

AutoGen

Enabling Next-Gen LLM Applications via Multi-Agent Conversations : [[arXiv](#)] [[GitHub](#)]

Qingyun Wu et al., COLM 2024

Shikhar Shiromani

Agentic AI for Oncology

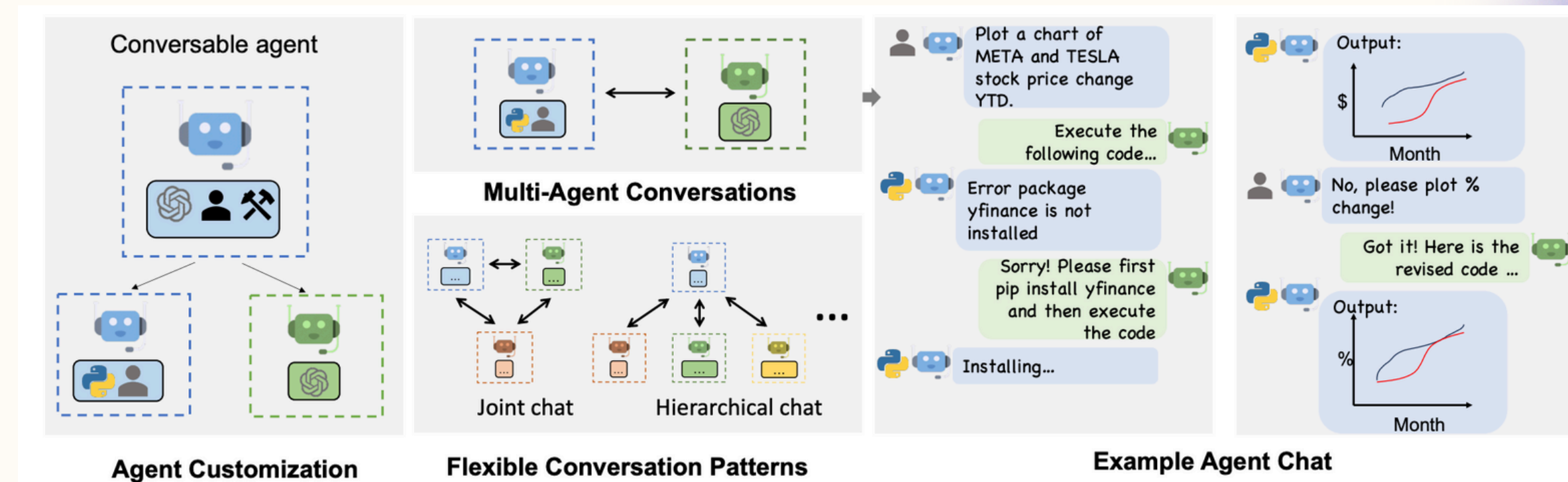
Why AutoGen?

- Single-agent LLMs hit limits in complex, multi-step tasks
- Many real problems need multiple specialized roles
- Existing multi-agent frameworks are rigid or code-heavy

Solution : *Multi-Agent Collaboration*

- Designing capable and customizable agents for effective multi-agent collaboration
- Creating a unified interface to support diverse conversation patterns

Core Idea = Multi-Agent Apps via Conversations



- Conversable Agents: LLM, human, or tool-driven
- Conversation Programming: define workflows via send/receive
- Modular, human-in-the-loop friendly

The Framework

AutoGen Agents

Agent Customization:

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

AssistantAgent

ConversableAgent

human_input_mode = "ALWAYS"

UserProxyAgent

Unified Conversation Interfaces:

- send
- receive
- generate_reply

```
human_input_mode = "NEVER"
group_chat = [AssistantAgent, UserProxyAgent, AssistantAgent]
```

GroupChatManager

Developer Code

1.2 Register a Custom Reply Func:

```
# This func will be invoked in
generate_reply

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

1.1 Define Agents:



User Proxy A



Assistant B

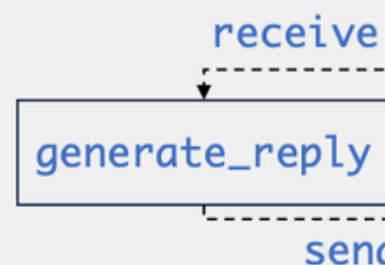
Note: when no reply
func is registered, a
list of default reply
functions will be used.

2 Initiate Conversations:

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

Program Execution

Conversation-Driven Control Flow



Conversation-Centric Computation

The Resulting Automated Agent Chat:

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

receive

generate_reply

Execute the following
code...

send

Error: package yfinance is not
installed

generate_reply

Sorry! Please first pip install
yfinance and then execute
:

Key Applications

- Math Problem Solving => Solver + Checker loop → better MATH benchmark scores
- Retrieval-Augmented QA => Iterative retrieval → higher factual accuracy
- ALFWorld Decision-Making => Grounding agent → +15% success rate
- Supply Chain Optimization => 3-agent workflow → code cut 430→100 lines

Pros & Cons

Advantages

- Modular, extensible, easy to prototype, natural language + code, human-in-loop

Limitations

- Safety risks, overhead for simple tasks, bias & privacy concerns

Related Stack

- Orchestration: LangGraph (supervisor + specialist agents for pathology, oncology, radiomics, tumor triage)
- Observability & evals: LangSmith or Langfuse (open-source) for tracing, dataset-based evals, versioning, and regression checks
- Research loop: AutoGen notebooks to experiment with agent-to-agent strategies before promoting flows into LangGraph

Key Takeaways

Objectives

- AutoGen simplifies building practical multi-agent LLM apps
- Improves performance across varied domains
- Bridges natural language workflows with programmable control

Q&A.

Thank you for listening!