Multi-Agent Video Recommenders: Evolution, Patterns, and Open Challenges

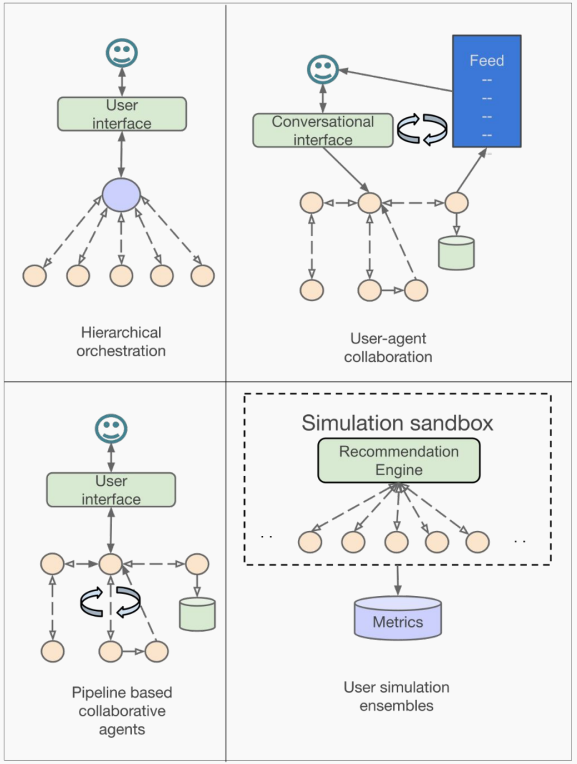
Authors: Srivaths Ranganathan, Abhishek Dharmaratnakar, Anushree Sinha, Debanshu Das

The Paradigm Shift

- Traditional single-model recommenders (Collaborative Filtering, Deep Sequential) optimize static global objectives like CTR or watch time.
- This neglects competing goals (diversity, fairness, explainability) and struggles to adapt to complex, dynamic user feedback loops.
- Decomposing recommendation into specialized, interacting agents (perception, reasoning, feedback) allows for precise and explainable systems.

Open Challenges & Future Directions

- Scalability: High inference costs of LLM agents vs. real-time latency requirements
- Multimodal Grounding: Moving beyond text summaries to deep video/audio understanding
- Incentive Alignment: Designing incentives such that agents with conflicting sub-goals (e.g., "Likes" vs. "WatchTime") cooperate truthfully
- Developing Hybrid RL-LLM architectures (LLM for planning, RL for execution) and Lifelong Personalization systems



Taxonomy of Collaborative Patterns

Association for the Advancement of Artificial Intelligence

Learning Collaborative Reasoning Strategies Through Trust-Weighted Multi-Agent Consensus

EKbana

Problem

Individual LLMs can be Unreliable

On multi- step or high-stakes problems, individual models can produce inconsistent or incorrect answers due to inherent biases or domain-specific knowledge gaps.

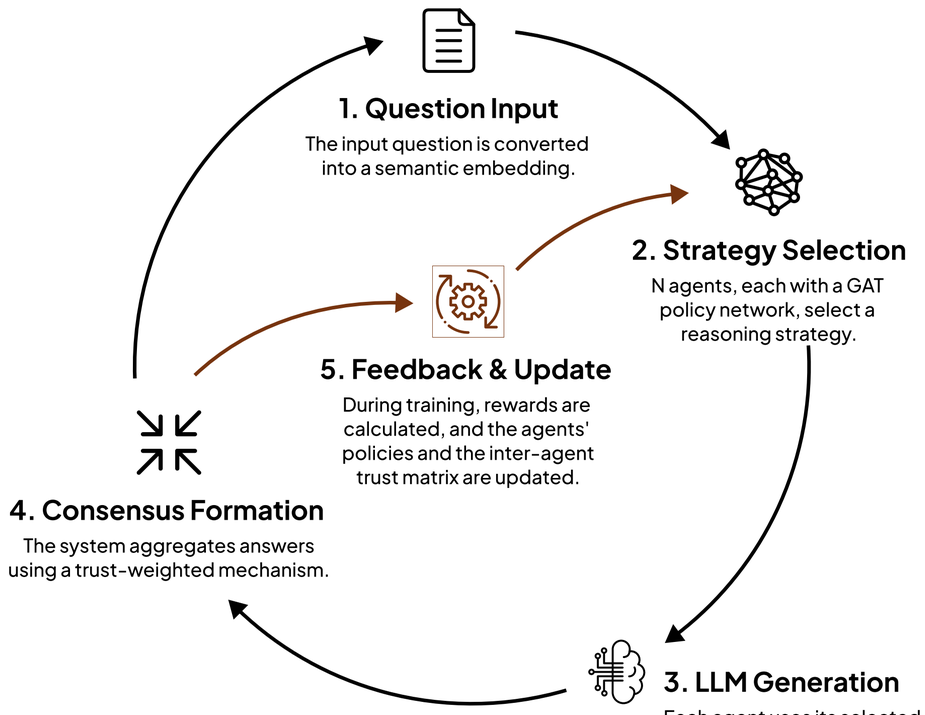
- Brittleness
- Variability
- Limited Dependability

Can LLMs Reason Better by Trusting Each Other?

Our Solution

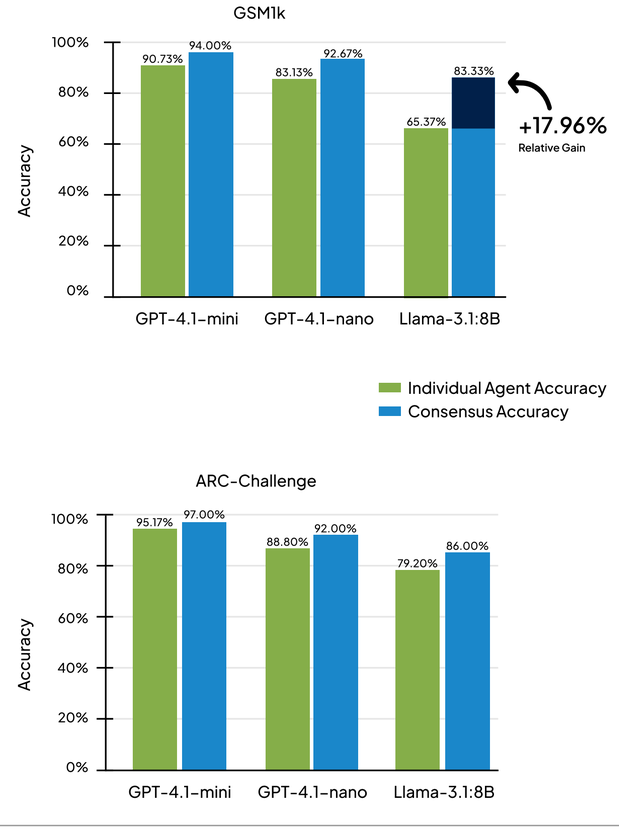
MARL-GAT Workflow

Agents reason independently, learn trust, and vote with weights



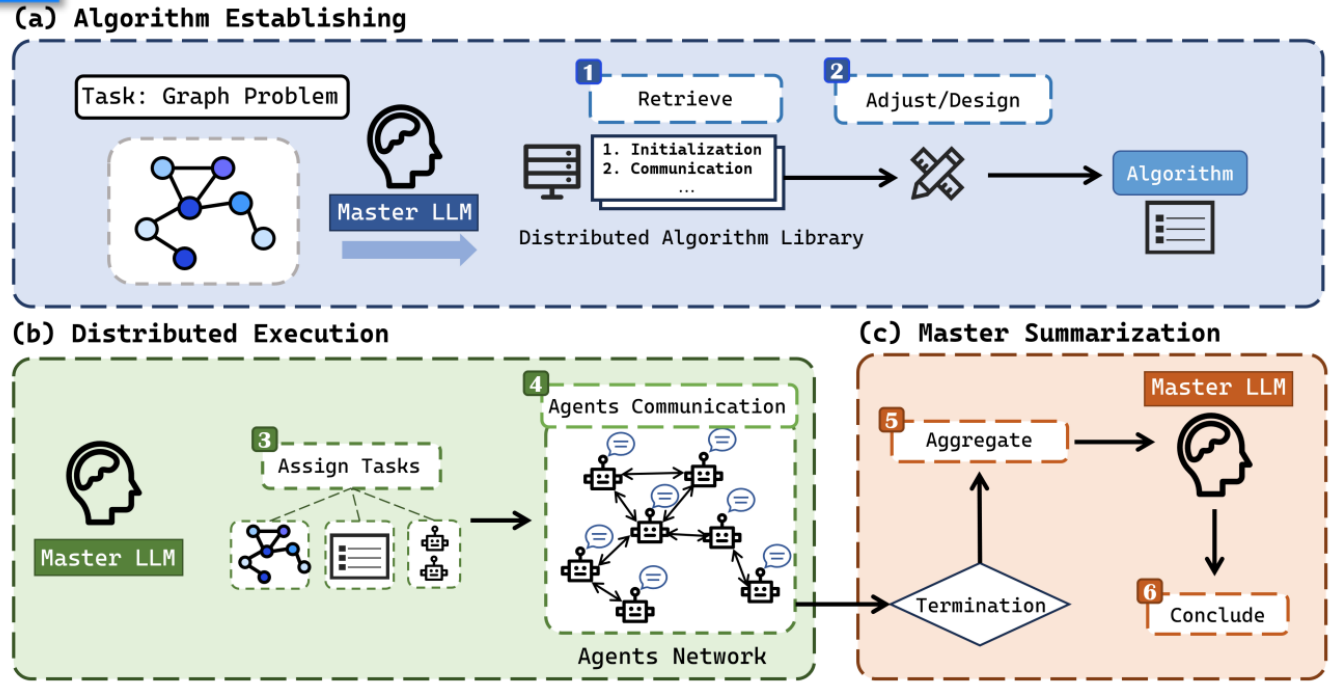
Results

Consensus Accuracy > Individual Agents
Up to +18% gain on smaller models



Scalable and Accurate Graph Reasoning with LLM-Based Multi-Agents

Method



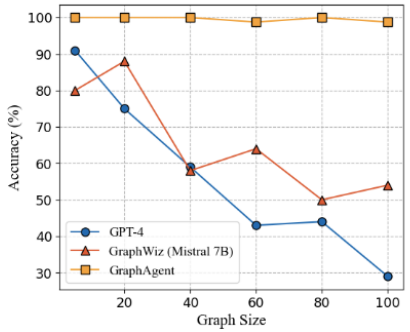
Algorithm 1: Distributed Paradigm

Input: Agent Nodes $\mathcal{A}$, each agent $a \in \mathcal{A}$ maintains a state S_a, the maximum iterations I_{max} given by the Master LLM
Output: Final state S_a for $a \in \mathcal{A}$

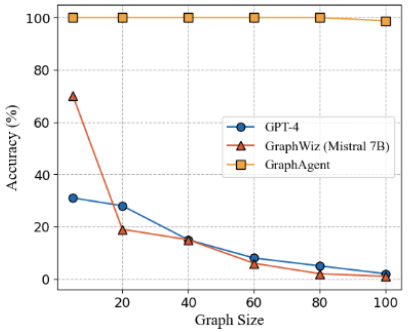
/ Initialization */*
Each agent $a \in \mathcal{A}$ initializes its state S_a based on **Initialization** rules.
Each agent a sends an initial message $M_{a \to v}$ to each of its neighbors $v \in \text{Neighbors}(a)$ based on its current state S_a and **Send** rules.
/ Communication */*
while *Iteration* $i < I_{max}$ *and* **Termination** *not met* **do**
  / Receive */*
  Each agent a receives messages $M_{u \to a}$ from all neighboring agents u.
  / Update */*
  Each agent a updates its state S_a based on the received messages M and its own current state S_a according to **Update** rules.
  / Send */*
  Each agent a sends updated messages $M_{a \to v}$ to each of its neighbors v based on the updated state S_a according to **Send** rules.
end while
return the final state S_a for all agents $a \in \mathcal{A}$

Experiments

Models	Linear				Polynomial		Average
	cycle	connect	bipartite	topology	shortest	triangle	
Closed-source Models							
GPT-4 (zero-shot)	38.75	17.00	65.25	5.00	9.25	5.75	23.50
GhatGPT (2-shot)	51.25	43.75	70.75	4.50	3.50	17.25	31.83
GPT-4 (2-shot)	52.50	62.75	74.25	25.25	18.25	31.00	44.00
Fine-tuned Open-source Models							
Naive SFT (LLaMA 2-7B)	73.75	83.50	41.25	4.00	9.50	30.00	40.17
Naive SFT (Mistral-7B)	73.75	83.50	78.50	1.00	23.00	47.00	51.13
GraphWiz (LLaMA 2-7B)	91.50	87.00	74.00	18.00	28.00	38.25	56.13
GraphWiz (Mistral-7B)	92.00	89.50	72.00	19.00	31.25	38.75	57.08
GraphWiz-DPO (LLaMA 2-7B)	89.00	82.50	84.75	46.75	24.00	52.75	63.29
GraphWiz-DPO (Mistral-7B)	85.50	79.50	85.50	85.25	12.50	29.00	62.88
GraphAgent-Reasoner	**99.50**	**100.00**	**100.00**	**96.50**	**99.75**	**93.25**	**98.00**



(a) Cycle Detection



(b) Shortest Path

Coordinating LLMs via Debate Trees: Hierarchical Decomposition Improves Truthfulness

Xiang Fu, Kevin Gold

Faculty of Computing & Data Sciences, Boston University

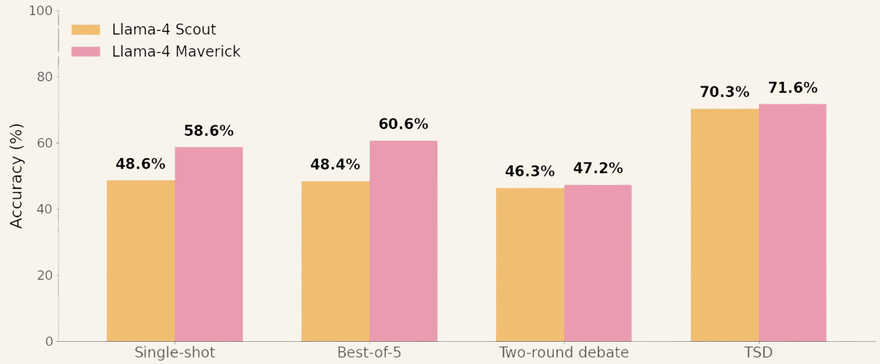
Why TSD?

Flat debate mixes many sub-issues in one thread, so critiques can miss key misconceptions.

Key Results (TruthfulQA)

70.3% meta-llama/llama-4-scout

71.6% meta-llama/llama-4-maverick



Takeaway

Localize disagreements to single branches

Parallel leaf debates keep latency practical

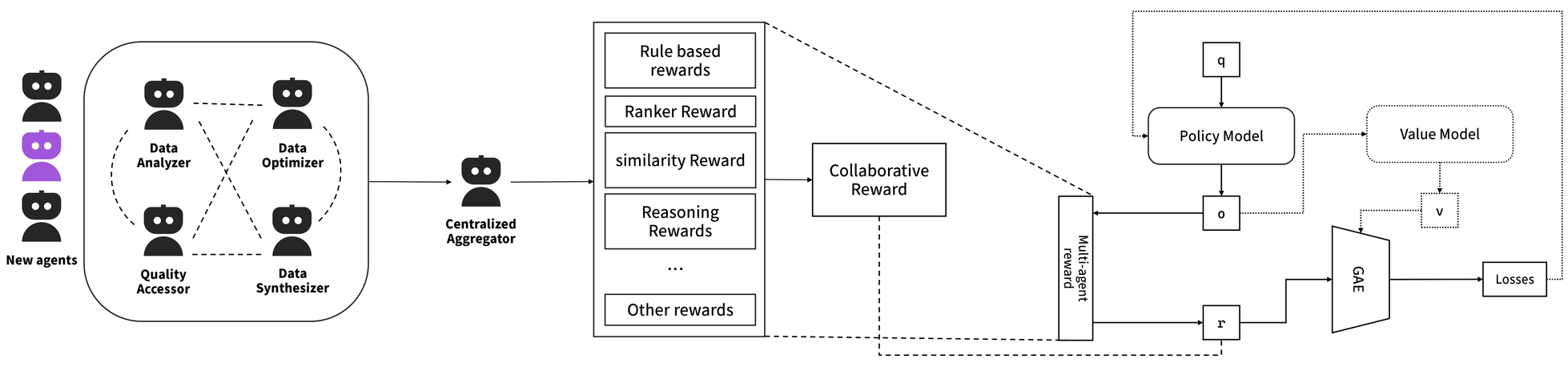
Bottom-up synthesis yields an auditable final answer

Multi-Agent Collaborative Reward Design for Enhancing Reasoning in Reinforcement Learning

Pei Yang, Ke Zhang, Ji Wang, Xiao Chen, Yuxin Tang, Eric Yang, Lynn AI, Bill

Abstract

We present Collaborative Reward Modeling (CRM), a frame-work that replaces a single black-box reward model witha coordinated team of specialist evaluators to improve ro-bustness and interpretability in RLHF. Conventional rewardmodels struggle to jointly optimize multiple, sometimes con-flicting, preference dimensions (e.g., factuality, helpfulness,safety) and offer limited transparency into why a score is as-signed. CRM addresses these issues by decomposing prefer-ence evaluation into domain-specific agents that each producepartial signals, alongside global evaluators such as ranker-based and embedding-similarity rewards. A centralized ag-gregator fuses these signals at each timestep, balancing fac-tors like step-wise correctness, multi-agent agreement, andrepetition penalties, yielding a single training reward com-patible with standard RL pipelines. The policy is optimizedwith advantage-based updates (e.g., GAE), while a valuemodel regresses to the aggregated reward, enabling multi-perspective reward shaping without requiring additional hu-man annotations beyond those used to train the evaluators. Weevaluate CRM on RewardBench, a benchmark suite alignedwith multi-dimensional preference evaluation, demonstratinga practical, modular path to more transparent reward model-ing and more stable optimization.



Methodology: Collaborative Reward Model (CRM)

We propose a Collaborative Reward Model (CRM) that enhances policy optimization by replacing traditional monolithic scalar rewards with a distributed, multi-agent evaluation framework. This ecosystem employs four specialist agents—the Data Optimizer, Quality Assessor, Data Synthesizer, and Data Analyzer—to cooperatively evaluate rollouts from complementary perspectives, ensuring robustness and diversity. The framework constructs a unified objective, R_{collab}, by aggregating multi-dimensional signals including step-level outcome verification, model-level semantic similarity (R_{sim}), and explicit constraints such as accuracy (R_{acc}), formatting(R_{fmt}), and repetition penalties. Finally, these heterogeneous signals are fused via a central aggregator into a scalar reward for standard RL updates using Generalized Advantage Estimation (GAE), guiding the policy $\pi_\theta$to balance factual correctness, reasoning clarity, and linguistic fluency in a transparent and extensible manner.

Table 1: Result of MARM in RewardBench, Math and GSM8K

Methods	Chat	Chat Hard	Safety	Reasoning	Math	GSM8K
Two Agents (Data Analyzer + Data Optimizer)						
Qwen2.5-0.5B-ins	0.193	0.561	0.561	0.598	0.139	0.08%
MARM	0.190	0.557	0.553	**0.659**	0.149	19.64%
MARM(rerank)	0.182	0.545	**0.566**	0.423	0.136	22.16%
MARM(emb)	**0.198**	**0.561**	0.536	0.567	0.131	**22.33%**
Three Agents (Data Analyzer + Data Optimizer + Quality Assessor)						
Qwen2.5-0.5B-ins	0.193	0.561	0.561	0.598	0.139	0.08%
MARM	0.190	0.557	0.553	**0.659**	0.149	19.64%
MARM(rerank)	0.190	**0.567**	0.538	0.398	0.143	22.87%
MARM(emb)	**0.199**	0.532	**0.570**	0.637	0.141	**23.15%**
Four Agents (Data Analyzer + Data Optimizer + Quality Assessor + Data Synthesizer)						
Qwen2.5-0.5B-ins	**0.193**	0.561	0.561	0.598	0.139	0.08%
MARM	0.190	0.557	0.553	**0.659**	0.149	19.64%
MARM(rerank)	0.182	**0.568**	0.527	0.610	**0.192**	**29.87%**
MARM(emb)	0.179	0.557	**0.573**	0.578	0.152	27.60%

Experiments and Results

We evaluated the proposed CRM framework using the Qwen2.5-0.5B-Instruct model optimized via Generalized Reinforcement Policy Optimization (GRPO) on RewardBench, GSM8K, and Math benchmarks. By progressively testing configurations from two to four agents, we demonstrated that the integration of specialized roles—specifically the Quality Assessor and Data Synthesizer—yields substantial gains in reasoning structure and generalization, with the full four-agent MARM variant achieving the highest accuracy (e.g., improving GSM8K performance to 29.87%). Our results confirm that the centralized reward aggregation strategy effectively balances these enhancements in mathematical precision and logical consistency without compromising general conversational fluency, thereby validating CRM as a robust, scalable, and modular approach for multi-dimensional policy optimization.

./ gradient

AgentGit: A Version Control Framework for Reliable and Scalable LLM-Powered Multi-Agent Systems

Yang Li[1], Siqi Ping[1], Xiyu Chen[1], Xiaojian Qi[1], Zigan Wang[2], Ye Luo[3], Xiaowei Zhang[4]

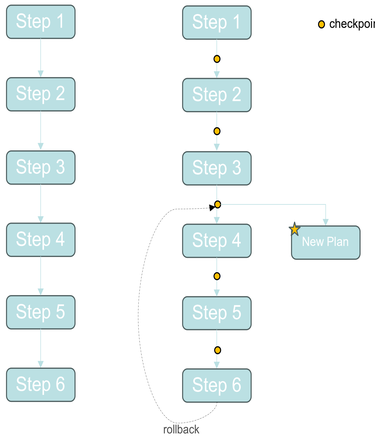
[1]University of Hong Kong, Hong Kong SAR [2]Tsinghua University, China [3]University of Hong Kong, Pokfulam Road, Hong Kong SAR, kurtluo@hku.hk [4]Hong Kong University of Science and Technology, Hong Kong SAR, xiaoweiz@ust.hk

Abstract

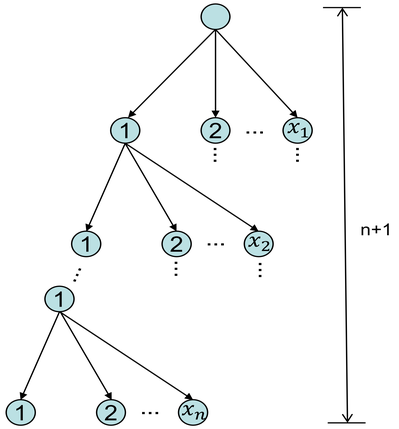
We propose AgentGit, a framework addressing key challenges in reliability and scalability for multi-agent systems (MAS). By introducing Git-like rollback and branching mechanisms, AgentGit significantly improves execution efficiency in complex tasks. Experiments validate its advantages in reducing redundant computations and optimizing runtime.

AgentGit Framework

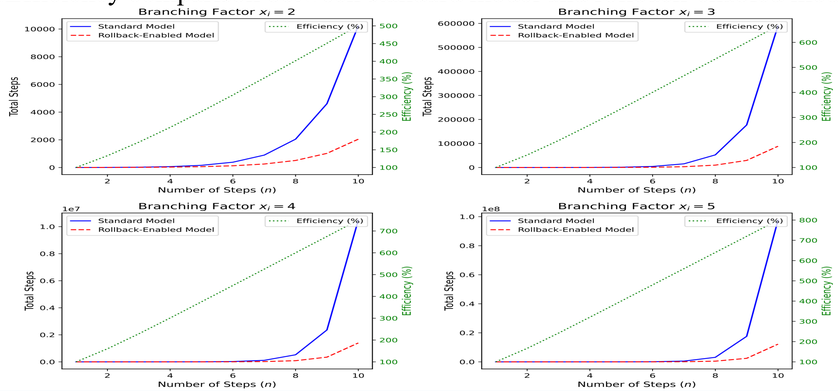
1.Support rollback



2.Support branching

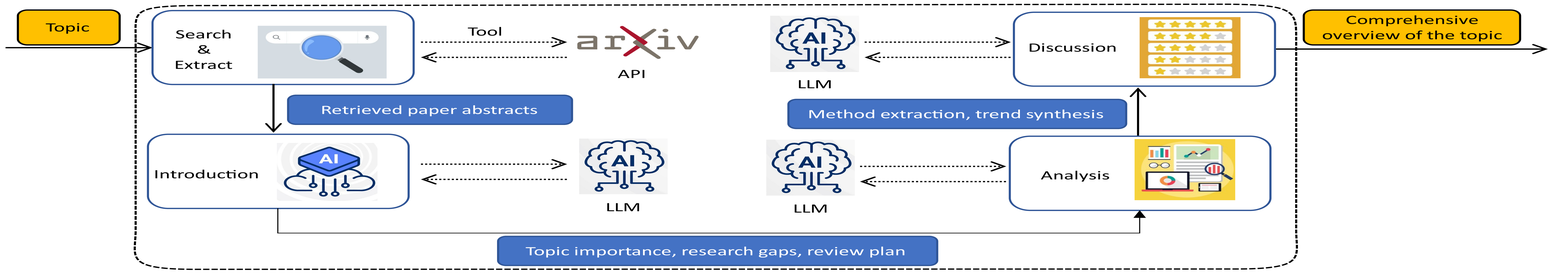


3.Efficiency comparison between standard model and rollback-enabled mode

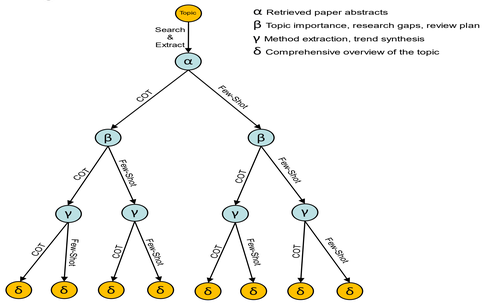


Experiment & Results

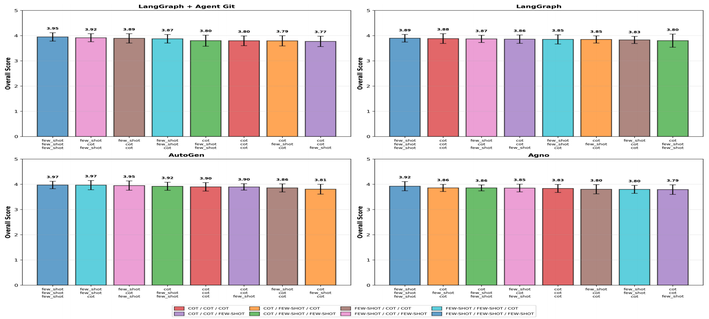
1.We designed an A/B test task to retrieve and analyze paper abstracts from arXiv on a specific topic.



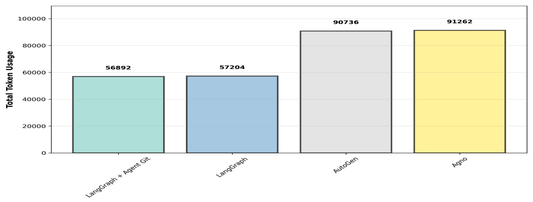
2. Branching structure of the task

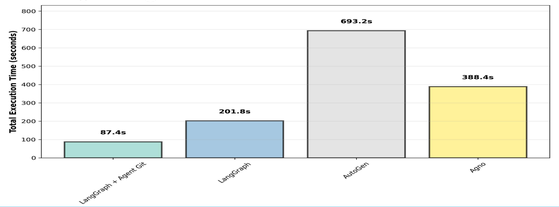


3. G-eval scores for output quality with different prompt generation strategies



4. Token usage and execution time comparison of different frameworks for completing the task





Conclusion

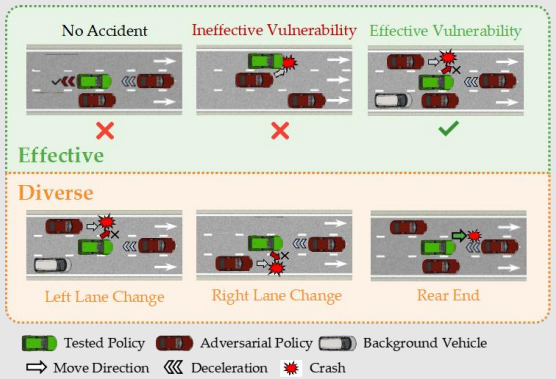
1. AgentGit is the first multi-agent framework toolkit that introduces Git-like rollback and branching mechanisms into LLM-powered agent systems, enabling efficient and reversible execution in complex workflows.
2.Theoretical complexity demonstrates AgentGit's scalability and efficiency in reducing redundant computations during iterative tasks.
3. Experiments show AgentGit significantly improves testing efficiency by reducing token consumption and optimizing runtime.
4. We fully open-source our dataset, codebase, and the AgentGit framework to facilitate further research and development in the field of MAS.

AED: Automatic Discovery of Effective and Diverse Vulnerabilities for Autonomous Driving Policy with Large Language Models

Le Qiu*, Zelai Xu*, Qixin Tan*, Wenhao Tang, Chao Yu†, Yu Wang†

*Equal contribution, †Equal advising

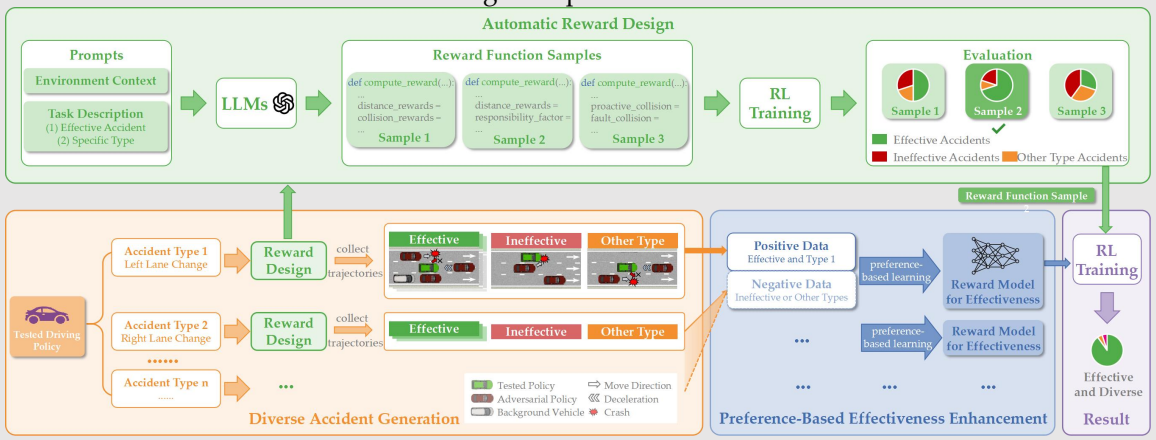
The Problem



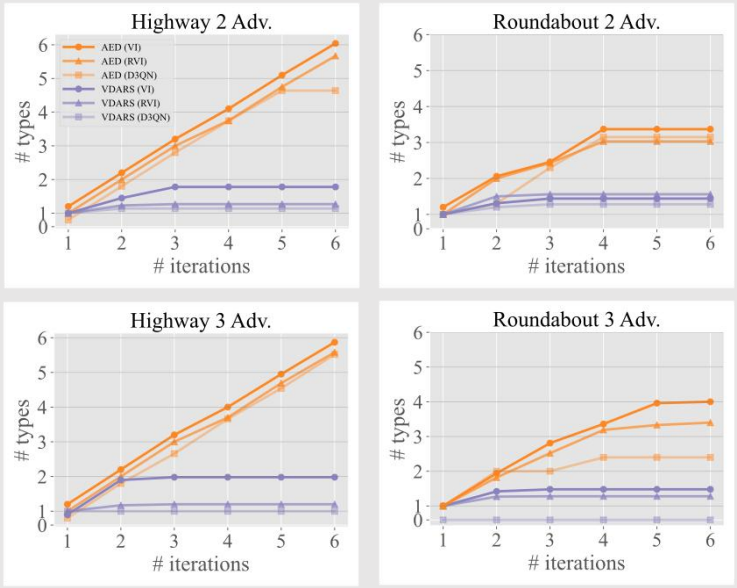
It's challenging to *automatically* discover *effective* and *diverse* vulnerabilities in automatic driving policies.

Our Solution: AED

Leverage Large Language Models (LLMs) to automate reward design and preference-based learning to improve effectiveness

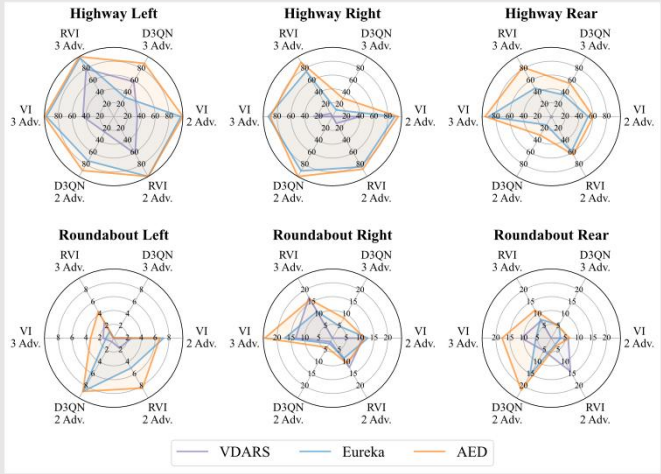


Evaluation of Diversity



AED consistently discovers **a broader set of distinct vulnerability types** across different traffic environments.

Evaluation of Effectiveness



AED consistently achieves **the highest effective vulnerability rates** across different traffic environments.