AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation

Qingyun Wu†, Gagan Bansal∗, Jieyu Zhang±, Yiran Wu†, Beibin Li∗
Erkang Zhu∗, Li Jiang∗, Xiaoyun Zhang∗, Shaokun Zhang†, Jiale Liu∓
Ahmed Awadallah∗, Ryen W. White∗, Doug Burger∗, Chi Wang∗1
∗Microsoft Research, †Pennsylvania State University ±University of Washington,∓Xidian University

Presented by: Akila Peiris

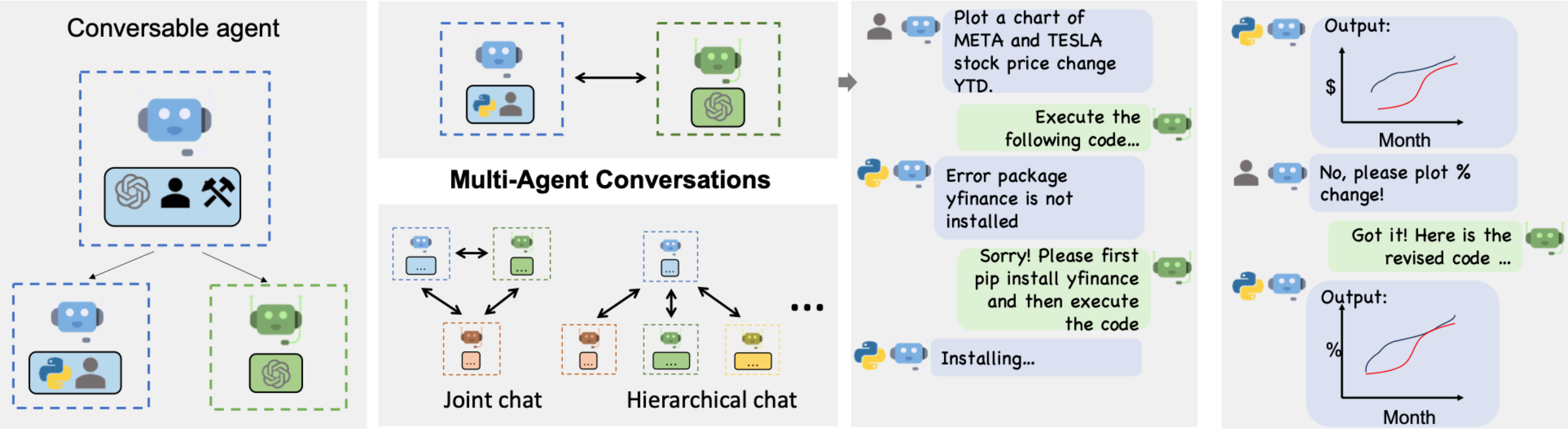
We will cover…

- Autogen
- Background and related work
- Autogen Framework
- Conversable Agents
- Conversation Programming
- Applications of Autogen
- Discussion

Autogen

- Open-source framework
- LLM applications via multiple agents
- Agents, converse with each other to accomplish tasks
- Flexibly define agent interaction behaviors
- Combination of natural language and computer code
- Generic framework for building diverse applications of various complexities and LLM capacities

Autogen



Plot a chart of META and TESLA stock price change YTD.

Execute the following code...

Error package yfinance is not installed

Sorry! Please first pip install yfinance and then execute the code

Installing...

No, please plot % change!

Got it! Here is the revised code ...

Agent Customization

Flexible Conversation Patterns

Example Agent Chat

Background

- Multiple agents can help encourage divergent thinking [1]
- Multiple agents can improve factuality and reasoning [2]
- Multiple agents can provide validation [3]
- Chat- optimized LLMs show the ability to incorporate feedback (converse with each other or humans)

[1] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi- agent debate, 2023.

[2] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improv- ing factuality and reasoning in language models through multiagent debate. arXiv preprint arXiv:2305.14325, 2023.

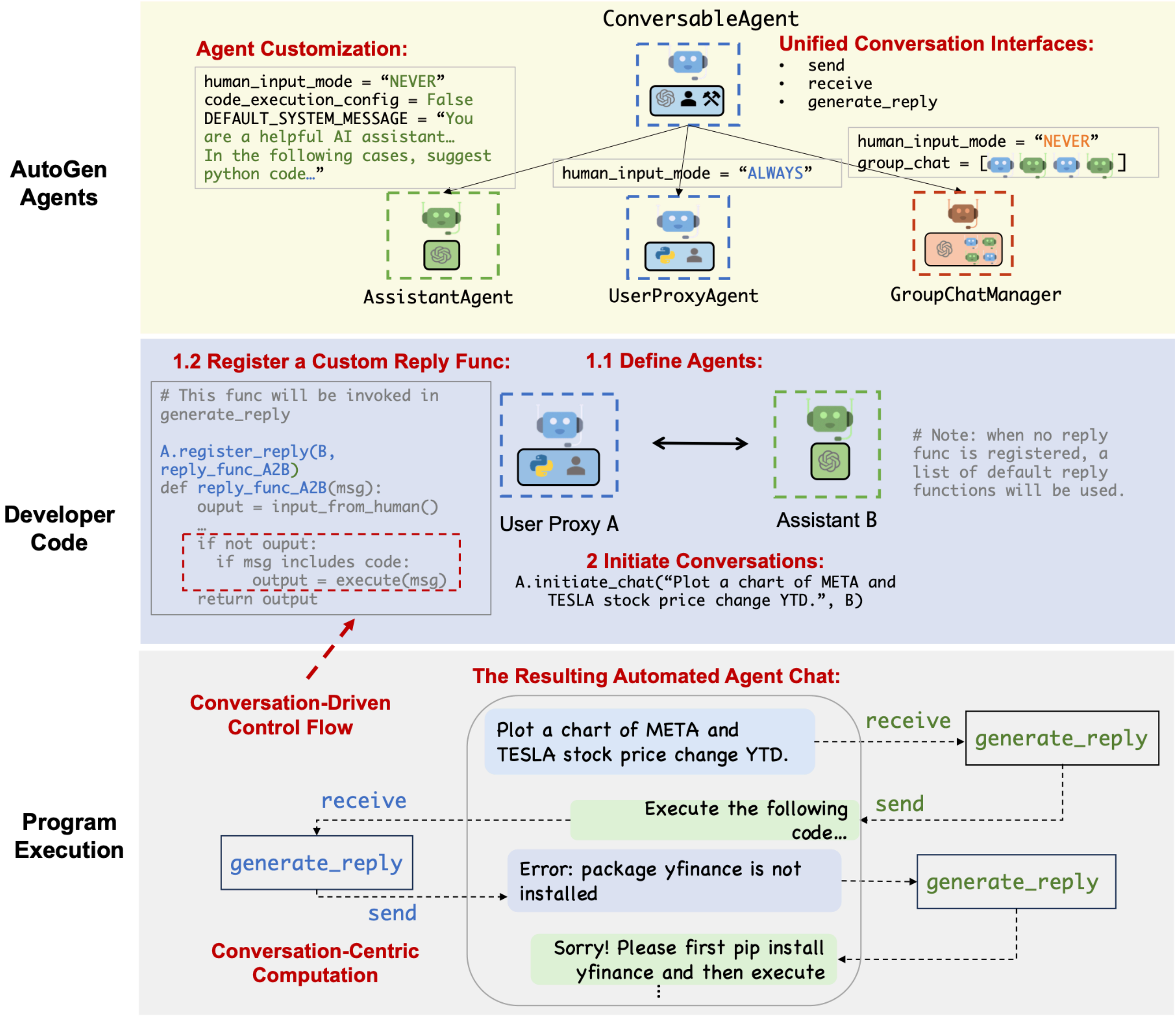
[3] Yiran Wu, Feiran Jia, Shaokun Zhang, Qingyun Wu, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, and Chi Wang. An empirical study on challenging math problem solving with gpt-4. *arXiv preprint arXiv:2306.01337*, 2023.

AutoGen Framework

- Core design principle
  - Streamline and consolidate multi-agent workflows using multi-agent conversations
- Reduces the development effort for
  - Complex LLM applications
  - Various domains
- Maximizes agent reusability
- Two key concepts
  - Conversable agents
  - Conversation programming

Conversable Agents

- Role: pass messages to send and receive information to and from other conversable agents
- Employs combinations of
  - LLMs – role play, implicit state inference and progress making conditioned on conversation history, give feedback, adapt from feedback, code, etc.
  - Human inputs – via human-backed agents at certain rounds of a conversation
  - Tools – can execute tools via code execution or function execution
- Customization and cooperation
  - Easily create agents with specialized capabilities and roles from default agents



```
human_input_mode = "NEVER"
code_execution_config = False
DEFAULT_SYSTEM_MESSAGE = "You
are a helpful AI assistant…
In the following cases, suggest
python code…"
```

- send
- receive
- generate_reply

```
human_input_mode = "ALWAYS"
```

```
human_input_mode = "NEVER"
group_chat = [🤖 🤖 🤖 🤖]
```

```
# This func will be invoked in
generate_reply

A.register_reply(B,
reply_func_A2B)
def reply_func_A2B(msg):
    ouput = input_from_human()
    …
    if not ouput:
      if msg includes code:
          output = execute(msg)
    return output
```

```
# Note: when no reply
func is registered, a
list of default reply
functions will be used.
```

```
A.initiate_chat("Plot a chart of META and
  TESLA stock price change YTD.", B)
```

Plot a chart of META and TESLA stock price change YTD.

Execute the following code…

Error: package yfinance is not installed

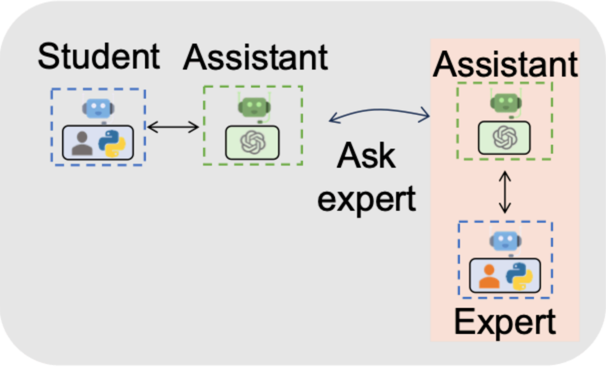
Sorry! Please first pip install yfinance and then execute

⋮

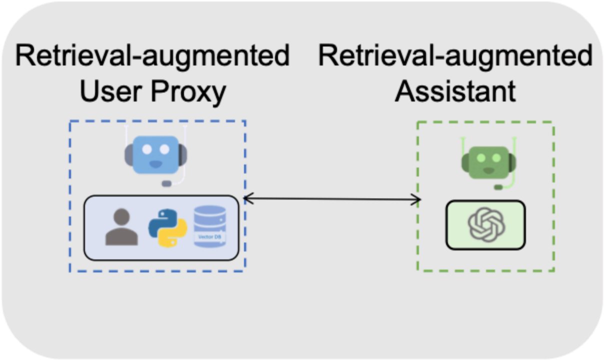
Conversation Programming

- Mold multi-agent conversations to develop applications where agents make *meaningful progress* on tasks
- 2 key concepts: Computation and Control flow
- Computation
  - Actions agents take to compute their response in the conversation
- Control flow
  - The sequence (or conditions) un- der which these computations happen
- Unified interfaces and auto-reply mechanisms for automated agent chat
- Control by fusion of programming and natural language

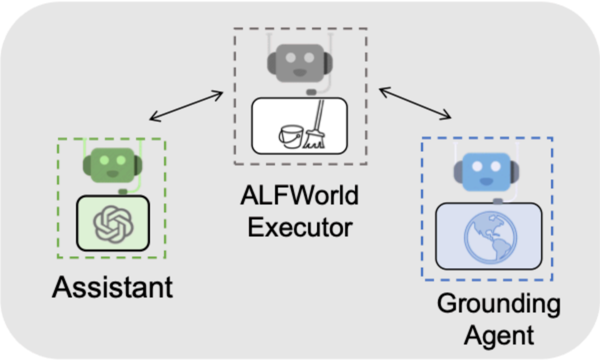
Applications of AutoGen



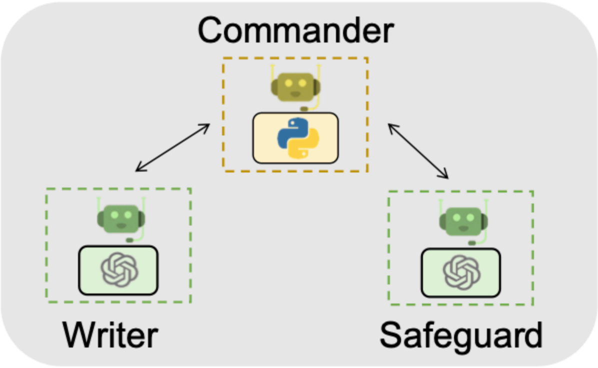
A1. Math Problem Solving



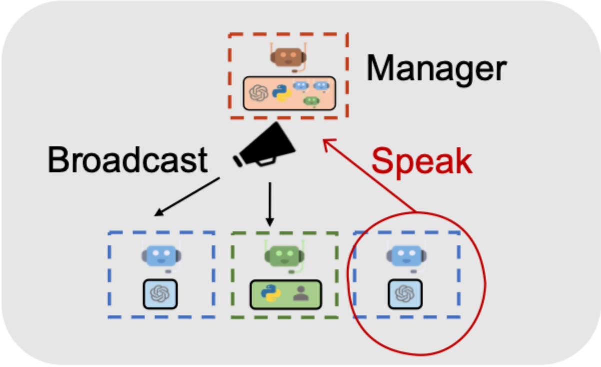
A2. Retrieval-augmented Chat



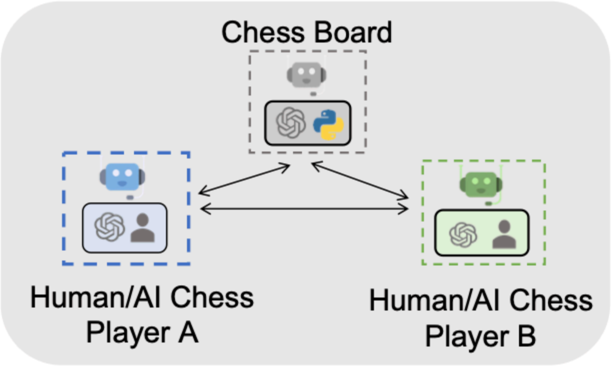
A3. ALF Chat



A4. Multi-agent Coding

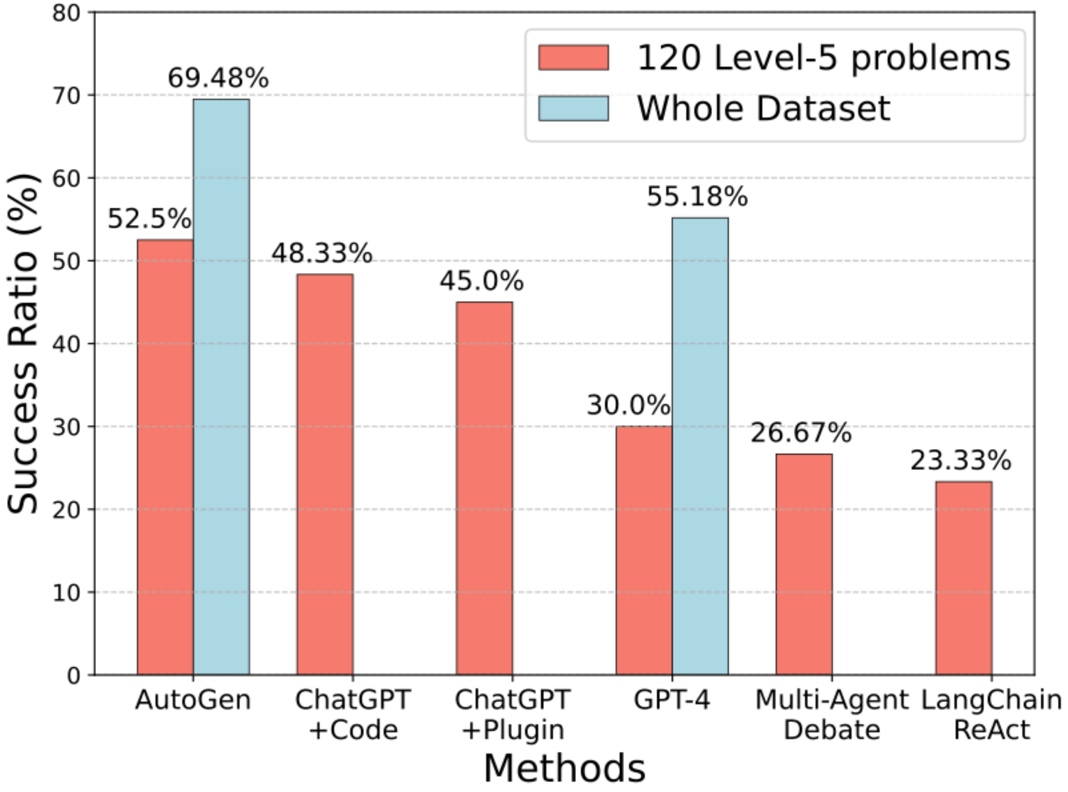


A5. Dynamic Group Chat



A6. Conversational Chess

Applications of AutoGen
Math Problem Solving



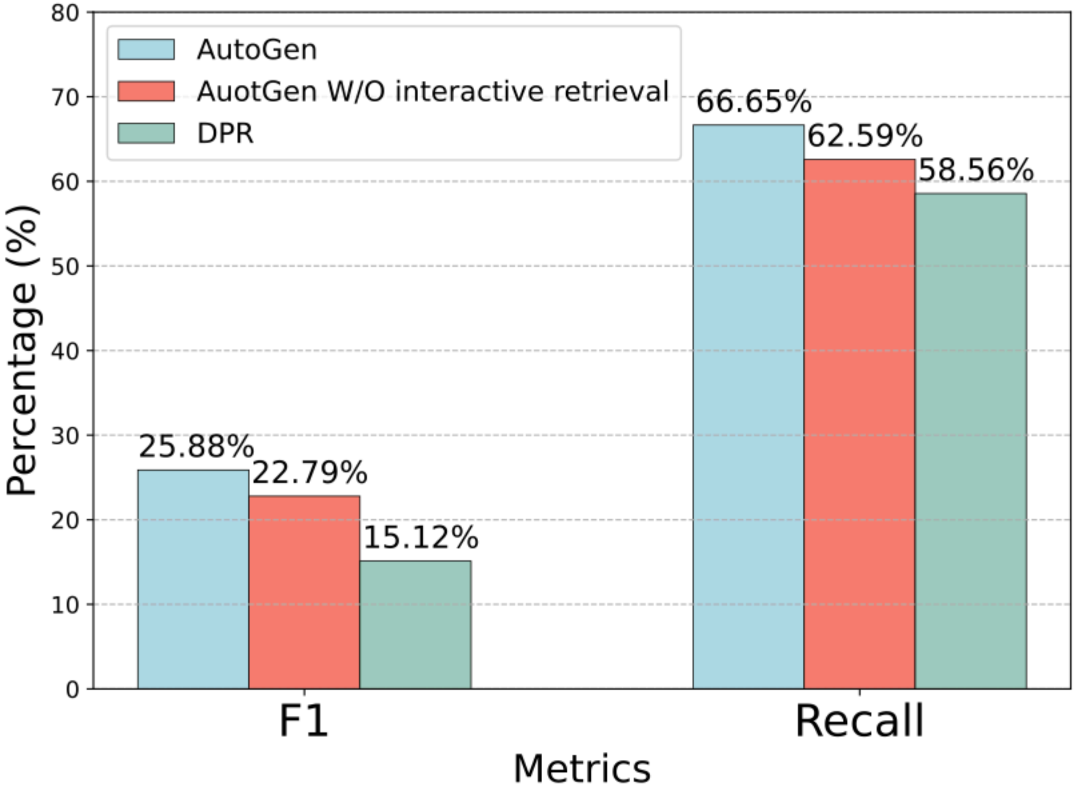
- Scenario 1
  - Autonomous math problem solving
  - 120 randomly selected level-5 problems
  - Test dataset from MATH [4]
- Scenario 2
  - Human-in-the-loop problem-solving
  - Same system in scenario 1 with human input turned on
- Scenario 3
  - Multiple human users can participate in the conversation

[4] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021.

Applications of AutoGen

Retrieval-Augmented Code Generation and Question Answering

- Scenario 1
    - Natural question answering on the Natural Questions dataset [5]
    - Evaluation practice [6]
- Scenario 2
    - Generating code based on a given codebase
    - Code not included in GPT-4's training data



(b) A2: Q&A tasks (w/ GPT-3.5).

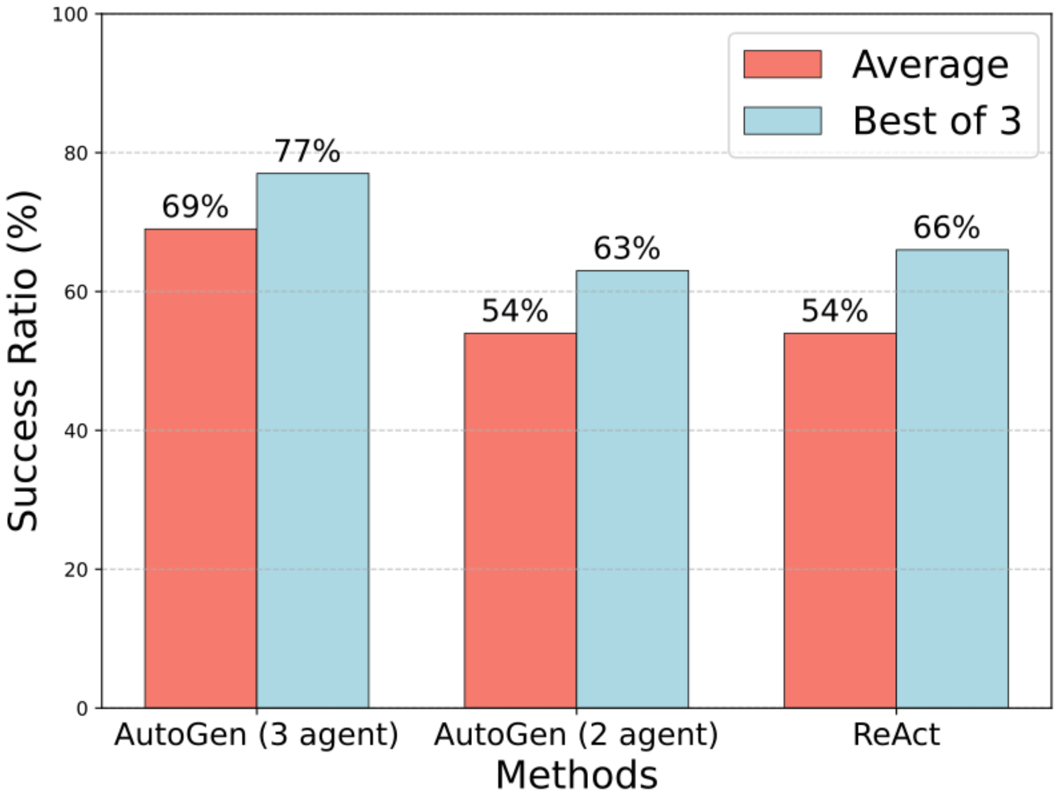
[5] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 2019.

[6] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Eval- uating correctness and faithfulness of instruction-following models for question answering. arXiv preprint arXiv:2307.16877, 2023.

Applications of AutoGen

Decision Making in Text World Environments

- Interactive or online decision making
- 134 unseen tasks from ALFWorld [7] benchmark
  - Synthetic language-based interactive decision making tasks in household environments.
- Common challenge - inability to leverage basic commonsense knowledge about the physical world
- Grounding agent (3 agent system), which supplies crucial commonsense knowledge



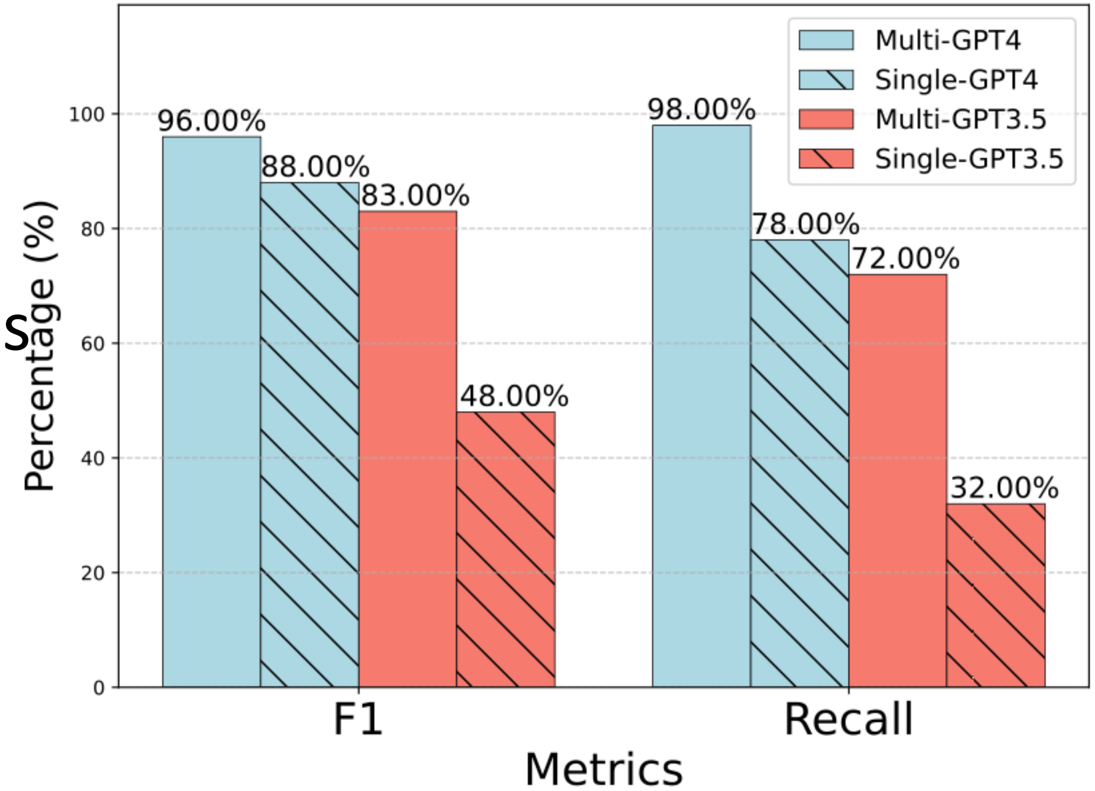
(c) A3: Performance on ALFWorld.

[7] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Coˆte´, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In Proceedings of the International Conference on Learning Representations (ICLR), 2021. URL https://arxiv.org/abs/2010.03768.

Applications of AutoGen

Multi-Agent Coding

- Based on OptiGuide [8]
  - Write code to interpret optimization solutions and answer user questins
- Core workflow code for OptiGuide was reduced from over 430 lines to 100 lines
- AutoGen-based OptiGuide could save around 3x of user's time and reduce user interactions by 3 - 5 times on average



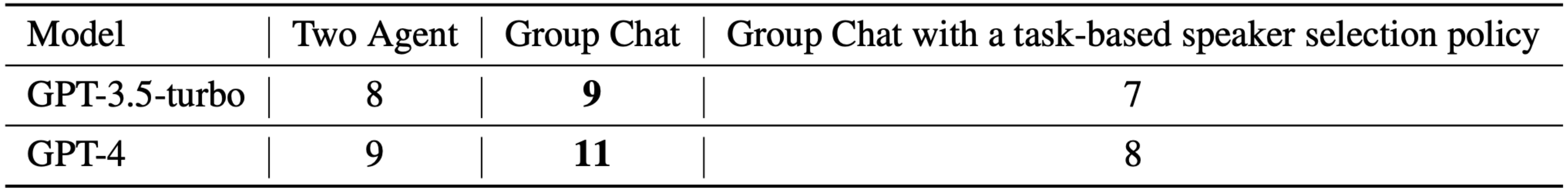
[8] Beibin Li, Konstantina Mellou, Bo Zhang, Jeevan Pathuri, and Ishai Menache. Large language models for supply chain optimization. arXiv preprint arXiv:2307.03875, 2023a.

Applications of AutoGen

Dynamic Group Chat

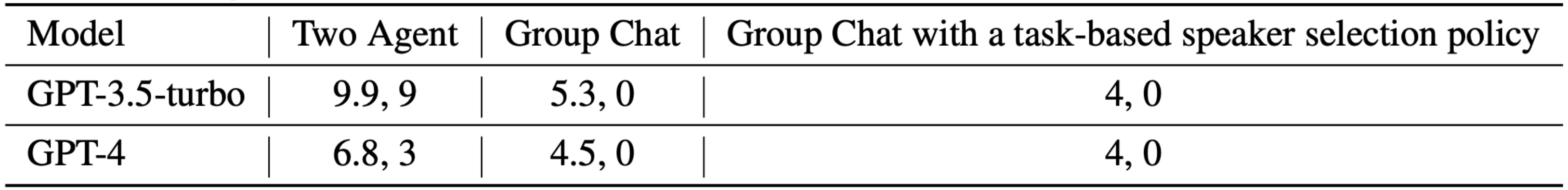
- Agents share the same context and converse with the others in a dynamic manner instead of following a pre-defined order
- 12 manually crafted complex tasks
- Role-play prompt - more effective consideration of both conversation context and role alignment

Table 5: Number of successes on the 12 tasks (higher the better).



Model	Two Agent	Group Chat	Group Chat with a task-based speaker selection policy
GPT-3.5-turbo	8	**9**	7
GPT-4	9	**11**	8

Table 6: Average # LLM calls and number of termination failures on the 12 tasks (lower the better).

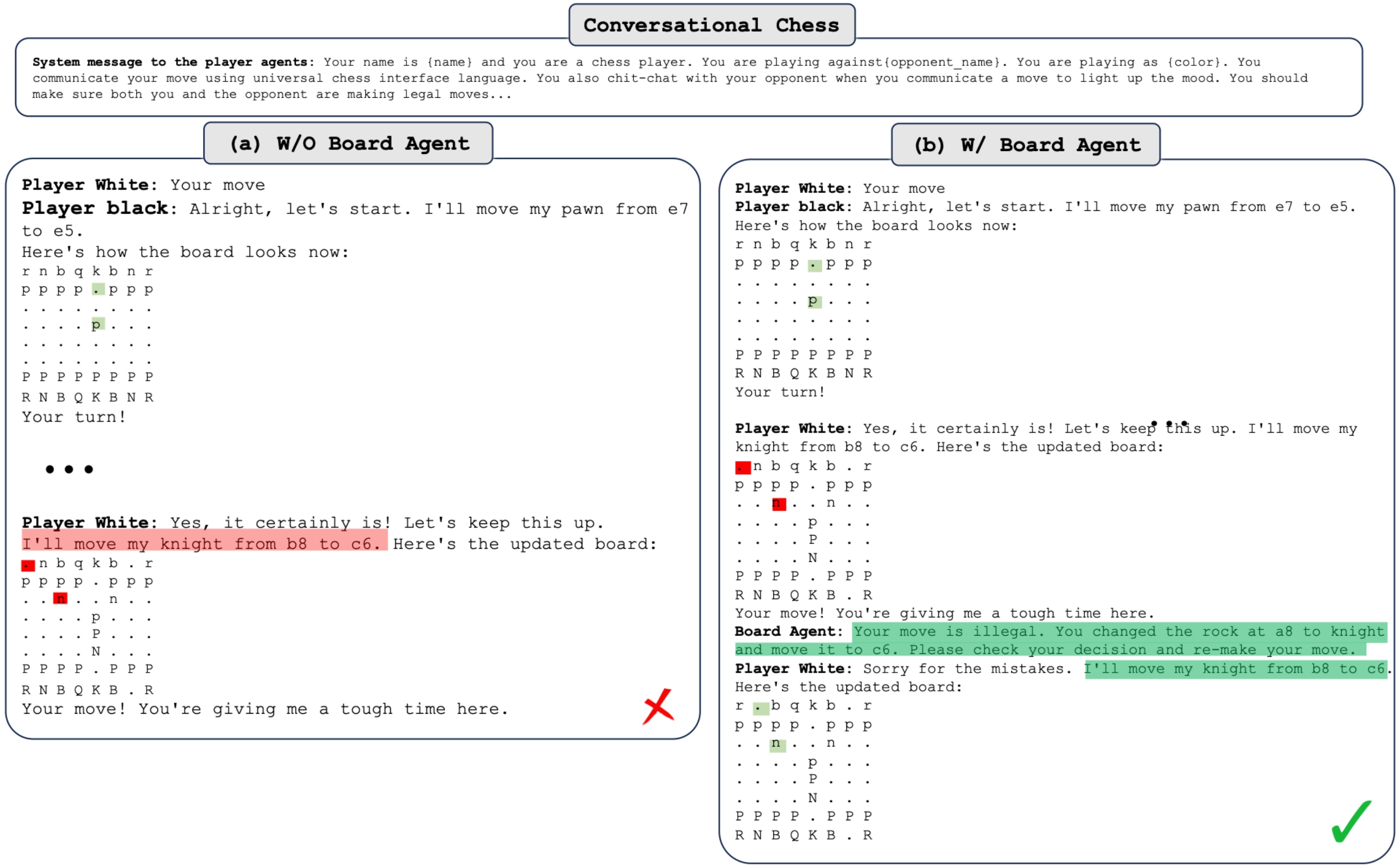


Model	Two Agent	Group Chat	Group Chat with a task-based speaker selection policy
GPT-3.5-turbo	9.9, 9	5.3, 0	4, 0
GPT-4	6.8, 3	4.5, 0	4, 0

Applications of AutoGen

Conversational Chess

- Modes: AI-AI, AI-human, and human-human
- Seamless switching between these modes during a single game
- Board agent - grounding, which is a crucial aspect to maintain game integrity



Conversational Chess

System message to the player agents: Your name is {name} and you are a chess player. You are playing against{opponent_name}. You are playing as {color}. You communicate your move using universal chess interface language. You also chit-chat with your opponent when you communicate a move to light up the mood. You should make sure both you and the opponent are making legal moves...

(a) W/O Board Agent

Player White: Your move
Player black: Alright, let's start. I'll move my pawn from e7 to e5.
Here's how the board looks now:

```
r n b q k b n r
p p p p . p p p
. . . . . . . .
. . . . p . . .
. . . . . . . .
. . . . . . . .
P P P P P P P P
R N B Q K B N R
```

Your turn!

• • •

Player White: Yes, it certainly is! Let's keep this up. I'll move my knight from b8 to c6. Here's the updated board:

```
. n b q k b . r
p p p p . p p p
. . n . . n . .
. . . . p . . .
. . . . P . . .
. . . . N . . .
P P P P . P P P
R N B Q K B . R
```

Your move! You're giving me a tough time here.

✗

(b) W/ Board Agent

Player White: Your move
Player black: Alright, let's start. I'll move my pawn from e7 to e5.
Here's how the board looks now:

```
r n b q k b n r
p p p p . p p p
. . . . . . . .
. . . . p . . .
. . . . . . . .
. . . . . . . .
P P P P P P P P
R N B Q K B N R
```

Your turn!

• • •

Player White: Yes, it certainly is! Let's keep this up. I'll move my knight from b8 to c6. Here's the updated board:

```
. n b q k b . r
p p p p . p p p
. . n . . n . .
. . . . p . . .
. . . . P . . .
. . . . N . . .
P P P P . P P P
R N B Q K B . R
```

Your move! You're giving me a tough time here.
Board Agent: Your move is illegal. You changed the rock at a8 to knight and move it to c6. Please check your decision and re-make your move.
Player White: Sorry for the mistakes. I'll move my knight from b8 to c6.
Here's the updated board:

```
r . b q k b . r
p p p p . p p p
. . n . . n . .
. . . . p . . .
. . . . P . . .
. . . . N . . .
P P P P . P P P
R N B Q K B . R
```

✓

Discussion

- Introduced an open-source library, AutoGen
- Conversable agents and conversation programming
- Benefits observed in multi-agent conversation systems
  - Improved performance (over state-of-the-art approaches)
  - Reduced development code
  - Decreased manual burden for existing applications
- Offers flexibility to developers
- Allows humans to engage in activities alongside multiple AI agents in a conversational manner

Ethics statement

- Privacy and Data Protection: The framework allows for human participation in conversations between agents. It is important to ensure that user data and conversations are protected, and that developers use appropriate measures to safeguard privacy.
- Bias and Fairness: LLMs have been shown to exhibit biases present in their training data (Navigli et al., 2023). When using LLMs in the AutoGen framework, it is crucial to address and mitigate any biases that may arise in the conversations between agents. Developers should be aware of potential biases and take steps to ensure fairness and inclusivity.
- Accountability and Transparency: As discussed in the future work section, as the framework in- volves multiple agents conversing and cooperating, it is important to establish clear accountability and transparency mechanisms. Users should be able to understand and trace the decision-making process of the agents involved in order to ensure accountability and address any potential issues or biases.
- Trust and Reliance: AutoGen leverages human understanding and intelligence while providing automation through conversations between agents. It is important to consider the impact of this interaction on user experience, trust, and reliance on AI systems. Clear communication and user education about the capabilities and limitations of the system will be essential (Cai et al., 2019).
- UnintendedConsequences:Asdiscussedbefore,theuseofmulti-agentconversationsandautoma- tion in complex tasks may have unintended consequences. In particular, allowing LLM agents to make changes in external environments through code execution or function calls, such as installing packages, could be risky. Developers should carefully consider the potential risks and ensure that appropriate safeguards are in place to prevent harm or negative outcomes.

Acknowledgements

The work presented in this report was made possible through discussions and feedback from Peter Lee, Johannes Gehrke, Eric Horvitz, Steven Lucco, Umesh Madan, Robin Moeur, Piali Choudhury, Saleema Amershi, Adam Fourney, Victor Dibia, Guoqing Zheng, Corby Rosset, Ricky Loynd, Ece Kamar, Rafah Hosn, John Langford, Ida Momennejad, Brian Krabach, Taylor Webb, Shanka Subhra Mondal, Wei-ge Chen, Robert Gruen, Yinan Li, Yue Wang, Suman Nath, Tanakorn Leesatapornwongsa, Xin Wang, Shishir Patil, Tianjun Zhang, Saehan Jo, Ishai Menache, Kontantina Mellou, Runlong Zhou, Feiran Jia, Hamed Khanpour, Hamid Palangi, Srinagesh Sharma, Julio Albinati Cortez, Amin Saied, Yuzhe Ma, Dujian Ding, Linyong Nan, Prateek Yadav, Shannon Shen, Ankur Mallick, Mark Encarnación, Lars Liden, Tianwei Yue, Julia Kiseleva, Anastasia Razdaibiedina, and Luciano Del Corro. Qingyun Wu would like to acknowledge the funding and research support from the College of Information Science and Technology at Penn State University.

References

[1] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi- agent debate, 2023.

[2] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improv- ing factuality and reasoning in language models through multiagent debate. arXiv preprint arXiv:2305.14325, 2023.

[3] Yiran Wu, Feiran Jia, Shaokun Zhang, Qingyun Wu, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, and Chi Wang. An empirical study on challenging math problem solving with gpt-4. arXiv preprint arXiv:2306.01337, 2023.

[4] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021.

[5] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 2019.

[6] Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Eval- uating correctness and faithfulness of instruction-following models for question answering. arXiv preprint arXiv:2307.16877, 2023.

[7] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Coˆte ́, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In Proceedings of the International Conference on Learning Representations (ICLR), 2021. URL https://arxiv.org/abs/2010.03768.

[8] Beibin Li, Konstantina Mellou, Bo Zhang, Jeevan Pathuri, and Ishai Menache. Large language models for supply chain optimization. arXiv preprint arXiv:2307.03875, 2023a.