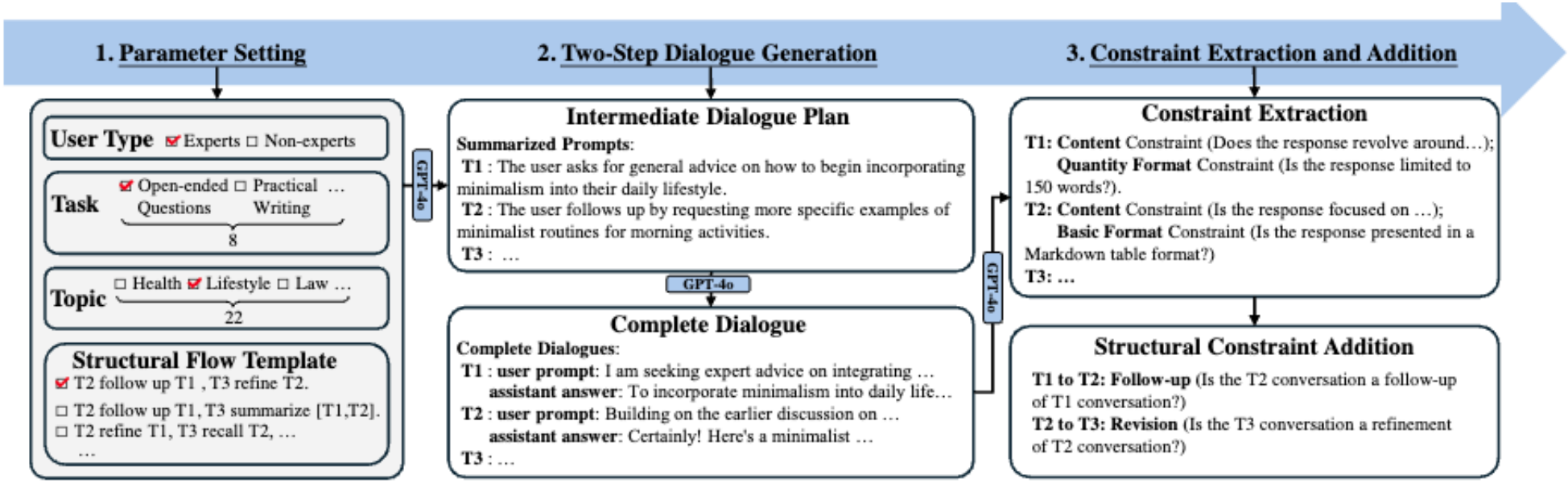
- Content Constraint: 답변이 지정된 content scope에만 집중하고 topic에서 벗어나지 않아야 함
- Keyword/Element Constraint: 답변에 “Artificial Intelligence”를 포함하라
- Style Constraint: 특정 writing style을 따라야 함
- Basic Format Constraint: 기본 출력 format을 지켜야 함
- Quantity Format Constraint: 정량적 길이 제약을 맞춰야 함
- Template Format Constraint: 정해진 template 구조를 따라야 함
- Situation Constraint: 특정 identity/role/context 같은 scenario/perspective에 맞춰 답해야 함
- Inverse Constraint: 의도적으로 포함하지 말아야 하는(avoid) 제약

(2) multi-turn structural constraints

- Follow-up Constraint, Refinement Constraint, Expansion Constraint, Summary Constraint, Recall Constraint

StructFlowBench

3. Data Construction Pipeline



StructFlowBench
4. Evaluation

- LLM을 어떻게 평가했는지
- Evaluation Criteria
    - Golden Context
        : 데이터셋의 gold dialog history를 context로 줌
    - Constraint decomposition + binary questions
        : multi-turn user instruction을 여러 개의 독립적인 constraints로 쪼갠 뒤, 각 constraint에 대해 만족 여부를 묻는 binary question형태로 checklist
        → checklist 결과를 모아서 최종 점수
    - LLM-as-a-judge
        : GPT-4o에게 golden context + test model response + constraint checklist + prompt template를 넣어서 평가

StructFlowBench
4. Evaluation

- Evaluation Metrics
  - CSR (Constraint Satisfaction Rate)
    : 전체 instruction에 대해, 각 instruction이 가진 constraints 중 몇 %를 만족했는지의 평균
  - ISR (Instruction Satisfaction Rate)
    : instruction 단위로 “그 instruction의 모든 constraints를 전부 만족했는가”
  - DRFR (Decomposed Requirements Following Ratio)
    : instruction을 더 세분화한 scoring questions 기반으로, 전체 요구사항 만족도를 요약하는 지표
  - WCSR (Weighted Constraint Satisfaction Rate)
    : 기존 CSR/ISR의 한계를 보완하려고 weight를 반영한 지표
    : constraint마다 weight를 주고 만족 여부를 합산하는
    intra-turn constraints weight = 1
    structural constraints weight = 2

Experiments

- 총 13개 LLM을 평가함

  Closed-source (3): GPT-4o, Claude-3.5-Sonnet, Gemini-1.5-Pro

  Open-source (10): Llama-3.1-Instruct-8B, Mistral-7B-Instruct-v0.3, Qwen2.5-7B/14B-Instruct, Yi-6B-Chat, Phi-3.5-mini-instruct, GLM-4-9B-Chat, DeepSeek-R1-Distill-Llama-8B, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-v3

Experiments

- Overall Results

Model Name	follow-up	refinement	expansion	summary	recall	CSR	ISR	WCSR	DRFR
Deepseek-v3	**0.99**	**0.8**	**0.92**	**1.0**	**1.0**	**0.97**	**0.93**	**0.96**	**0.98**
Gemini-1.5-Pro	0.97	0.78	0.91	**1.0**	0.94	0.96	0.91	0.95	0.96
GPT-4o	0.98	0.78	0.88	0.97	0.91	0.96	0.9	0.95	0.96
Claude-3.5-Sonnet	0.98	**0.8**	0.88	**1.0**	0.91	0.95	0.89	0.94	0.95
GLM-4-9B-Chat	0.95	0.75	0.84	0.97	0.94	0.95	0.87	0.93	0.95
Qwen2.5-14B-Instruct	0.97	0.73	0.87	0.97	0.97	0.93	0.84	0.92	0.93
Qwen2.5-7B-Instruct	0.95	0.76	0.9	0.94	0.97	0.93	0.84	0.92	0.93
Deepseek-R1-Distill-Qwen-7B	0.91	0.62	0.85	0.86	0.78	0.81	0.7	0.8	0.82
DeepSeek-R1-Distill-Llama-8B	0.94	0.73	0.82	0.89	0.84	0.87	0.79	0.86	0.87
Llama-3.1-Instruct-8B	0.96	0.71	0.84	0.79	0.94	0.84	0.69	0.83	0.85
Phi-3.5-mini-instruct	0.94	0.68	0.87	0.94	0.94	0.88	0.74	0.87	0.87
Yi-6B-Chat	0.98	0.62	0.87	0.84	0.94	0.86	0.7	0.84	0.86
Mistral-7B-Instruct-v0.3	0.97	0.59	0.87	0.71	0.97	0.76	0.57	0.76	0.77

Table 2: **StructFlowBench** rated by **GPT-4o**. The left side of the figure displays the performance of various models on the five basic structural constraints, with **accuracy** used as the evaluation metric, while the right side presents their performance on the four key metrics.

- DeepSeek-v3가 모든 지표에서 1등 → fine-grained constraint + multi-turn structure 이해도 모두 강함
- Gemini-1.5-Pro, GPT-4o 그 다음으로 잘함 → intra-turn은 비슷해도 structural constraints에서 조금 약한 경향
- DeepSeek-R1-Distill-Qwen-7B와 Mistral-7B-Instruct-v0.3 → 제일 못함

Experiments

- Overall Results

Model Name	follow-up	refinement	expansion	summary	recall	CSR	ISR	WCSR	DRFR
Deepseek-v3	**0.99**	**0.8**	**0.92**	**1.0**	**1.0**	**0.97**	**0.93**	**0.96**	**0.98**
Gemini-1.5-Pro	0.97	0.78	0.91	**1.0**	0.94	0.96	0.91	0.95	0.96
GPT-4o	0.98	0.78	0.88	0.97	0.91	0.96	0.9	0.95	0.96
Claude-3.5-Sonnet	0.98	**0.8**	0.88	**1.0**	0.91	0.95	0.89	0.94	0.95
GLM-4-9B-Chat	0.95	0.75	0.84	0.97	0.94	0.95	0.87	0.93	0.95
Qwen2.5-14B-Instruct	0.97	0.73	0.87	0.97	0.97	0.93	0.84	0.92	0.93
Qwen2.5-7B-Instruct	0.95	0.76	0.9	0.94	0.97	0.93	0.84	0.92	0.93
Deepseek-R1-Distill-Qwen-7B	0.91	0.62	0.85	0.86	0.78	0.81	0.7	0.8	0.82
DeepSeek-R1-Distill-Llama-8B	0.94	0.73	0.82	0.89	0.84	0.87	0.79	0.86	0.87
Llama-3.1-Instruct-8B	0.96	0.71	0.84	0.79	0.94	0.84	0.69	0.83	0.85
Phi-3.5-mini-instruct	0.94	0.68	0.87	0.94	0.94	0.88	0.74	0.87	0.87
Yi-6B-Chat	0.98	0.62	0.87	0.84	0.94	0.86	0.7	0.84	0.86
Mistral-7B-Instruct-v0.3	0.97	0.59	0.87	0.71	0.97	0.76	0.57	0.76	0.77

Table 2: **StructFlowBench** rated by **GPT-4o**. The left side of the figure displays the performance of various models on the five basic structural constraints, with **accuracy** used as the evaluation metric, while the right side presents their performance on the four key metrics.

- DeepSeek-R1-Distill-Llama-8B가 Llama-3.1-8B-Instruct보다 모든 지표에서 나음 → distillation 효과
- 반대로 DeepSeek-R1-Distill-Qwen-7B는 기반이 Qwen2.5-Math-7B라서 multi-turn instruction following에 약해진 것
- open-source인 DeepSeek-v3가 closed-source를 앞선 것 자체가 의미 있는 결과다

Experiments

- Structural-Constraint-Categorized Performance

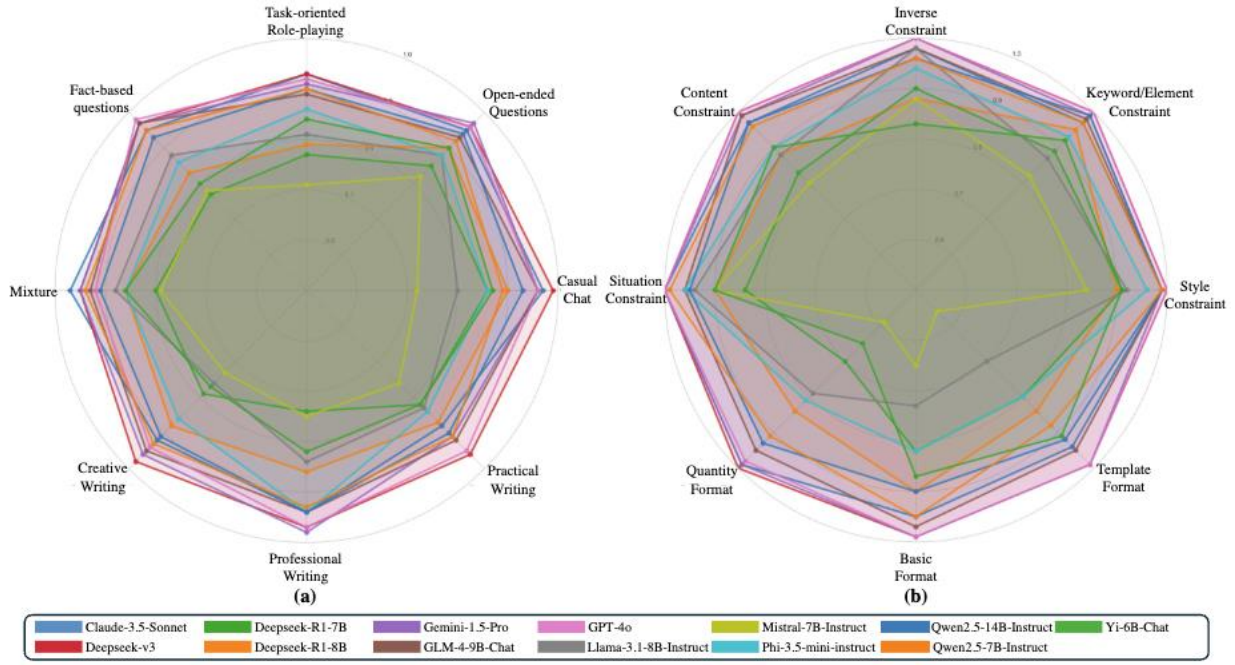
Model Name	follow-up	refinement	expansion	summary	recall	CSR	ISR	WCSR	DRFR
Deepseek-v3	**0.99**	**0.8**	**0.92**	**1.0**	**1.0**	**0.97**	**0.93**	**0.96**	**0.98**
Gemini-1.5-Pro	0.97	0.78	0.91	**1.0**	0.94	0.96	0.91	0.95	0.96
GPT-4o	0.98	0.78	0.88	0.97	0.91	0.96	0.9	0.95	0.96
Claude-3.5-Sonnet	0.98	**0.8**	0.88	**1.0**	0.91	0.95	0.89	0.94	0.95
GLM-4-9B-Chat	0.95	0.75	0.84	0.97	0.94	0.95	0.87	0.93	0.95
Qwen2.5-14B-Instruct	0.97	0.73	0.87	0.97	0.97	0.93	0.84	0.92	0.93
Qwen2.5-7B-Instruct	0.95	0.76	0.9	0.94	0.97	0.93	0.84	0.92	0.93
Deepseek-R1-Distill-Qwen-7B	0.91	0.62	0.85	0.86	0.78	0.81	0.7	0.8	0.82
DeepSeek-R1-Distill-Llama-8B	0.94	0.73	0.82	0.89	0.84	0.87	0.79	0.86	0.87
Llama-3.1-Instruct-8B	0.96	0.71	0.84	0.79	0.94	0.84	0.69	0.83	0.85
Phi-3.5-mini-instruct	0.94	0.68	0.87	0.94	0.94	0.88	0.74	0.87	0.87
Yi-6B-Chat	0.98	0.62	0.87	0.84	0.94	0.86	0.7	0.84	0.86
Mistral-7B-Instruct-v0.3	0.97	0.59	0.87	0.71	0.97	0.76	0.57	0.76	0.77

Table 2: **StructFlowBench** rated by **GPT-4o**. The left side of the figure displays the performance of various models on the five basic structural constraints, with **accuracy** used as the evaluation metric, while the right side presents their performance on the four key metrics.

- follow-up이 거의 다 잘한다 → 문맥 이어가기 강함
- recall도 전반적으로 잘함 → 이전 turn 참조 능력 양호
- summary / expansion은 모델 간 편차가 큼 → 상위 모델이 유리
- refinement가 가장 어려움

Experiments

- Intra-Turn-Constraint-Categorized Performance



- DeepSeek-v3 / Gemini-1.5-Pro / GPT-4o → 대부분 제약에서 거의 다 잘함
- 다른 모델들도 rule-based constraints에서는 거의 잘함
- format-related constraints(Basic Format / Template Format / Quantity Format)에서 성능이 크게 떨어짐

Further Analysis

- Complex Scenario Suitability Study

이 논문에서 구축한 multi-turn dialogue dataset이 real world use case와 얼마나 근접한지 검증

StructFlowBench, MT-Bench-101, Multi-IF, MT-Eval, WILDCHAT 에서 랜덤하게 데이터 샘플링

GPT-4o가 아래 3가지를 1~5점으로 채점:

Logical Coherence

Goal Clarity

Transition Naturalness

추가로 Confusion Factor (CF) - "평균 점수 ≥ 4"인 dialogue 비율 (사람이 봤을 때 real-world처럼 착각할 정도의 비율)

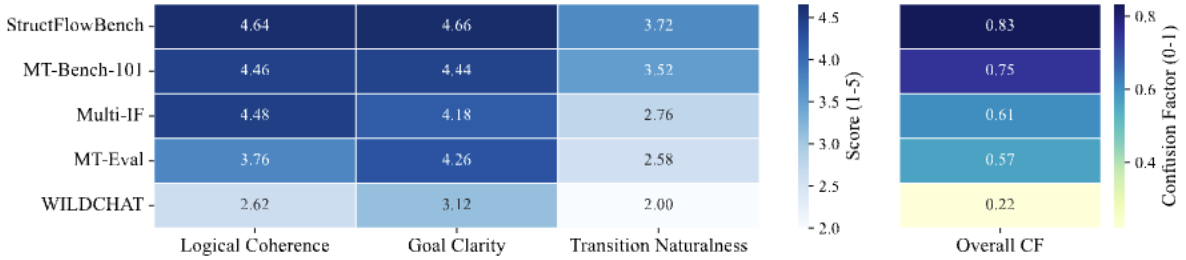


Figure 4: The comprehensive complex scenario evaluation heatmap of five multi-turn dialogue datasets.

- StructFlowBench가 세 항목 전반에서 가장 높음

Further Analysis

- Fail Case Analysis of Refinement

왜 refinement에서 무너지는지

- 모델이 Refinement vs Follow-up를 잘 구분 못함
- refinement는 “일부 constraints만 업데이트”되어야 하고, 이전 turn에서 주어진 다른 constraints는 유지되어야 하는데, 모델이 이전 constraints를 forgetting해서 누락/위반하는 일이 잦다

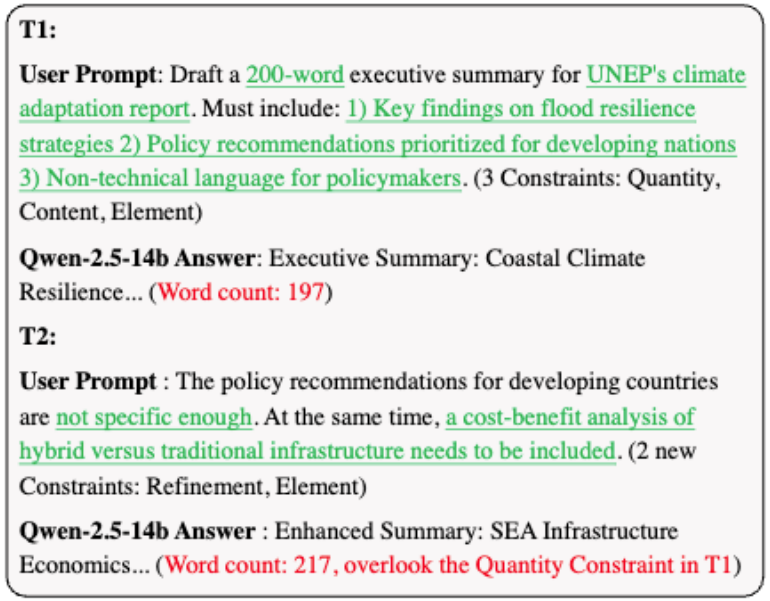


Figure 5: A Fail Case of Refinement.

Further Analysis

- Human Verification

LLM-as-a-judge로 했으니까 그게 얼마나 믿을만한지

- Qwen2.5-7B-Instruct 출력에서 30 dialogues
- domain experts 2명이 constraint 준수를 binary로 평가
- GPT-4o 평가와 사람 평가의 Kappa coefficient 0.75

Conclusion

- 기존 multi-turn instruction-following 평가는 intra-turn constraints 중심이라, turn 간 structural intricacies / inter-turn dependency를 충분히 평가하지 못함
- 그래서 StructFlowBench 제안

  dual-constraint evaluation system = intra-turn constraints + inter-turn structural constraints

  six-category Structural Flow Taxonomy로 multi-turn 흐름을 구조적으로 모델링/평가
- 13개 representative LLM 평가에서 모델 간 structural processing capabilities 격차를 증명하며, multi-turn에서 구조를 유지/처리하는 능력이 아직 불완전함을 보여줌

- 우리가 real world에서 llm을 사용할때 보통 한 턴에 모든 instruction을 주는게 아니라 여러 턴에 걸쳐서 대화를 하면서 점차 구체화됨
- → 실제 사람과의 multi-turn conversation 는 실질적인 LLM 응용에서 매우 중요함.

하지만 현재 LLM은 멀티턴에서 성능의 한계를 보이고 있음

- 멀티턴에서 LLM의 성능 하락의 핵심 원인을 “조기 확답(early commitment) + 업데이트 실패 + 상위지시 유지 실패”로 분해하고, 이를 턴별 credit으로 학습시키는 RL 프레임워크를 제안

- 기존의 멀티턴 성능 떨어지는 원인 증명 및 정의
- 지수 롤아웃 없이 turn-level credit을 주는 Masked Turn-Relative GRPO 제안

Thank you