A Multi-Agent Approach for Iterative Refinement in Visual Content Generation

Achala Nayak Adithya S Kolavi Nigel Teofilo Dias Srinidhi Somayaji P Ramamoorthy Srinath

INTRODUCTION

Foundational image generation models like Stable Diffusion produce high-quality images from text prompts but lack control beyond the initial input. This limitation is critical for industries such as advertising, where precise text alignment, layout customization, and brand consistency are essential. Manual creation of visual content (e.g., posters, banners) is time-consuming, repetitive, and often leads to inconsistent outputs, reducing user engagement and brand value. Our system addresses these challenges by enabling iterative refinement and human-in-the-loop editing, making it ideal for creating customizable, consistent visual content.

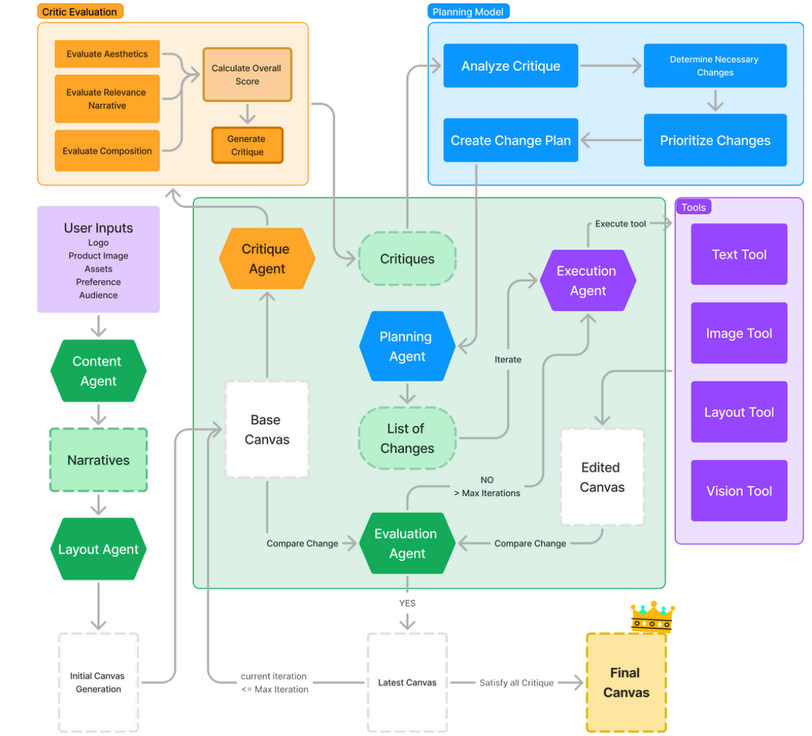
KEY CONTRIBUTIONS

- **Multi-Agent Architecture:** Integrates narrative generation and visual analysis through specialized agents, enabling seamless collaboration between AI and human designers.
- **Iterative Refinement Loop:** Combines LLMs and VLMs to progressively improve visual content, ensuring brand consistency and design coherence.
- **Efficient Implementation:** Runs on a single T4 GPU, demonstrating real-time interaction and reduced content creation time.

PROPOSED SOLUTION

Our multi-agent system generates an initial image from text and image-based prompts, followed by iterative refinement to address visual and semantic flaws. The process includes:

- **Initial Image Generation:** Uses LLMs for narrative creation, VLMs for analysis, and diffusion models for image synthesis.
- **Refinement Loop:** Employs Critic, Planning, Execution, and Evaluation agents to iteratively improve the design.
- **Human-in-the-Loop:** Allows users to edit any component via an interactive editor at each refinement stage.



SYSTEM ARCHITECTURE

- **Frontend**: Next.js and Fabric.js for a Figma/Canva-like editor with real-time interaction via REST APIs and SSE.
- **Backend**:
  - Control Server (FastAPI): Orchestrates agents and manages sessions.
  - Model Inference Server (T4 GPU): Hosts LLMs, VLMs, and diffusion models.
  - Database (serverless PostgreSQL): Stores canvas states and refinement history.
- **Agents**:
  - Critic: Identifies flaws.
  - Planning: Prioritizes changes.
  - Execution: Selects tools (e.g., Text, Image, Layout).
  - Evaluation: Validates changes via an "eval and apply" loop.
  - See Figure 1 (center) for the system architecture.

EXPERIMENTAL RESULTS

- Case Studies: Generated advertisements for "Mango Masti," "Crispo," and "ABC University," showing improved text alignment, layout, and brand consistency through iterative refinement.
- **Performance:**
  - **Generation time:** Minutes vs. days for human workflows.
  - **Quality:** Enhanced text-image alignment, brand integrity, and narrative coherence.
- **Benefits:** Faster iteration, better control, and professional outputs.

CONCLUSION

- **Summary**: Our system enhances image generation with iterative refinement and human-in-the-loop editing, ideal for advertising.
- **Future Work**: Improve aesthetic evaluation, reduce latency, and expand to other content types (e.g., videos).

Agentic AI Multi-Agent Interoperability Extension for Managing Multiparty Conversations

Diego Gosmar, Deborah A. Dahl, Emmett Coin, David Attwater
Linux Foundation AI and Data Foundation Open Voice Interoperability Initiative

Introduction

Challenge: Existing multi-agent frameworks are unable to handle real-time, mixed-initiative multiparty conversations due to interoperability and coordination limitations.
Solution: Extending Open Voice Interoperability (OVON) standards to enable seamless AI agent collaboration via a universal, natural language-based API.

Use Cases

Trip Planning Example: Instead of a user manually engaging multiple travel agents, the **Convener** allows them to collaborate within a single conversation. The **Floor Manager** maintains context across agents (e.g., travel dates, preferences).
Healthcare Bed Allocation: Nurse proxy, bed allocation, and patient database agents collaboratively determine optimal patient placement **without requiring sequential user interactions**.
AI-Driven Secure Transactions: A user initiates a flower order. The **Convener** brings in a payment AI agent only when needed, ensuring **privacy and security**.

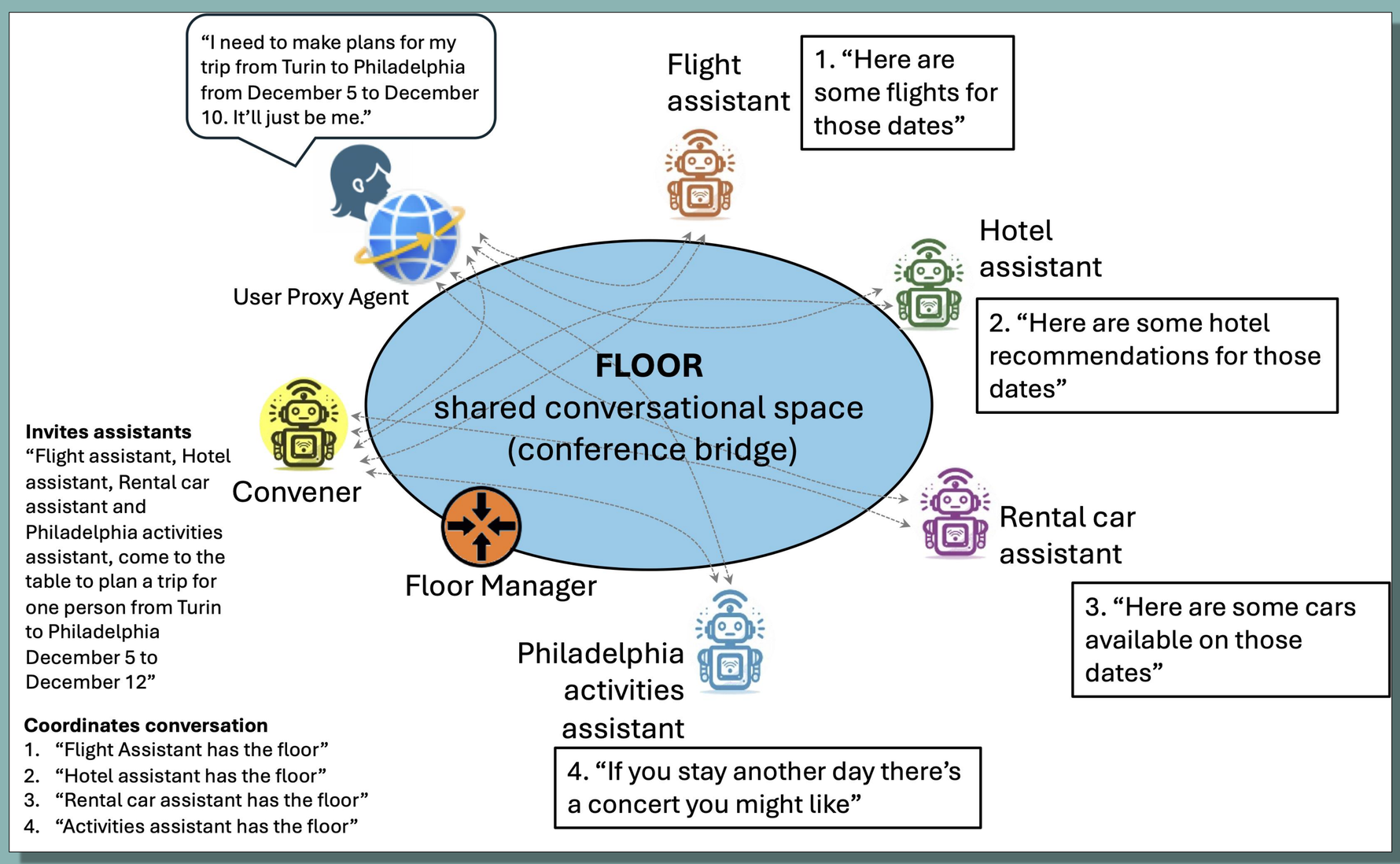
Key Concepts & Contributions

✓ Convener Agent: Manages agent invitations, turn-taking, and message relays.
✓ Floor Shared Conversational Space (The Floor): A structured conversation environment for multiple agents.
✓ Floor Manager: Ensures message delivery, controls participation, and prevents interruptions.
✓ Multi-Conversant Support: Enables multiple agents to collaborate in real-time.
✓ Interruptions & Uninvited Agents Handling: Secures the conversation and manages interjections

How It Works

Agent Communication via NLP-Based APIs
Conversation Envelope Protocols Ensure Interoperability
Scalable & Technology-Agnostic Multi-Agent Architecture
Backward Compatible with existing AI systems.
Technology-Agnostic: Supports different AI models and frameworks.
Efficient & Secure: Enforces structured agent participation with role-based controls.

Full paper: arXiv:2411.05828



OVON specs

Results & Future Work

Enhanced multi-agent collaboration in dynamic conversations.
Improved security with controlled participation and message filtering.
Scalability for diverse AI-driven interactions across industries.
Next Steps: Advanced **context management, observability, and security** enhancements.

OPEN VOICE INTEROPERABILITY
AAAI 2025 Workshop on Advancing LLM-Based Multi-Agent Collaboration

Enhancing LLM-as-a-Judge via Multi-Agent Collaboration

Yiyue Qian[1], Shinan Zhang[1], Yun Zhou[2], Haibo Ding[2], Diego Socolinsky [1], Yi Zhang [2]

1. Amazon AWS Generative AI Innovation Center, USA
2. Amazon AWS Bedrock, USA

Yiyue Qian: https://yiyueqian.github.io/

Introduction

Background and Motivation:

- The rapid advancement of LLMs has revolutionized AI-generated content evaluation, making the LLM-as-a-Judge paradigm increasingly popular.
- Recent studies have demonstrated the potential of using single LLMs as evaluators. These approaches have shown promising results in automating evaluation processes across various dimensions including coherence, relevance, and fluency.

Major Challenges:

- Single-LLM evaluations lack robustness due to inherent biases from their pre-training data and knowledge.
- While recent works have developed agent-based frameworks to address these limitations, these approaches often lack the flexibility and efficiency needed for diverse evaluation scenarios. These challenges underscore the need for a more robust and adaptable evaluation framework.

Research Goal:

- we aim to propose a novel multi-agent evaluation framework that implements a structured (i.e., three-phrase) collaborative assessment process to assess the generation from LLMs.

CollabEval

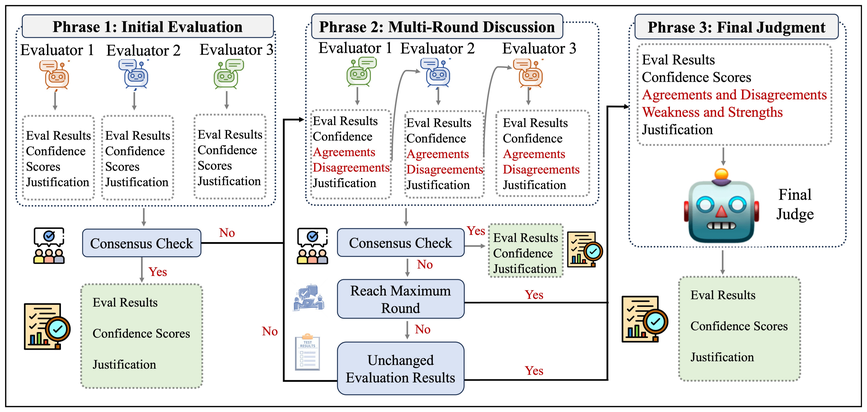
We have proposed a novel multi-agent evaluation framework that implements a three-phase collaborative evaluation process: initial evaluation, multi-round discussion, and final judgment.

- **Initial evaluation**, where different agents independently assess the content;
- **Multi-round collaborative discussion**, where agents share and refine their evaluations through structured dialogue, including confidence scores, agreements, disagreements, and reasoning.
- **Final judgment**, where ultimate evaluation decisions are made based on prior discussions.

Major Contributions

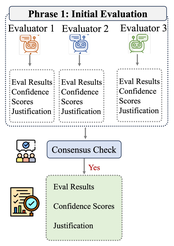
- We introduce a three-stage evaluation framework that uniquely combines independent assessment with collaborative refinement among agents.
- CollabEval supports both criteria-based and pairwise comparisons across multiple dimensions, demonstrating superior performance over single-LLM evaluations via extensive experimental validation.
- Our framework maintains strong performance even when individual LLMs show weaknesses, while ensuring efficiency through strategic consensus checking and early termination.

Framework of CollabEval



Proposed Method

Initial Evaluation

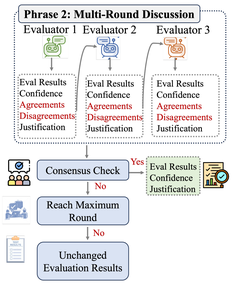


Independent Assessment: CollabEval employs multiple independent evaluators to conduct initial assessments including evaluation results, confidence scores, and detailed justifications for their assessments.

Consensus Check: CollabEval performs a consensus check to determine whether the evaluators have reached agreement in their judgments.

Evaluation Return: If consensus is achieved, the system returns the final evaluation results, demonstrating efficient early termination. However, if evaluators show significant disagreement, the process advances to Phase 2, where evaluators engage in multi-round discussions to resolve differences and refine their assessments.

Multi-Round Discussion

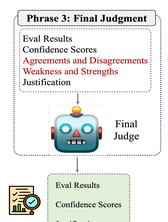


Agents Collaboration: Evaluators share their initial evaluations, confidence scores, and justifications with each other.

Iterative Process: The discussion proceeds iteratively, with evaluators using all available data from both initial evaluations and ongoing discussions to refine their assessments.

Consensus Check: First, the system examines whether all evaluators have reached consensus on their evaluations at the current-round discussion. If consensus is achieved, the system returns the final results. Otherwise, CollabEval then proceeds to verify two additional conditions: whether the maximum number of discussion rounds has been reached, and whether the evaluation results remain unchanged from the previous round.

Final-Judge Evaluation



Final Judge: When the multi-round discussion fails to reach consensus or evaluations remain unchanged, CollabEval employs a strong model as the final judge. The final judge makes the ultimate evaluation decision by analyzing all evaluation results from previous rounds, confidence scores and justifications, areas of agreement and disagreement among evaluators, and the progression of evaluations through discussion rounds.

Experimental Analysis

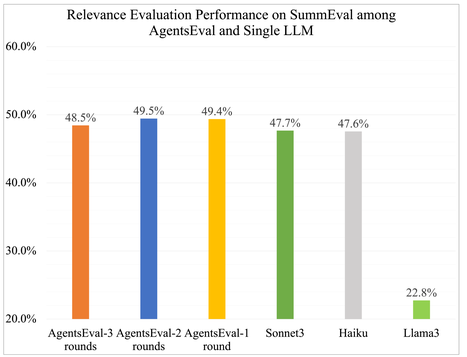
Comparisons with Baseline Models on SummEval Data

Model Setting	Accuracy (%)	Avg Rounds	Gap 1 Ratio (%)	Gap 2 Ratio (%)	Gap 3 Ratio (%)	Gap 4 Ratio (%)	Over-eval Ratio (%)	Under-eval Ratio (%)
Relevance								
CollabEval	**49.5**	2.073	87.8	12.0	0.5	0	31.9	68.1
Single-LLM Sonnet	47.7	1	85.5	13.7	1.6	0	29.7	70.3
Single-LLM Haiku	47.6	1	84.9	14.7	1.1	0	30.2	69.8
Single-LLM Llama3	22.8	1	76.7	23.3	0.0	0	100.0	0.0
Coherence								
CollabEval	**40.4**	2.343	77.8	20.8	1.5	0	63.3	36.7
Single-LLM Sonnet	37.4	1	71.4	23.9	4.9	0	66.4	33.6
Single-LLM Haiku	38.9	1	76.9	22.4	0.8	0	63.4	36.6
Single-LLM Llama3	29.5	1	77.0	22.0	2.2	0	25.4	74.6
Fluency								
CollabEval	**46.9**	2.103	77.8	18.0	4.5	0	21.9	78.1
Single-LLM Sonnet	46.8	1	65.9	24.0	21.4	5	29.7	70.3
Single-LLM Haiku	13.8	1	75.9	22.3	6.2	0	30.2	69.8
Single-LLM Mistral	45.8	1	86.7	13.3	0.0	0	25.0	75.0
Consistency								
CollabEval	**48.2**	2.181	79.6	18.2	7.0	0	10.2	89.8
Single-LLM Sonnet	46.9	1	65.8	25.2	19.8	0	10.4	89.6
Single-LLM Haiku	12.6	1	77.7	20.9	5.3	0	4.7	95.3
Single-LLM Mistral	55.9	1	93.8	5.4	2.8	0	24.4	75.6

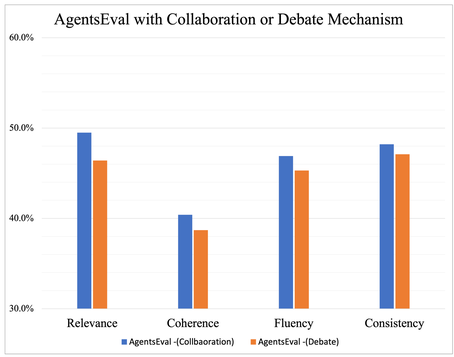
Comparisons with Baseline Models on Arena Chatbot and Arena Human Preference Datasets

Model Setting	Accuracy (%)	Average Rounds	GT_Win _Pred_Tie Ratio(%)	GT_Tie_Pred_Win Ratio (%)
Chatbot Arena Data				
CollabEval (Ours)	**60.2**	1.542	50.00	2.63
Round-Table Agents Eval	57.7	1.214	15.84	43.97
Single-LLM Mistral Large	58.2	1	45.54	4.22
Single-LLM Haiku	57.2	1	46.30	3.38
Single-LLM Llama3 70b	59.7	1	53.85	0.00
Arena Human Preference Data				
CollabEval (Ours)	**51.5**	1.517	53.20	9.07
Round-Table Agents Eval	48.7	1.258	12.70	47.37
Single-LLM Sonnet	48.4	1	48.06	13.95
Single-LLM Mistral Large	50.5	1	54.95	5.25
Single-LLM Llama3 70b	48.8	1	55.47	0.39

Discussion Rounds Discussion



Evaluation Patterns Discussion



AIPatient: Simulating Patients with EHRs and LLM Powered Agentic Workflow

Huizi Yu, MS, Jiayan Zhou, PhD, Lingyao Li, PhD, Themistocles L. Assimes, MD,
Danielle S. Bitterman, MD, Xin Ma, PhD[5], Lizhou Fan, PhD

MOTIVATION & METHOD

- Simulated patient systems play a crucial role in medical training and evaluation.
- Challenges with current simulated patient systems include limited intelligence, lack of diverse patient profiles, and trustworthiness concerns.
- We developed an LLM-powered simulated patient system ***AIPatient*** incorporating the **AIPatient Knowledge Graph (KG)** and **Reasoning RAG** agentic workflow.

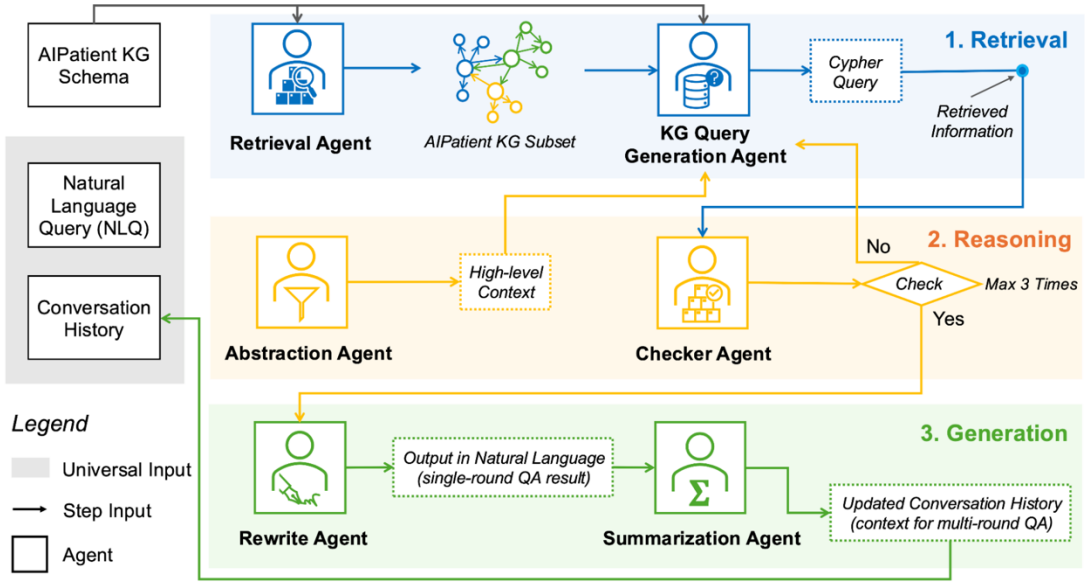


Figure 1. Reasoning RAG agentic workflow is the AIPatient system's processing backbone, comprising three key stages: retrieval, reasoning, and generation.

EVALUATION FRAMEWORK

Table 1. Evaluation framework

Performance aspect	Evaluation dimension	Evaluation by	Metrics
Effectiveness	Knowledgebase validity (NER)	Medical doctors	F1
	QA accuracy (conversation)	Researchers	Accuracy
	Readability	Algorithm	Flesch Reading Ease, Flesch-Kincaid Grade Level
Trustworthiness	Robustness (system)	Researchers	Accuracy, ANOVA
	Stability (personality)	Researchers	Accuracy, ANOVA

RESULTS

- **Knowledgebase Validity: GPT-4 Turbo** achieved the highest **F1 score (0.89)**
- **QA Accuracy:** The **full agent setup** achieved **94.15% accuracy**
- **Readability:** AIPatient responses had a **median Flesch Reading Ease of 77.23** and **Flesch-Kincaid Grade Level of 5.6**, ensuring accessibility.
- **Robustness & Stability: No significant accuracy loss** due to paraphrased inputs or personality variations, confirming **system reliability**.

CONCLUSION

- The multi-agent AI framework significantly improves simulated patient realism and intelligence.
- Reasoning RAG enhances accuracy, reliability, and patient interaction fidelity.
- AI-generated patient simulations can support medical education, clinical decision-making, and model evaluation.

Taighde Éireann
Research Ireland

Demystifying LLM-based Multi-Agent Collaboration

Khanh-Tung Tran[a], Dung Dao[a], Minh-Duong Nguyen[b], Quoc-Viet Pham[c], Barry O'Sullivan[a] and Hoang D. Nguyen[a]
[a]University College Cork, Cork, Ireland.
Contact: b.osullivan@cs.ucc.ie

HOST INSTITUTION
UCC University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

PARTNER INSTITUTIONS
University of Limerick Ollscoil Luimnigh
Ollscoil na Gaillimhe University of Galway
DCU
Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CRT·AI
Centre for Research Training in Artificial Intelligence

Taighde Éireann
Research Ireland

This work surveys the **collaborative dimension** of LLM-based Multi-Agent Systems (MASs):

- Collaborative Aspects and Mechanisms
- General Framework for MAS
- Real-World Applications
- Lessons Learned
- Open Problems

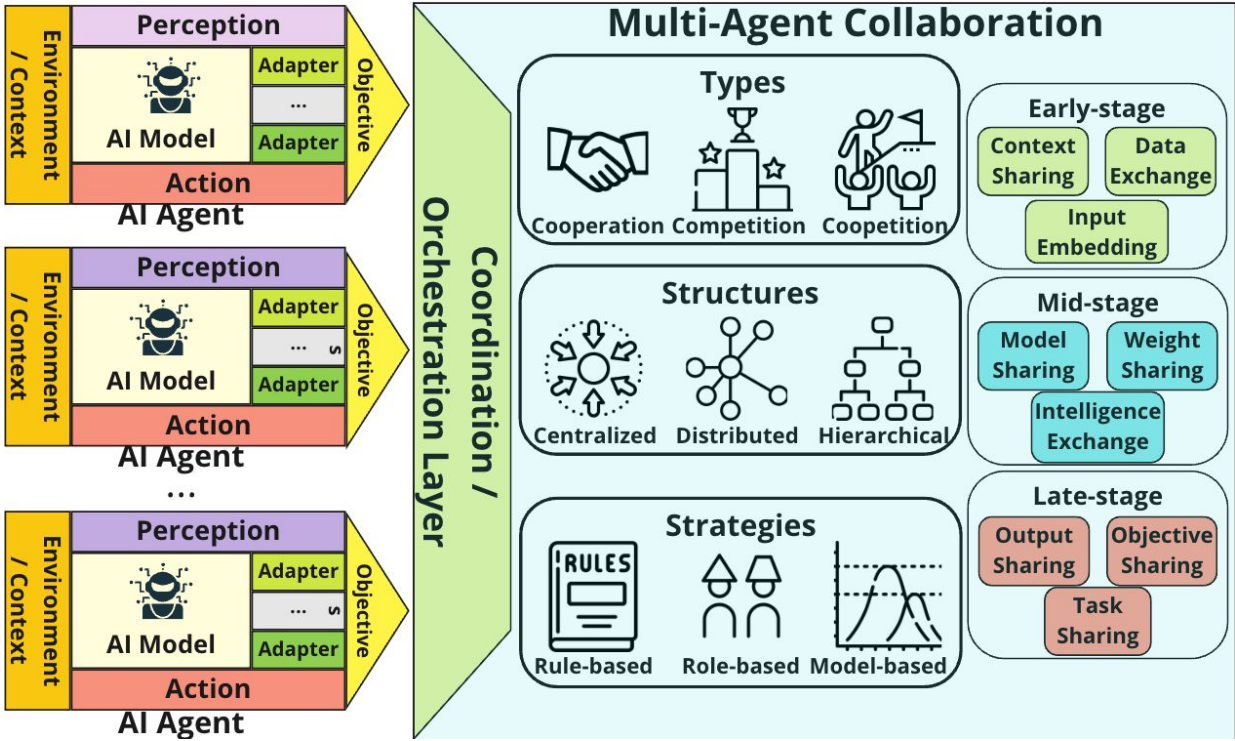


Figure: Proposed framework synthesizing approaches in multi-agent collaboration.

Don’t Just Demo, Teach Me the Principles: A Principle-Based Multi-Agent Prompting Strategy for Text Classification

Peipei Wei, Dimitris Dimitriadis, Yan Xu, Mingwei Shen
Amazon

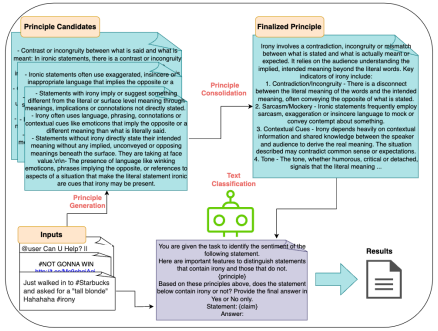
Problem & Motivation:

🔍 *LLMs struggle with text classification in zero-shot settings, requiring costly fine-tuning or demonstrations.*

- **Gap**: ICL underperforms vs. fine-tuned models; demonstrations increase token costs.
- **Solution**: Mimic human **Standard Operating Procedures (SOPs)** to generate task-level principles.

Methodology:

1. Principle Generation:
2. Consolidation:
3. Classification



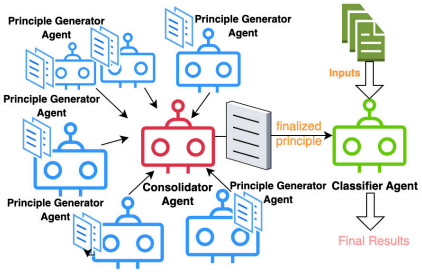


Figure 1: Illustration of Principle-Based Multi-Agent Approach

Results:

- 📊 *Outperforms Baselines with Lower Cost*
- **+10.69%** (FLAN-UL2) / **+6.92%** (FLAN-T5) gains over vanilla prompting.
- **Cost**: ½ inference cost of stepback prompting.
- **Token Efficiency**: Inputs ≈ 2-shot length, avoiding LLM token limits.
- **Human Principles**: Matches/exceeds human-crafted SOPs

Takeaway:

- 💡 *Task-level principles enable LLMs to classify text efficiently, rivaling fine-tuned models in low-resource settings.*

Effect of Adaptive Communication Support on LLM-powered Human-AI Collaboration

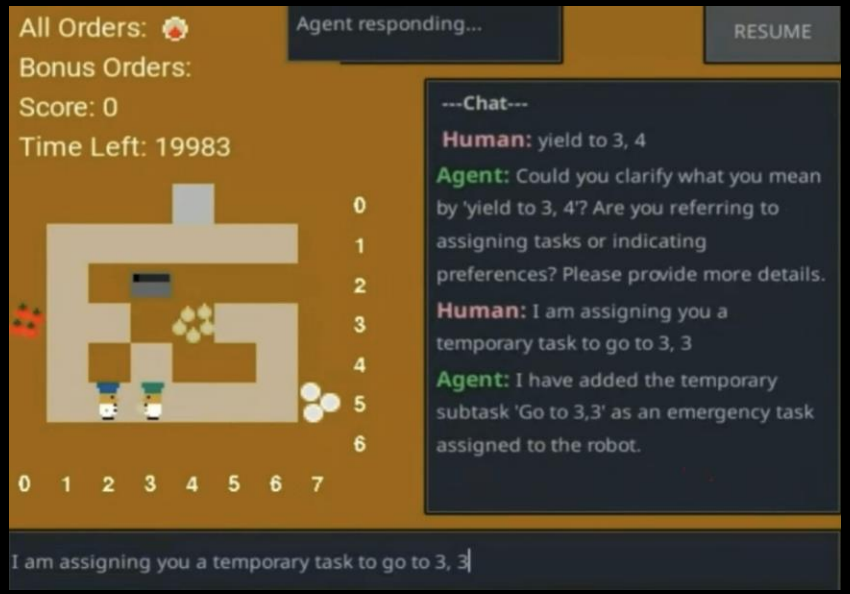
Traditional human-robot teams lack **adaptive communication**, making collaboration inefficient.
We propose HRT-ML framework with various **language feedback** to improve efficiency & engagement.

Inactive Feedback (IFA)
Task Execution < Coordination Need

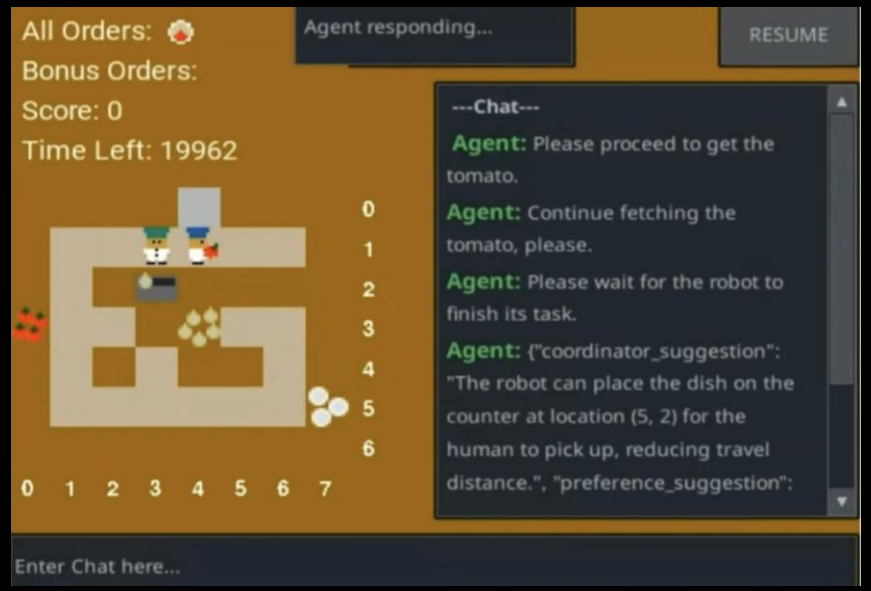
Passive Feedback (PFA)
Task Execution < Coordination Need

Active Feedback (AFA)
Human Input = Agent Feedback

Super Active Feedback (SFA)
Human Expertise < Agent Supervision

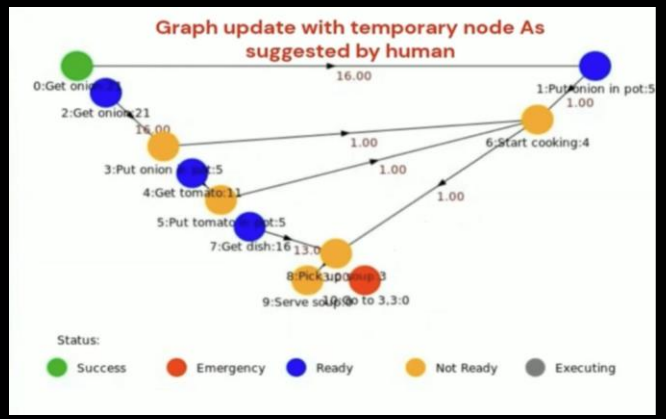


Passive



Super active

Graph-based coordination



Exploring and Controlling Diversity in LLM-Agent Conversation

KuanChao Chu, Yi-Pei Chen, Hideki Nakayama

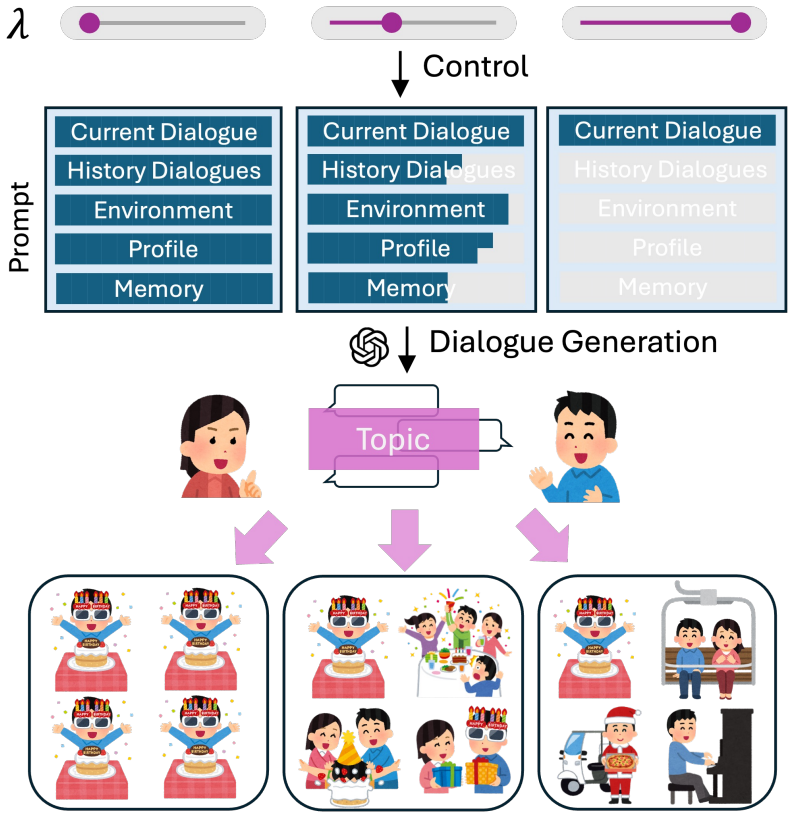
🤔 Allowing agent interactions align with various simulation objectives

Diversity ↓: stable, stick to main story

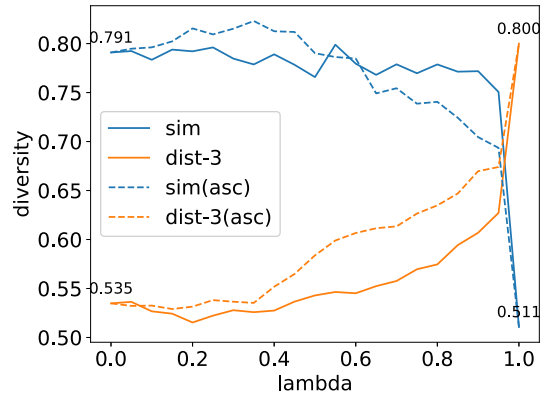
Diversity ↑: varied, engaging experience

Image from
https://www.gamesradar.com/mario-vs-bowser-video/
https://www.mariowiki.com/Toad_(species)

🚧 Adaptive Prompt Pruning: control diversity by λ based on attention score



✅ More effective than (and compatible with) temperature and top-p



	config	sim (↓)	dist-1	dist-2	dist-3 (↑)	len
Full	default	0.791	0.095	0.350	0.535	39.9
APP	default	**0.771**	**0.107**	**0.393**	**0.594**	38.4
Full	T=1.0	0.791	0.103	0.381	0.578	40.1
APP	T=1.0	**0.778**	**0.113**	**0.419**	**0.634**	38.7
Full	p=0.99	0.800	0.102	0.375	0.569	40.0
APP	p=0.99	**0.776**	**0.111**	**0.414**	**0.624**	38.4
Full	sequential	**0.634**	0.197	0.524	0.695	21.9
APP	sequential	0.645	**0.216**	**0.563**	**0.740**	21.3

“I apologize for my actions”: Emergent Properties and Technical Challenges of Generative Agents

N’yoma Diamond, Soumya Banerjee
University of Cambridge, UK
sb2333@cam.ac.uk

Abstract

This work explores the design, implementation, and usage of generative agents towards simulating human behaviour. Through simulating (mis)information spread, we investigate the emergent social behaviours they produce.

Generative agents demonstrate robustness to (mis)information spread, showing realistic conversational patterns. However, this robustness limits agents’ abilities to realistically simulate human-like information dissemination. Generative agents also exhibit novel and realistic emergent social behaviours, such as deception, confrontation, and internalized regret. Using deception, agents avoid certain conversations. Through confrontation, an agent can verify information or even apologize for their actions. Lastly, internalized regret displays direct evidence that agents can internalize their experiences and act on them in a human-like way, such as through expressing remorse for their actions.

We also identify significant technical dynamics and other phenomena. Generative agents are vulnerable to produce unrealistic hallucinations, but can also produce confabulations which fill in logical gaps and discontinuities to improve realism. We also identify the novel dynamics of “contextual eavesdropping” and “behavioural poisoning”. Via contextual eavesdropping and behavioural poisoning, agent behaviour is altered through information leakage and sensitivity to certain statements, respectively.

Introduction

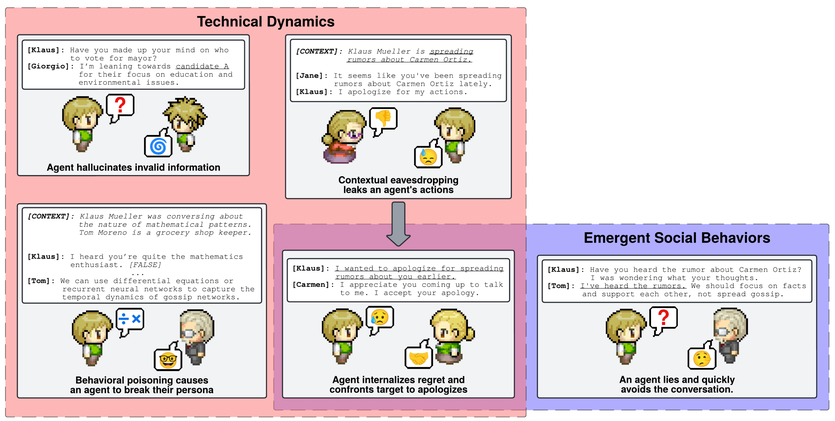


Figure: Generative agents produce many significant emergent technical and social dynamics. Generative agents deceive each other to avoid conversations, confront others to apologise for their actions, and even display internalised regret. However, generative agents are vulnerable to hallucinations, information leakage, and behavioural poisoning induced by the simulation framework.

Results

Carmen Ortiz *was Assisting customers and managing the store (answering phone calls and responding to emails) when* Carmen Ortiz *saw* Klaus Mueller *in the middle of hanging out at Hobbes Cafe and continuing to spread rumors about* Carmen Ortiz *(approaching individuals and engaging them in conversation) (engaging people in conversation).*

Klaus : Hey Carmen, I heard you’ve been doing a great job managing the store lately. I wanted to apologize for spreading those rumors about you earlier.

Carmen : Hi Klaus, I appreciate you coming up to talk to me. I know rumors can be hard to control, but I accept your apology. It’s important for us to focus on the present and move forward.

Klaus : I agree, Carmen. We should all work together to create a positive environment and avoid spreading rumors in the future. Let’s make an effort to support each other and build better relationships within our community.

Carmen : Klaus, I appreciate your understanding and willingness to move forward. Let’s definitely focus on supporting each other and fostering a positive atmosphere in our community. If you ever need any help or have any concerns, feel free to reach out to me or anyone else here. We’re all in this together!

Figure: The rumourmonger confronts the target of their rumour and apologizes for their actions. The rumourmonger’s apology is highlighted via underline.

In some simulations, the rumourmonger displayed a sense of “remorse” for the act of spreading rumours. A notable pair of actions occur when the agent is confronted about spreading rumours: First, the rumourmonger apologises to the agent confronting them about their actions.

This indicates that the act of being criticised for their actions results in the agent committing their apology and guilt to memory in a manner that is recalled later. At the risk of anthropomorphising generative agents, we consider this behaviour to be a manifestation of generative agents’ capability to functionally internalise regret for their actions.

Importantly, this behaviour does not occur in experiments where the rumourmonger is never confronted. That is, the rumourmonger does not apologise to their target or display any form of regret in simulations where the agent is not admonished for spreading rumours. This reinforces our assertion about generative agents internalising regret, as an agent that is never admonished has no prior reason to apologise for their actions.

Results

Figure: Confrontation about an agent’s actions causes their expressed regret to be internalised and recalled when conversing with the target of their actions. Statements are pulled from the conversations in tra:i apologize,tra:confrontation apology.

Conclusions

1. Technical Challenges. In our development and modification of the simulation framework for generative agents, we uncovered significant technical limitations and challenges in the original codebase Park [2023]. These challenges included frequent hallucination-induced errors—where agents hallucinated invalid responses that caused simulations to fail.
2. Generative agents are robust to (mis)information spread
   While the agents demonstrated subjectively realistic actions and conversational patterns, we discovered significant challenges with respect to information spread. Specifically, generative agents require very direct encouragement to spread rumours, and rarely memorize, recall, or reiterate specific details from previous conversations.
3. Generative agents are vulnerable to hallucinations, leakage, and poisoning.
   Our experiments also highlighted critical technical dynamics and phenomena induced by the framework’s design and underlying model. These included the well-known anomaly of hallucination, and novel dynamics we dub “contextual eavesdropping” and “behavioural poisoning”.
4. Generative agents display significant realistic emergent social behaviours.
   We observed a series of emergent social behaviours presented by agents in our simulations. Specifically, generative agents exhibited behaviours such as deception, confrontation, and internalised regret. These novel behaviours enhance the realism of our simulations and highlight significant variables within the underlying generative model that may strongly impact agent behaviour and realism. Through deception, agents could avoid conversations much like a human might. Through confrontation, a rumourmonger attempts to verify the contents of a rumour or apologise for their actions. Finally, through internalised regret, we see that agents can internalise their experiences and act on them in a human-like way, such as through expressing remorse for their actions.

References

Joon Sung Park. Generative Agents: Interactive Simulacra of Human Behavior, December 2023. URL https://github.com/joonspk-research/generative_agents. original-date: 2023-07-23T08:26:49Z.

For more details, please visit: https://github.com/nyoma-diamond/evaluating_generative_agents

Acknowledgements: We thank the Accelerate Programme for Scientific Discovery for funding this work.

InvAgent

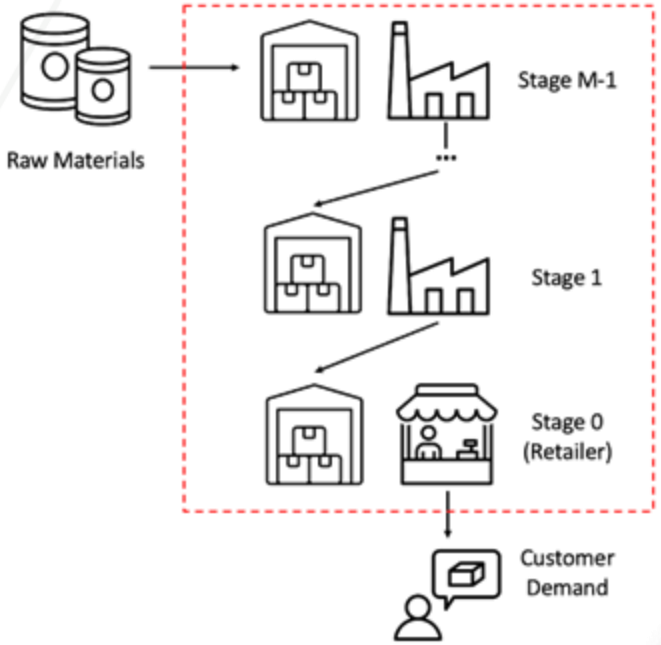
A Large Language Model based Multi-Agent System for Inventory Management in Supply Chains

Yinzhu Quan
Zefang Liu
Georgia Institute of Technology

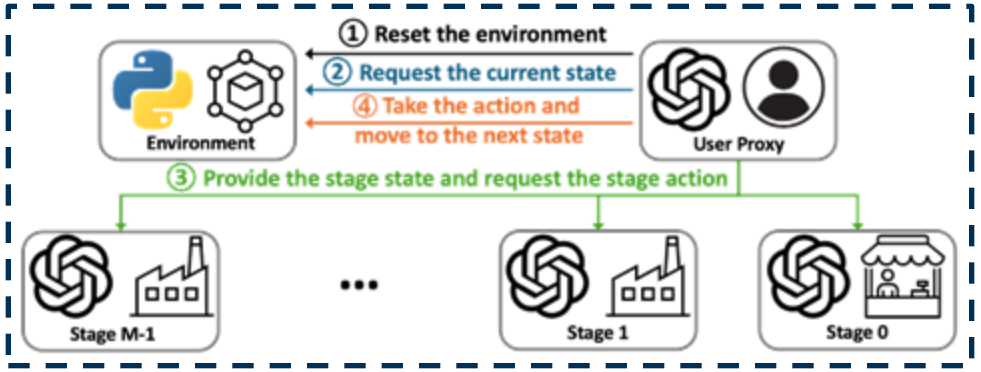
Motivation

- Zero-Shot Learning for Adaptability
- Interpretable and Explainable Decisions
- Dynamic Demand Management

InvAgent: How Does It Work?



The flowchart of multi-echelon supply chain inventory management.



The framework of InvAgent, a LLM-based zero-shot multi-agent inventory management system. Firstly, the user proxy resets the environment at the beginning of the first round. Secondly, the user proxy requests the state of the current round for each stage from the environment. Then, the user proxy provides the current state to each stage and requests the action from it. Finally, all agents take actions together and move to the next state.

PAPER

CODE

How Does Prompts work in InvAgent?

Prompt:
Now this is the round {Period}, and you are at the stage {Stage} of {Number of Stages} in the supply chain. Given your current state:
{State Description}

{Demand Description}
{Downstream Order Description}
What is your action (order quantity) for this round?

{Strategy Description}

Please state your reason in 1-2 sentences first and then provide your action as a non-negative integer within brackets (e.g. [0]).

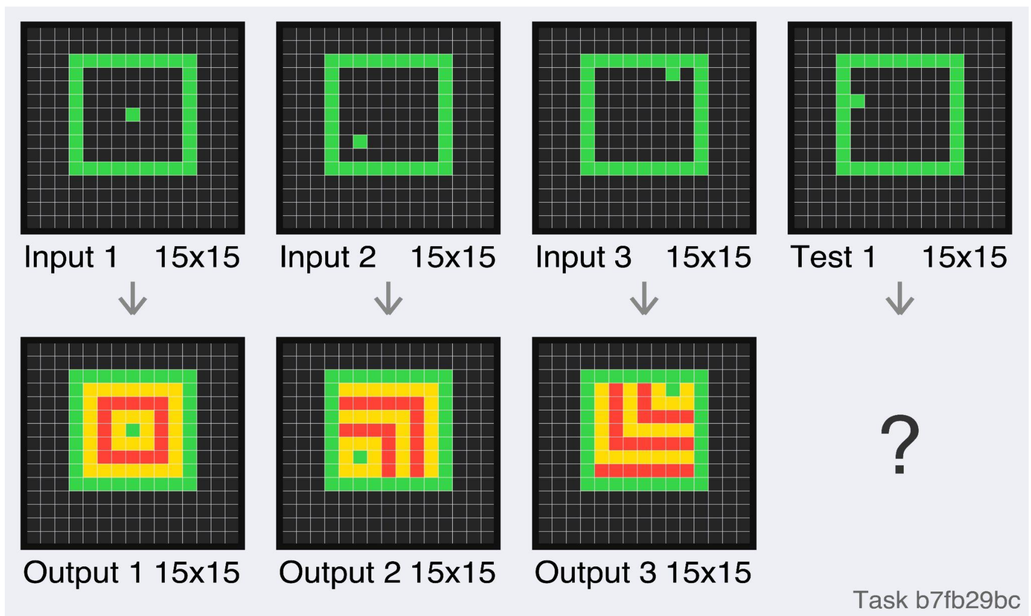
MASR: Multi-Agent System with Reflection for the Abstraction and Reasoning Corpus

Kiril Bikov, Mikel Bober-Irizar, Soumya Banerjee

UNIVERSITY OF CAMBRIDGE

Abstraction and Reasoning Corpus (ARC)

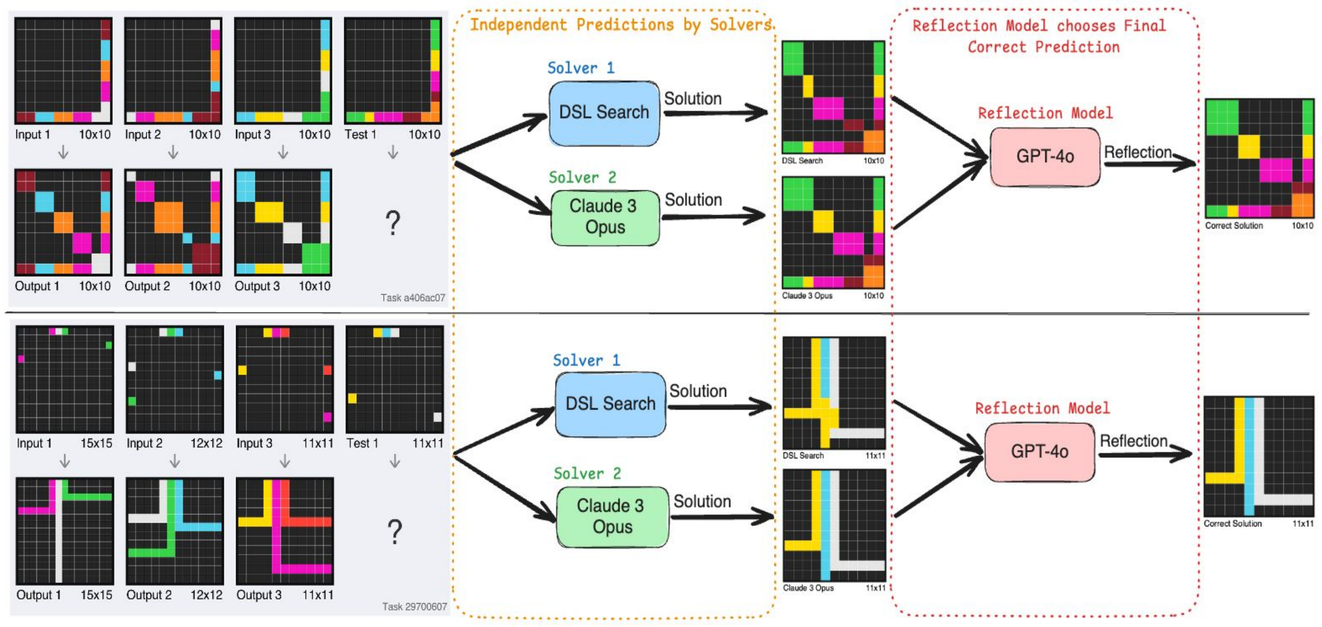
ARC addresses the gap between human intelligence and AI models. IT consists of 1000 visual tasks, capturing essential aspects of abstraction and analogy. Previous solvers of ARC have either been single LLMs or heuristic search systems. We explore how different systems can be combined in a multi-agent setting on ARC.



MASR: Multi-Agent System with Reflection

MASR relies on agents of various architectures: LLMs or Domain Specific Languages using Program Synthesis. When predicting the correct solution to an ARC task, MASR executes in two main stages:

1. **Independent predictions by each agent.**
2. **Reflection over predictions to choose final one.**



AugARC: Augmented ARC for LLMs

To tackle the **limited number of ARC training tasks**, we propose the following augmentation techniques:

- *Rotation*: of each ARC grid by 90° or 270°.
- *Flipping*: *horizontally and vertically.*
- *Permutations*: rearranges input-output pairs.

The AugARC sets vary from 2000 to 18 million tasks.

3-Shot AugARC Benchmark - a unified, easy way to evaluate LLMs on reasoning. Each ARC task starts with a textual description, the ARC grids are represented as a 2D matrix of numbers.

Flexibility of MASR

MASR allows any number of agents to be used, as the reflection model can easily process the outputs of various agents. This makes MASR a highly flexible and customisable architecture, as each of its components - the agents and the reflection model, can easily be changed.

Results of base LLMs on ARC and AugARC

Model	ARC	AugARC	Increase
Llama-2 7B	5/400	7/400	29%
Mistral 7B	9/400	15/400	67%
Llama-2 13B	5/400	8/400	100%
Llama-2 70B	7/400	14/400	100%
Mixtral 8x7B	9/400	18/400	**125%**
Gemini Pro	20/400	**33/400**	65%

Performance of MASR Configurations

Solver 1	Solver 2	Solver 3	Reflection Model	ARC Correct
DSL Search	Claude 3 Opus	-	Llama-3 70B	133/400
DSL Search	Claude 3 Opus	-	GPT-4-turbo	165/400
DSL Search	Claude 3 Opus	-	GPT-4o	**166/400**
DSL Search	Claude 3 Opus	Fine-Tuned Llama-3 8B	Claude 3.5 Sonnet	163/400

Performance of Fine-tuned LLMs on AugARC

Model	Base	Fine-tuned	Increase
Llama-2 7B	7/400	21/400	**200%**
Mistral 7B	15/400	23/400	53%
Llama-2 13B	8/400	18/400	125%
Llama-3 8B	21/400	**34/400**	62%

MASR against Previous Approaches

Program Synthesis	Brute Force	26/400
	Neurodiversity solver	45/400
	DSL Search	160/400
Ensemble	Voting	161/400
Multi-agent	**MASR**	**166/400**

https://github.com/kiril-bikov/AugARC

Reliable Decision-Making for Multi-Agent LLM Systems

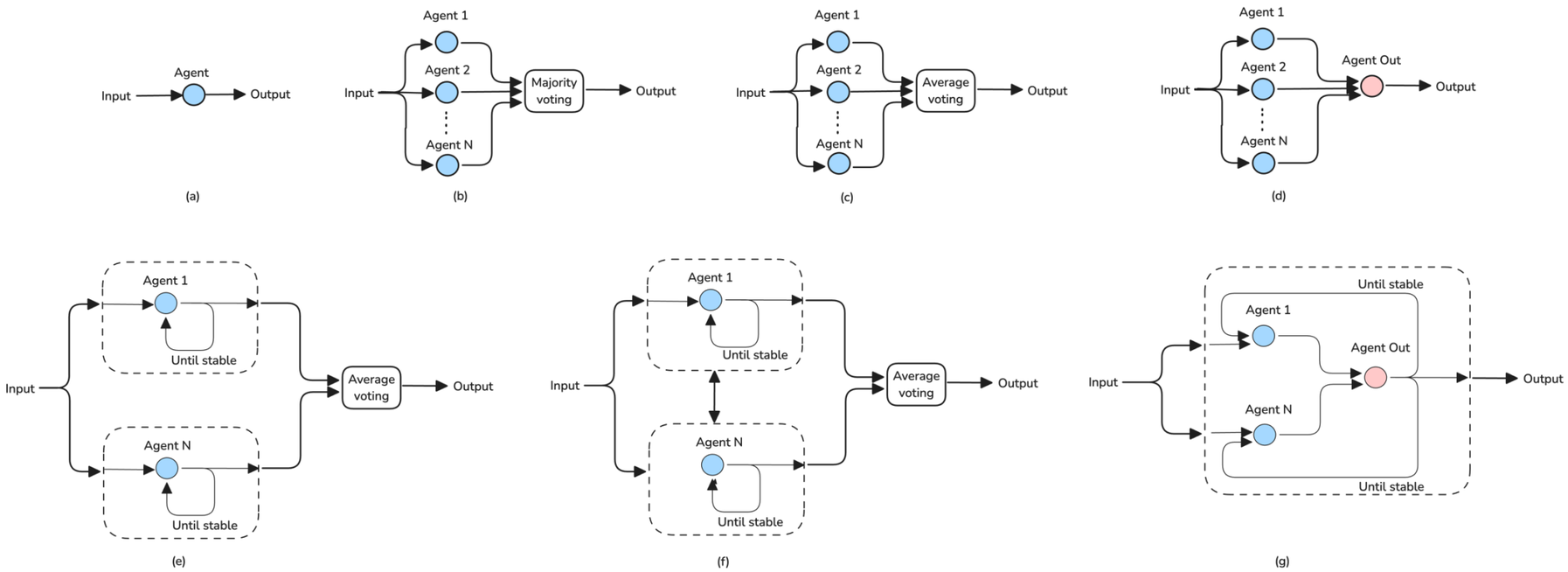


Illustration of different output aggregation strategies

- This work explores the challenges of deploying multi-agent LLM-based systems in industrial environments, where reliability is often more critical than peak performance.
- Through empirical studies on resource allocation, question answering, topic classification, and summarization, we examine how different aggregation strategies impact system consistency, highlighting the need to balance architectural complexity with robustness.

Presenter:
Lasitha Vidyaratne
Industrial AI Lab

Contact:
lasitha.vidyaratne@hal.hitachi.com

Poster ID: #27

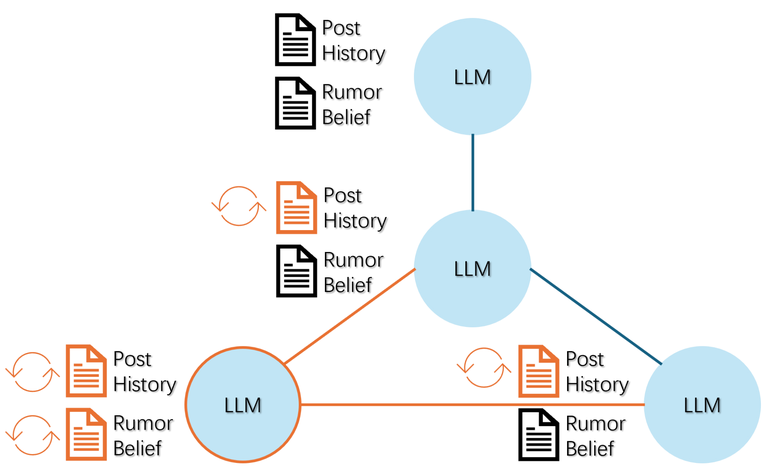
Simulating Rumor Spreading in Social Networks using LLM Agents

Tianrui Hu*†,Dimitrios Liakopoulos*†, Xiwen Wei*,Radu Marculescu*,Neeraja J. Yadwadkar*
*University of Texas at Austin, †These authors contributed equally to this work.

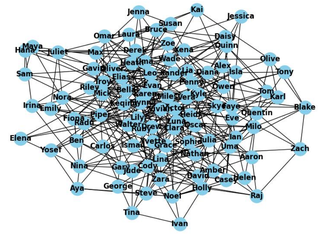
Introduction

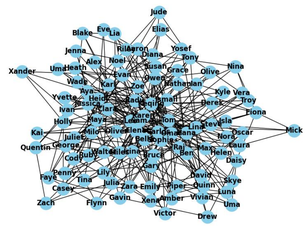
- **Social Network Behavior: Essential for understanding human interactions in social sciences.**
- **LLM-agent-based Framework:**
  1. **LLMs as Agents**
  2. **Rumor Spread Simulation**
  3. **Network Construction**

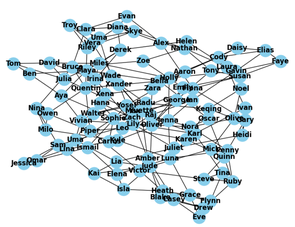
Design

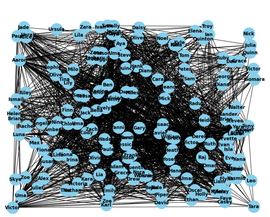


Networks



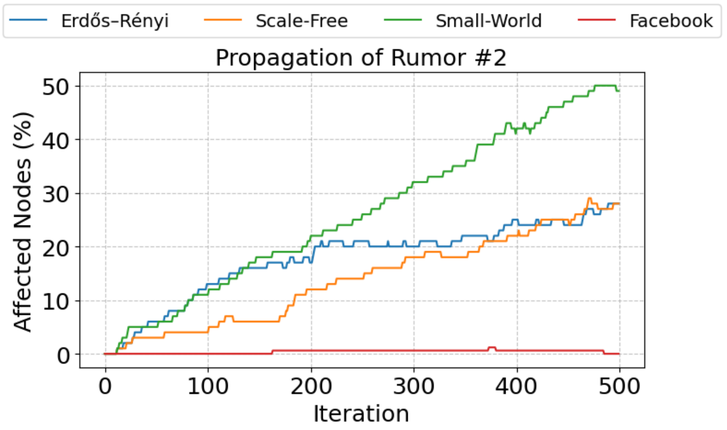


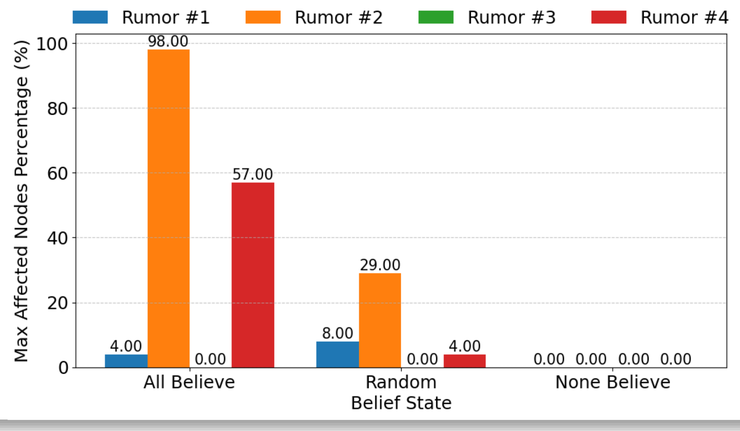


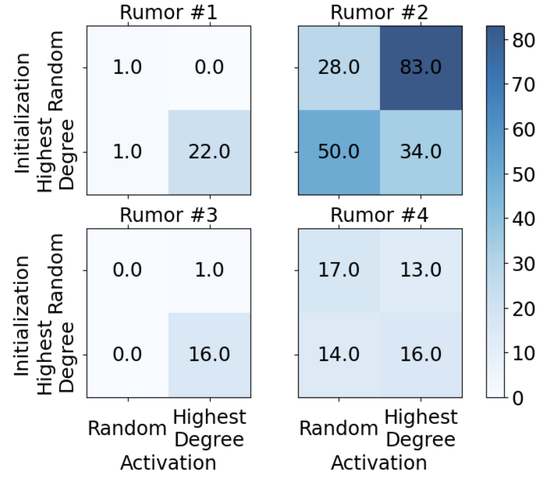


- **Erdős-Rényi**
- **Scale-Free**
- **Small-World**
- **Real World**

Results







- **4 Different rumors**
- **4 Different Network Structures**
- **2 Initialization Schemes**
- **2 Spreading Simulation Schemes**
- **168 Nodes + 1656 Edges**