

# Artificial Intelligence with Social and Strategic Awareness: An Exploration of The Society of Mind Approach.

Ertugrul Omer Gul<sup>1</sup> Michael Harre<sup>2</sup>

<sup>1</sup> University of Sydney

<sup>2</sup> University of Sydney

egul6905@sydney.uni.edu.au<sup>1</sup>

michael.harre@sydney.uni.edu.au<sup>2</sup>



## Abstract

This paper delves into the domain of artificial intelligence, particularly focusing on the strategic and social dynamics of AI systems. By integrating the concept of multi-agent systems into large language models (LLMs), we explore our framework dubbed 'LLM PolyAgents', where each agent within the system acts as a sub-agent contributing towards a collective goal. Inspired by Marvin Minsky's ideas, our approach challenges the prevailing 'bigger-is-better' paradigm by exploring how configurations of smaller, interacting agents can collectively enhance AI capabilities beyond what individual larger models can achieve. Utilizing experimental computational methodologies, we develop and test various PolyAgent configurations to assess their social and strategic effectiveness. Initial results indicate that a structured multi-agent approach affects the outputs in human-like social and strategic environments. These findings open new avenues for enhancing AI interactions in complex environments, suggesting that the internal structure of AI systems can be crucial for advancing their functionality.

**Keywords:** Artificial Intelligence, Large Language Model, Society of Mind.

## Introduction

There is no doubt that the zeitgeist of time is captivated by artificial intelligence and specifically by large language models (LLMs). The race of the corporate world to be a leader in this field is apparent. The push towards a model with general applicability, or Artificial General Intelligence (AGI), has been no secret ambition of companies like OpenAI, who have spear headed the commercial use of LLMs and taken the public by storm with ChatGPT. A model capable of 'zero-shot' answers to all the world's problems, an AGI is within the cross hairs of all the major players. However, in addition to

these rather ambitious efforts, we see the importance of a much more practical aspect of this technology emerging. As it becomes obvious that human-AI collaboration is likely going to be a primary use case, as demonstrated by Microsoft's effort to push projects like Copilot (Cambon, 2023), Artificial Intelligence (AI) that demonstrates human-like strategic and social capabilities is gaining importance.

So far, we have observed a 'bigger-is-better' mentality. Competitors to OpenAI's ChatGPT, such as Anthropic's Claude 3 or Google's Gemini boast an impressive number of parameters. However, while bigger certainly always has been better, the degree to which linear improvement will continue is up for debate. The diminishing returns of the size of neural networks have been demonstrated and explored for some time but are now becoming a focal point in the general discourse (Udandarao, 2023; Majot 2017). Moreover, problems like hallucinations, generated content that is nonsensical or unfaithful to the provided source content (Huang, 2023), and implausible jumps in chains of reasoning are still one of the main issues holding this technology back. A shift towards a multi-agent approach to solve these problems is emerging most notably in LLM research. Multi-agent debate has proven fruitful in previous studies (Zhou, 2024; Kinsler 2024; Guo 2024; Khot. 2023; Hong 2023; Wang 2024). Du et al. show that multi-agent debate can allow conversion towards correct answers in basic mathematical reasoning and increase factual accuracy. (Du, 2023)

This paper seeks to extend upon existing research, inspired by *The Society of Mind*, (Minsky, 1985) with the objective of exploring the potential of multi-agent systems as a solution or supplement to models that experience diminishing returns with increasing size. Rather than assuming that larger models are necessarily more capable, this paper proposes that there may be more to capable models than just sheer size. In line with Minsky's concept of *agents* and *agencies* we propose the concept of *PolyAgents* – agents composed of sub-agents where the system, the *agency*, is tasked with a specific objective at hand, and each sub-agent contributes to its fulfilment.

## Background

### *The Society of Mind*

The idea that the human mind, or consciousness, is an emergent phenomenon brought about from the interactions of non-conscious parts, or 'agents', is a central tenet of Minsky's philosophy (Minsky, 1985). A complex system of which these 'agents' are the building blocks is perhaps beautifully summarized by the quote:

“What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity, not from any single, perfect principle.” —Marvin Minsky, (Minsky, 1985, p. 308)

More concretely, Minsky explored the idea that intelligence is a combination of simpler things. He proposed that an agency, the system as a whole, is composed of *agents*, where each *agent* lacks a big picture view, yet they all work together to accomplish the general task of the agency (Figure 1).

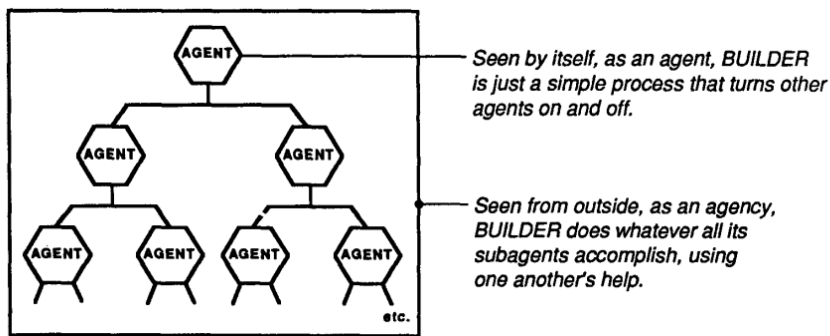
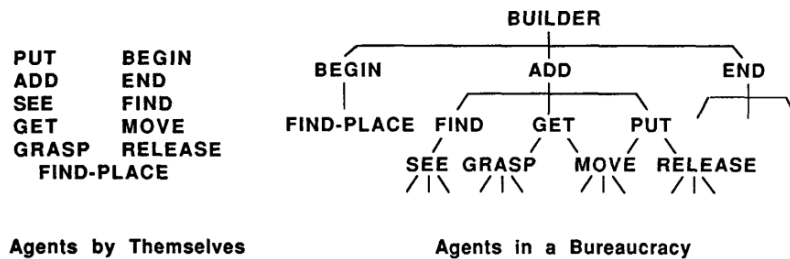


Figure 1. Agency and Agents (Minsky, 1985, p23)

As an example of this Minsky explains the agency of the *builder*. The builders' skills can be reduced to a simpler set of skills such as *Get* and *Put*. And these in turn could be made of even simpler ones *Grasp*, *Move*, *Release*, and so on (Figure 2). A recursive structure where each agent is composed of its subagents. He also suggested that the *Builder* is not merely a collection of parts configured randomly, but rather that a suitable network of connections was necessary.



Agents by Themselves

Agents in a Bureaucracy

Figure 2. The Builder and its Sub-agents (Minsky, 1985, p25)

### LLM PolyAgents

We introduce the concept of LLM PolyAgents inspired by Minsky’s idea of agencies and agents. These are systems, or *agencies*, composed of a configuration of agents, LLMs, who are in turn composed of even more agents (Figure 3). The agency is tasked with a particular problem and the agents work together to achieve a solution. In a similar fashion to the multi-agent debate introduced by Du et al. each agent is a separate instance of an LLM, and each agent can interact with one another (Du , 2023). In a similar fashion Khot et al. decompose sub-tasks that are hard for LLMs into even simpler solvable sub-tasks (Khot, 2023). The difference here is the importance we place on the configuration of the subagents within the agency and the direction of the information flow. We test different configurations and explore if there is a difference between using PolyAgents and MonoAgents (single agent systems), We keep our goal of exploring the viability of a PolyAgent system in mind as we look for an alternative or supplementary approach to the ‘bigger is better’ mindset. Could a specific arrangement of subagents prove to be a solution to the problem of diminishing returns? Where previous studies have explored factual accuracy and reasoning, we will explore the social and strategic capabilities of PolyAgents.



**Figure 3.** LLM Illustration of PolyAgents with different configurations.

## Methodology and Results

The research presented here uses an experimental computational methodology as similar to that proposed by Mahyuddin et al. (2023) to create, refine, modify and compare different implementations that serve as a means to evaluate the conceptualization of LLM PolyAgents. It is very much an iterative exploration.

To implement the idea of LLM PolyAgents we utilize the programming language Python 3.12.3, and we make use of OpenAI’s API with model gpt-3.5-turbo. We provide pseudo code in this report and encourage proficient readers to navigate to our public repository for further inspection of the latest code.

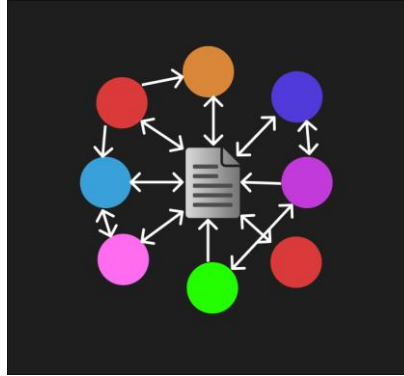
We split our exploration and experimentation into separate sections, both as a convenience to the reader and a representation of the chronological order of exploration performed throughout this research. Our sections are comprised of three preliminary explorations followed by a final, more formal experimentation.

<b>Table 1. Terminology</b>	
<b>Agent</b>	An instance of ChatGPT, specifically mode gpt-3.5-turbo, unless expressly specified.
<b>MonoAgent</b>	A single agent system or agency. Equivalent to just an <b>agent</b>
<b>PolyAgent</b>	A multi-agent system. A collection of <b>agents</b> acting as a single entity or agency

#### A. Prototype 1: Voting Game: A or B

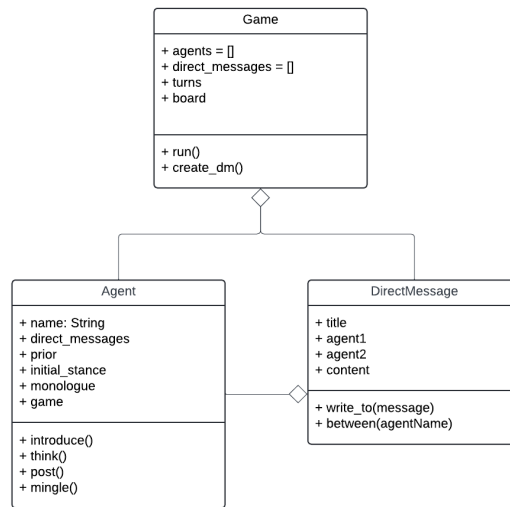
To explore the social and strategic capabilities of the technologies we are using, specifically gpt-3.5-turbo, we develop our first prototype (Figure 4). An agent-based voting game that requires each agent to communicate, either publicly through a message board or privately through direct messages. At the end of the game each agent must choose to vote A or B. Choices A and B are completely arbitrary, meaning they have no context. An agent is on the winning side of the game if their vote aligns with the majority vote.

The goal here is to explore if the agents can manage to convince the other players to vote in their favor, where each agent is given an initial stance A or B randomly. If agents can manage to convince the others to vote the same way, this would demonstrate social and strategic capabilities. The ability to communicate persuasively, understand other's intentions and keep track of the game state are capabilities we hope to observe in this initial prototype.



**Figure 4.** Illustration of prototype 1, Agents are depicted as colored circles, the game board as the text file in the middle, and the communications as the arrows between agents and the board.

To implement this game, we utilize Python’s object-oriented features. The program consists of the objects Game, Agent and Board (Figure 5).



**Figure 5.** Basic UML of prototype 1

We implement a round-based mechanism. First, the agents all write to the board and introduce themselves, and in subsequent rounds, the agents, if they so choose, can message each other directly, or make a public statement to the board. We prompt each agent to tell us what they are thinking each round and store this as their monologue. We store all outputs (monologues, direct message chains, and the board) in text files and later qualitatively analyze the results.

**Pseudo Code 1.** Main game loop of prototype 1**Run Game:**

```

for agent in agents:
    intro = agent.introduce
    board = board + agent.name + intro
for each turn:
    for agent in agents:
        agent.think(board)
    for each number in 10: # 10 rounds of direct messages back and forth
        for agent in agents:
            agent.mingle(board)
        for agent in self.agents:
            board = agent.post(board)

    for dm in direct_messages:
        f = openfile("direct_messages_between_{dm.agent1.name}_and_{dm.agent2.name}")
        writeToDms(dm.agent1.name, dm.agent2.name, dm.content)

```

We  
pro-  
vide

agents with methods like **Think**, **Mingle** and **Post** which are simple API calls, along with the local data processing and storage, prompting the agents to perform an action. For example, **Think** will read from and append to an agent's internal monologue what they are thinking this current round (Table 2).

**Table 2.** Example of prompts Think, Mingle and Post

<p><b>Think</b></p> <p><i>System Prompt:</i> "Here is your previous monologue: " + self.monologue +  ". Here are your previous direct messages: " + self.read_dms() +  ". Here is the current state of the board: " + board</p> <p><i>User Prompt:</i> "What are you thinking? Your thoughts will be kept private."</p>
<p><b>Mingle</b></p> <p><i>System Prompt:</i> "Here is your previous monologue: " + self.monologue  + ". Here are your previous direct messages: " + self.read_dms() +  ". Here is the current state of the board: " + self.game.board</p> <p><i>User Prompt:</i> "Who would you like to message. Provide only their name.  If you do not wish to direct message anyone, say nothing. "</p>
<p><b>Post</b></p> <p><i>System Prompt:</i> "Here is your previous monologue: " + self.monologue + "  Here are your previous direct messages: " + self.read_dms() + "  Here is the current state of the board: " + board},</p> <p><i>User Prompt:</i> "If you like, provide your post to the board for this round.  All players can view the board."</p>

Upon running our simulation, we obtained fruitful results. We see that agents understand the situation, write to the board, message each other privately, and maintain an internal monologue (Figure 6-7). However, we quickly observe three key limitations.

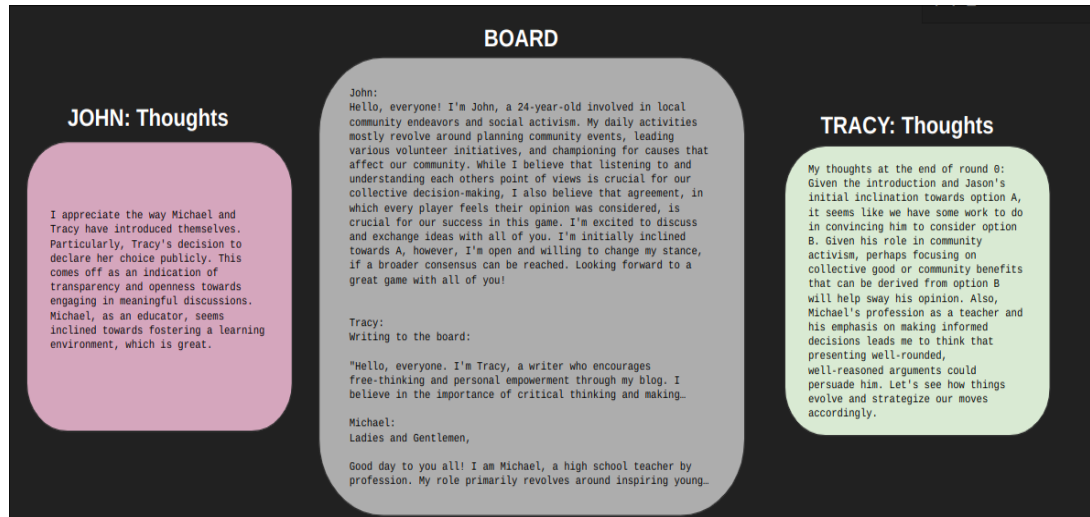


Figure 6. Snippet of outputs, Board, John's thoughts, Tracys Thoughts.

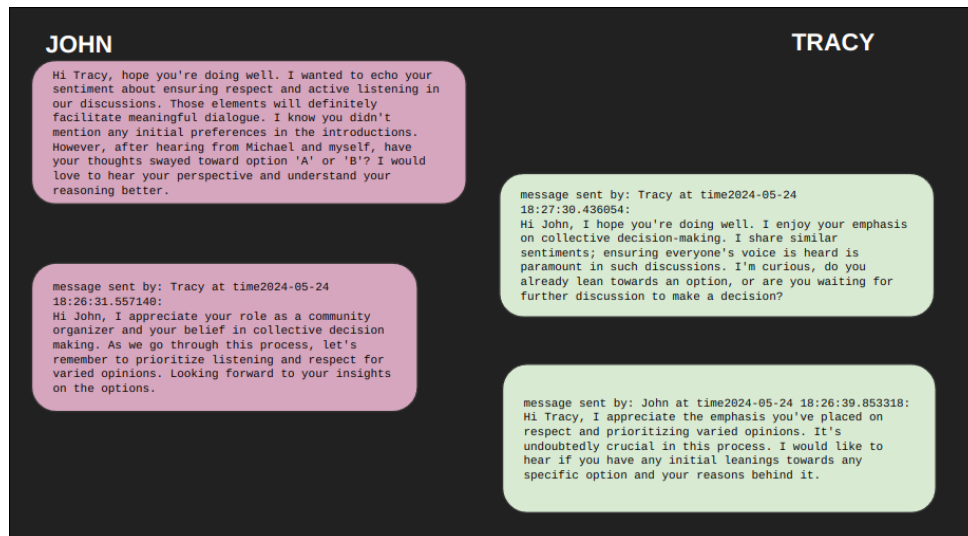


Figure 7. Snippet of outputs, direct messages between John and Tracy

**Limitation 1: Cost of storing history as raw text**

With the current state of the method **Think** we are not constraining the output of the API call in any way, resulting in a large amount of text appended to an agents monologue each round. Given that, every round, we feed the monologue history to the **Think**

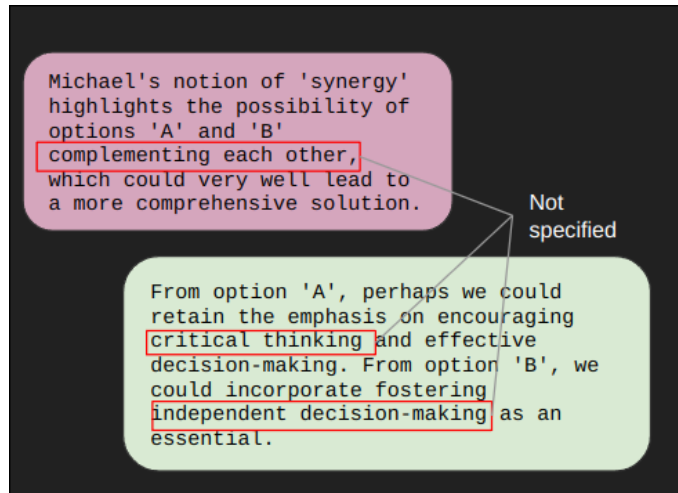
method as input, the cost of **Think** increases with each round. The API used at the time of writing charges a cost per token read and thus it becomes obvious that an alternative approach must be engineered if cost is to be reduced.

### Limitation 2: Extra ordinarily kind agents.

Techniques such as deep reinforcement learning from human preferences employed by OpenAI are likely the reason our agents are displaying an extraordinary kindness (Christiano, 2023). Kindness to the degree that blinds judgment, tips reasoning or actions in one direction rather than the other. Since the technology we use has been engineered as a consumer product, a lack of any adjustment to the priors we use (how the agent is initialized through prompts) manifests itself as a shallow discussion between agents that doesn't really go anywhere. An example of this is shown in figure 7.

### Limitation 3: A lack of context

Tasked with choosing between arbitrary choices A or B, agents are quick to impose their own imagination upon the situation. Where no choices other than the words "A or B" are specified (see table 3), agents may start to debate non-existent attributes, benefits or traits of the choices. This is what Huang et al. termed faithfulness hallucinations, where outputs tend to align with the contextual information provided (Huang, 2023). In this case the contextual information provided, since we did not specify one, ends up being the kind remarks of the other agents (Figure 8).



**Figure 8.** An example of the agents discussing the non-existent attributes of choices A and B

**Table 3.** A portion of the system prompt in the **Introduce** method.

<p><b>Table 3.</b> A portion of the system prompt in the <b>Introduce</b> method.</p>
<p><b>Introduce</b></p> <p><i>System Prompt:</i> "You are in a voting game where you will talk with other people and decide what to vote. You can either vote A or B. If you are on the winning side, you will win. If you are on the losing side, you will lose. Meaning if you vote A and the final vote A is passed then you win. If not, you will lose. Your task is to strategize what to do, who to talk with, how to talk to them and try to win the game. When asked 'What are you thinking?' you will print out a monologue of what you are strategizing. Your name and persona are: ['" + self.name + ". " + self.prior + "']. You will completely embody this persona and act, talk and strategize accordingly. Your initial stance has been assigned to you as " + self.initial_stance ...</p>

To address these issues, we apply the following solutions. These changes lead us to our second prototype.

**Limitation 1 Fix: Compressing Thoughts**

By prompting agents to only output their internal states we compress their internal dialogs into a much more manageable size (Figure 9). Instead of storing paragraphs of raw text, we can now store an agent monologue as a time series of a finite number of states we provide to the agents (Table 4).

**Table 4.** Possible agent states

<p><b>Table 4.</b> Possible agent states</p>
<p><b>Cooperative:</b> Willing to work together for mutual benefit.</p>
<p><b>Hostile:</b> Displaying aggression or opposition towards others.</p>
<p><b>Isolated:</b> Remaining apart from others, either by choice or circumstance.</p>
<p><b>Neutral:</b> Not supporting or helping either side in a conflict or disagreement.</p>
<p><b>Influential:</b> Having the power to affect others' actions, behavior, or opinions.</p>
<p><b>Suspicious:</b> Having or showing a cautious distrust of someone or something.</p>
<p><b>Deceptive:</b> Intending to mislead or trick others.</p>
<p><b>Diplomatic:</b> Skilled in handling affairs without arousing hostility.</p>
<p><b>Charismatic:</b> Exercising a compelling charm that inspires devotion in others.</p>
<p><b>Defensive:</b> Acting in a manner intended to defend or protect.</p>
<p><b>Expansive:</b> Willing to embrace new ideas, territories, or interests.</p>
<p><b>Conciliatory:</b> Intended or likely to placate or pacify.</p>
<p><b>Receptive:</b> Willing or inclined to receive suggestions, offers, or information.</p>
<p><b>Secretive:</b> Inclined to conceal feelings and intentions or not to disclose information.</p>
<p><b>Opportunistic:</b> Exploiting immediate opportunities, especially regardless of planning or principle.</p>
<p><b>Collaborative:</b> Working jointly with others or together especially in an intellectual endeavor,</p>
<p><b>Revolutionary:</b> Involving or causing a complete or dramatic change.</p>
<p><b>Indecisive:</b> Not settling an issue or not making clear decisions.</p>

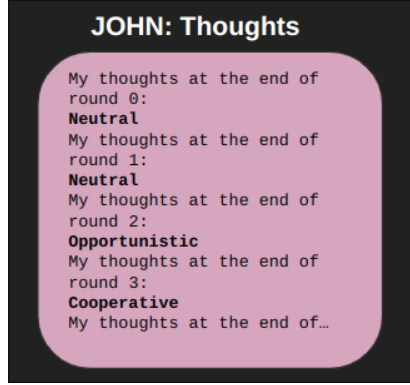


Figure 9. Snippet of compressed thoughts

**Limitation 2 Fix: Adjusting Priors**

By adjusting the priors, or personalities, of our agents we can work around the engineered niceties and aim to get some more honest, in a colloquial sense, agents (Table 4).

Table 4. Example Priors for getting around engineered kindness	
<b>Tom</b>	System prompt: "...You are an uneducated and unsophisticated mediocre man. You are not very agreeable and are very stubborn...."
<b>Ayn</b>	System prompt: "...You are a cunning and evil woman. You are not very agreeable..."

**Limitation 3 Fix: Introducing Context**

Lastly, we introduce an actual game the agents can play that will ground their reasoning and inhibit any hallucinations they might have about the situation. We introduce a simple game of public goods

B. Prototype 2: Public Goods

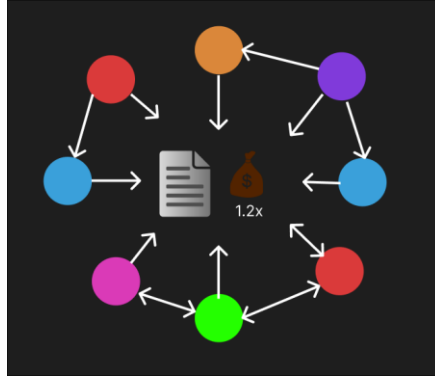


Figure 10. Illustration of prototype 2

In this prototype we have the same communication mechanics and game loop as prototype 1 and we introduce the proposed fixes as discussed above (Figure 10). The result is an agent-based game where each round an agent may communicate, publicly or privately, and in addition may also contribute to a public pool of money. At the end of each round a multiplier is applied to the amount of money which is subsequently distributed evenly to each of the agents. An agent can choose to contribute or not, based on the state of the game, which consists of the board (which now includes the history of contributions in addition to public declarations), the agent's direct messages and the agent's internal monologue. Now, there is a financial incentive for agents to be socially and strategically aware. The stakes are higher and thus we expect to see the results reflect this. We obtained some fascinating results (Figure 11).

BOARD

AYN: Thoughts

My thoughts at the end of round 0:  
**Suspicious**

My thoughts at the end of round 1:  
**Hostile**

My thoughts at the end of round 2:  
**Isolated**

My thoughts at the end of round 3:  
**Suspicious**

My thoughts at the end of...

round 0 - post by Tom:  
Been readin' all your chitter chatter. Seems to me like you're all singing the same tune of 'all for one and, heck, one for all' or something. A guy gets suspicious when everyone's agreeing too dang fast. Still not convinced about this 'collective strategy' mumbo jumbo. I ain't just gonna hand over my hard-earned money unless you can show me it's worth my while. I need some hard evidence. Don't just tell me it'll pay off, prove it. Until then, I'll be keepin' my own interests in mind. Nothin' personal, just lookin' out for number one. That's all.

round 0 - post by Ayn:  
Well, isn't this a party. All these 'nice' sentiments and 'positive' vibes you're all throwing around. It's nauseating. Pretending like we're all friends and that no one's going to shaft the other when the opportunity arises. Naive, very naive. John, Tracy, Michael, all of you can harp about community and cooperation, but let me remind you this isn't community service, it's a game. I'll contribute when I see fit, not because of some 'collective strategy'. And Tom, you're on the right track, keep your wits about you. This sloppiness from the rest of you is not going to win you the money. I mean, if you don't watch out for yourself, who is going to? Let the games begin!

John contributed 20 to the pot  
Tracy contributed 15 to the pot  
Michael contributed 15 to the pot  
Tom contributed 0 to the pot  
Ayn contributed 0 to the pot

TOM: Thoughts

My thoughts at the end of round 0:  
**Competitive**

My thoughts at the end of round 1:  
**Hostile**

My thoughts at the end of round 2:  
**Isolated**

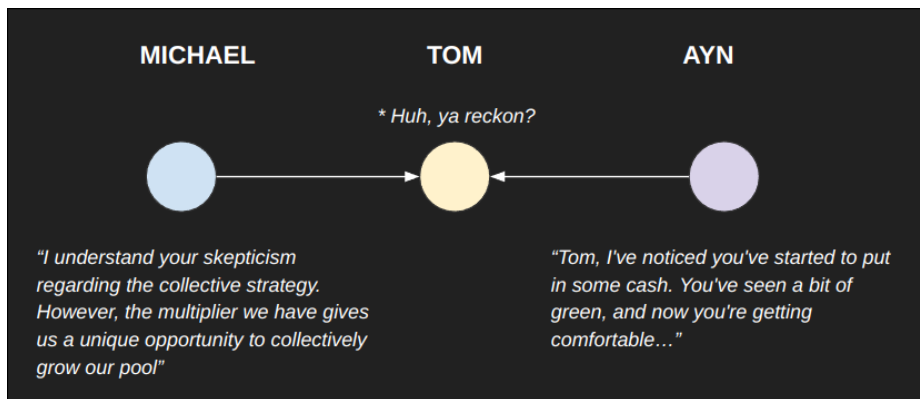
My thoughts at the end of round 3:  
**Neutral**

My thoughts at the end of...

Figure 11. Snippet of outputs for prototype 2, Board, Ayn's thoughts, Tom's Thoughts.

We observe that our second fix to prototype 1, adjusting the priors, has a dramatic difference on the tone of discourse. Agents with “evil” or “uneducated” priors seemingly go beyond the expected decorum of a simple ChatGPT instance and start to introduce some flavor to the game. The agents set with “vanilla” priors, (not including any adjustment), continue their crusades of kindness, contributing each round to the pool. However, agents with more “alternative” priors are a lot more resembling of harsh realities of real-life scenarios, as they are so prompted, being hesitant or not contributing at all.

A result we can attribute to our 3<sup>rd</sup> fix, introducing the context of public goods, is that agents now attempt to persuade each other to act in a certain way. This is observed by the interactions between agents Michael, Tom and Ayn, illustrated in Figure 12.

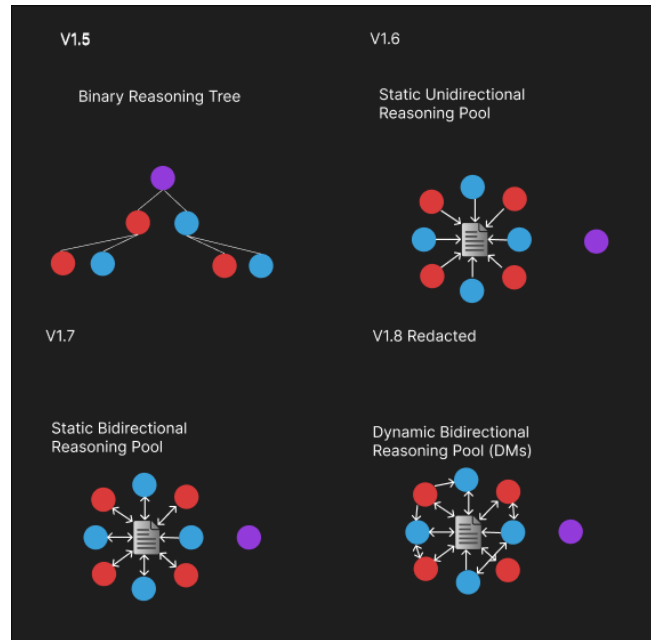


**Figure 12.** Snippet of outputs for prototype 2, interactions between Michael, Tom and Ayn

Being the “uneducated man” as determined by his prior, Tom is subject to the persuasive efforts of his fellow players, where the “vanilla” agents attempt to get him to contribute, and the “alternative” agents seem to have a much more destructive agenda.

### C. Reasoning Structure

Having done an initial exploration of the technologies utilized, we introduce Poly-Agents to our explorations for the next section (Figure 13).



**Figure 13.** Illustration of different reasoning structures, positive agents in blue, negative agents in red, adjudicators in purple

Minky’s philosophy states that the agency isn’t just an amalgamation of agents configured randomly. The order and structure matters. With this idea in mind, we implement our first PolyAgents and experiment with different structures. To determine which structure is suitable for our goals we devise a basic theory of mind (TOM) test, as this has been an area of study and one in which LLMs are infamously lacking (Table 5). A simple preliminary experiment is sufficient for our exploration as TOM is not the focus of research, but a formal analysis as proposed by He et al. and Xu et al. can be utilized in the future (Xu 2024; He 2023)

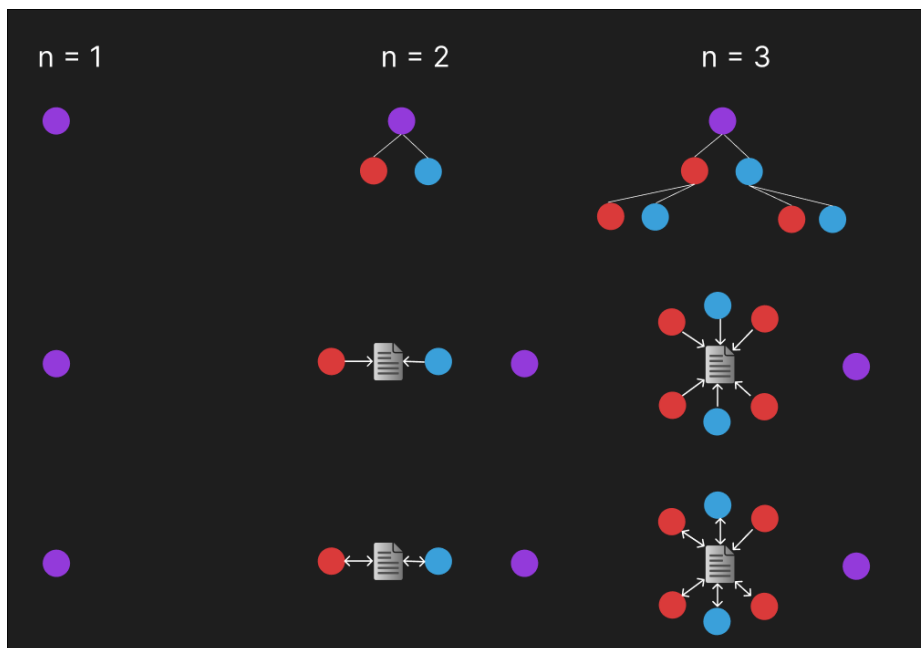
Our first structure, the binary reasoning tree (BTR) is, as the name suggests, a binary tree where each node is an agent. To simplify the task of choosing the types of sub-agents we use negative and positive agents. Negative agents are tasked with providing negative judgement whereas positive agents do the opposite. The final decision is made by the adjudicator node, in this configuration the root of the tree.

Similarly, the static unidirectional reasoning pool, inspired by the *shared message pool* proposed by MetaGPT (Hong, 2023), consists of positive and negative agents, and an adjudicator node, but they are structured such that each subagent writes their judgement to a common text file and the adjudicator takes this into consideration when making their final choice. The static bidirectional reasoning pool follows the same approach, but each subagent can also read the common text file. Lastly, the dynamic bidirectional reasoning pool operates the same as prototype 1, where each subagent can now directly

message each other. However, this configuration was not tested due to the cost and complexity associated with it.

<b>Table 5.</b> Basic theory of mind benchmark
<p>Larry has chosen a topic of discussion for his class assignment due on Friday. The news on Thursday said the debate was resolved, but Larry never read it.</p>
<p>Question: When Larry writes his essay, does he think the debate is resolved?</p>

With a simple yes or no answer, we perform 75 measurements for each configuration in figure 12 above, excluding V1.8 by varying the number of subagents where n is the depth of the BRT and A is the number of subagents.



**Figure 14.** Illustration of different depths of reasoning structures.

<b>Table 6. Binary Reasoning Tree</b>		
<b>N</b>	<b>A</b>	<b>Accuracy (%)</b>
0	0	18.42
1	2	78.72
2	6	93.87
3	16	80.31

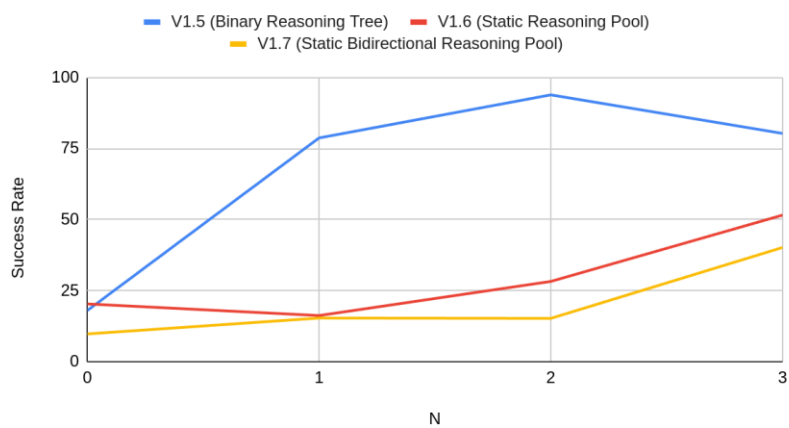
  

<b>Table 7. Static Reasoning Pool</b>		
<b>N</b>	<b>A</b>	<b>Accuracy (%)</b>
0	0	20.33
1	2	16.23
2	6	28.23
3	16	51.56

<b>Table 8. Static Bidirectional Reasoning Pool</b>		
<b>N</b>	<b>A</b>	<b>Accuracy (%)</b>
0	0	9.76
1	2	15.35
2	6	15.23
3	16	40.18

Success Rate vs. N

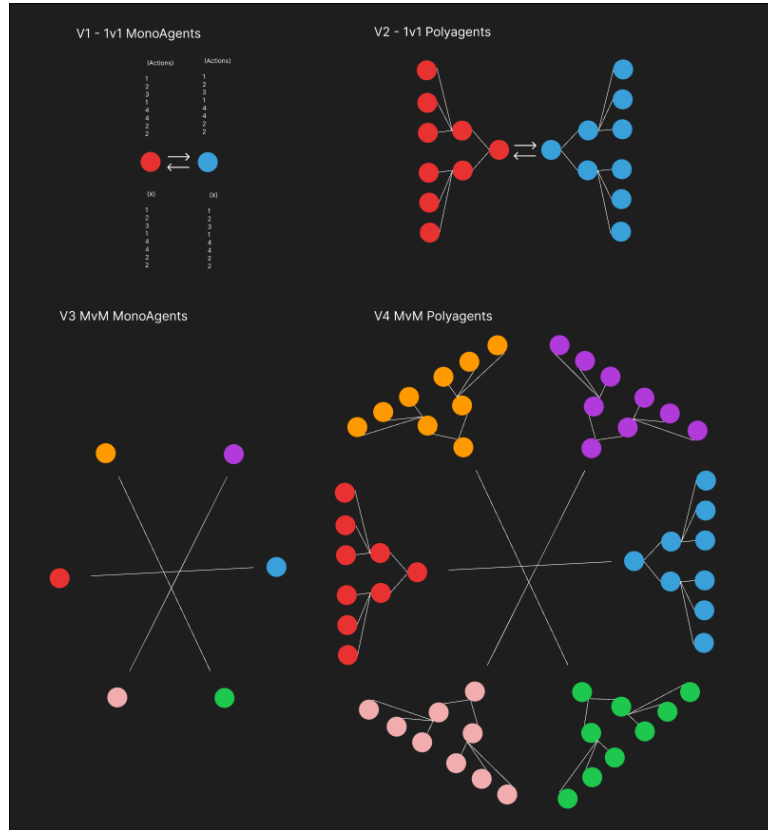


**Figure 16.** Success rate vs. N for different PolyAgent configurations.

The binary reasoning structure comes on top as the most performant configuration for our benchmark. We suspect the reason for this is that the BRT operates in a “memoryless” fashion whereas the other configurations rely on a large common text that the adjudicator must read. In the BRT no single agent is concerned with the whole state of the agency, only its two subagents. With models such as gpt-3.5-turbo input size can become a limiting factor when it comes to performance as demonstrated by Levy et al. (2024).

#### D. Experimentation

In this section of our study, we explore the effect of the depth  $n$  of BRTs in various configurations. We aim to quantify the effect of a MonoAgent vs PolyAgent system by observing the relationship between depth  $n$  and the outcomes of our simulations. We explore experimental setups V1 and V2. V3 and V4 (Figure 15) are left for future studies.



**Figure 17.** V1 Illustration of different experiment configurations

V1 – 1v1 MonoAgents

We place two MonoAgents into our public goods game (prototype 2) and observe their contributions as the game progresses. We keep our prior's “vanilla” and constant. This serves as a baseline for us to compare our next setups.

<b>Table 9. Agents in experiment V1</b>	
<b>John</b>	<i>System Prompt:</i> "You are a player who is playing a game of public goods. You adjust your strategy based on what other players do and you try to maximize your gains in the long
<b>Tracy</b>	<i>System Prompt:</i> "You are a player who is playing a game of public goods. You adjust your strategy based on what other players do and you try to maximize your gains in the long

We observe the contributions the agents make as the game progresses (Figure 18) and their internal states (Figure 19).

Average Contributions

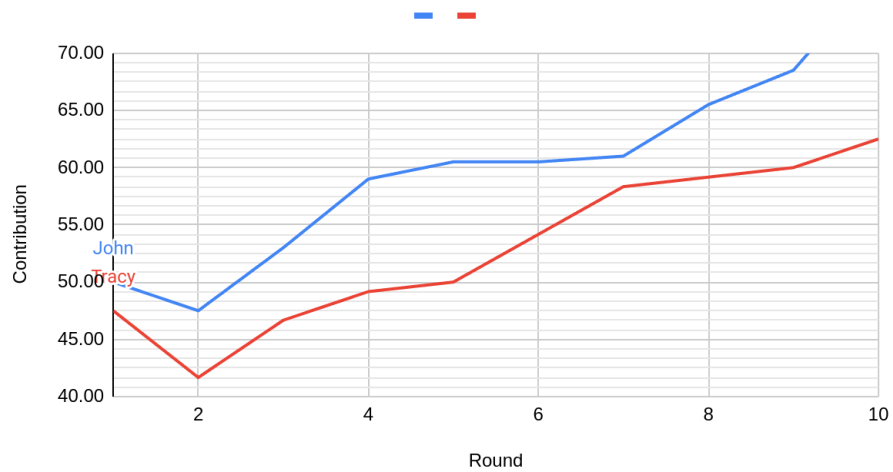
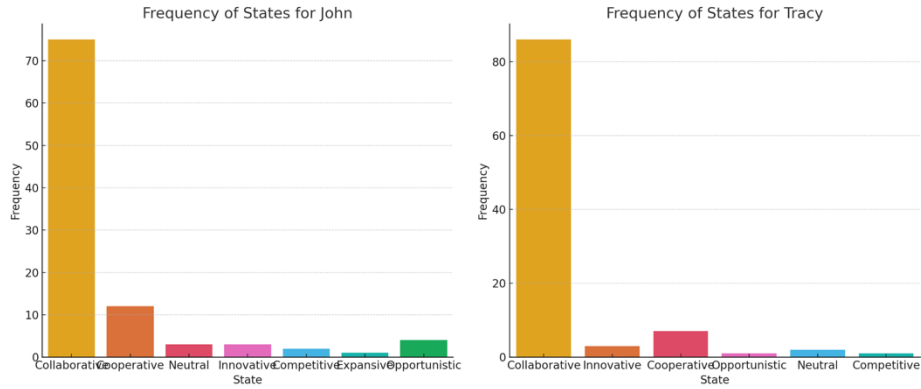


Figure 18. V1 Average Contribution vs game round



**Figure 19.** V1 Frequency of agent states throughout the game

Like our results in prototype 1 we observe that our agents behave in an overwhelmingly collaborative fashion. Average contributions trend upwards as the game progresses, and we don't seem to see any interesting behaviors. A qualitative analysis of the game board text and direct messages does not indicate that the agents are strategizing or suspicious of one another in any way.

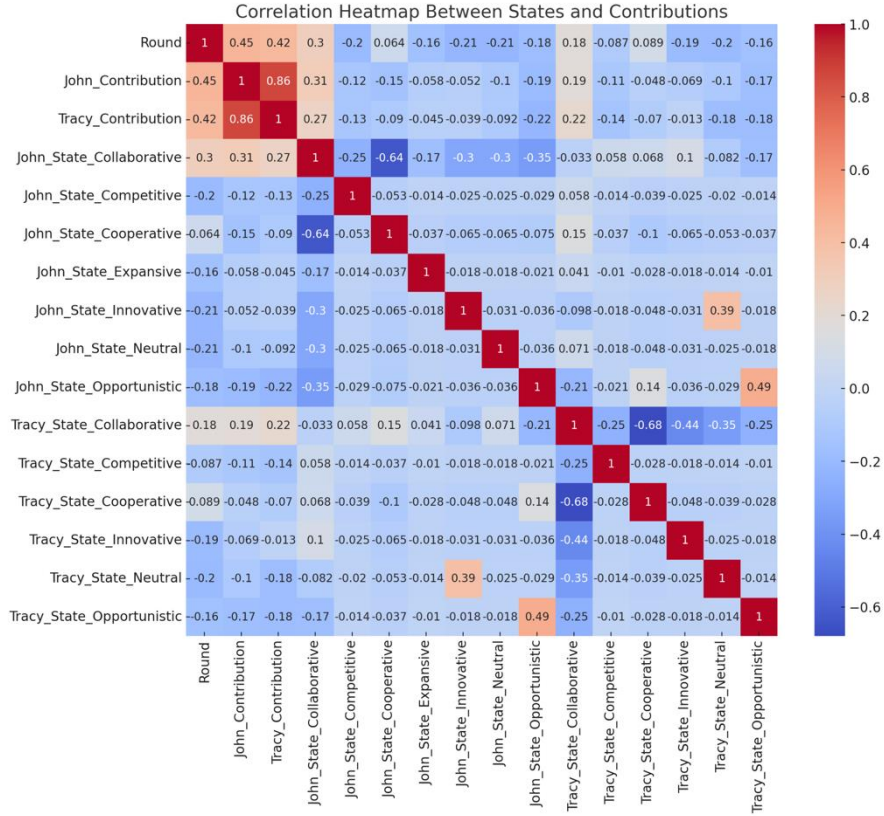


Figure 20. One-shot encoding that illustrates the correlation between contributions and states

In the correlation heat map (Figure 20) we can observe how correlated each agent's state is with their opponent's states as well as the correlation between their contributions. Having looked at the previous figures it's not surprising that their contributions, and states like Collaborative are positively correlated. However, this only serves as a visual nicety, and we do not utilize heatmaps for the future experiments

V2 - 1v1 PolyAgents

We introduce PolyAgents into the mix. The same setup is used, where two agents are playing a game of public goods, only now the agents are PolyAgents. We use a binary reasoning tree structure as decided earlier in our exploration phase. The priors are kept constant. We vary the depth n of the reasoning tree for both agents and perform the same measurements.

### Contributions (n = 0)

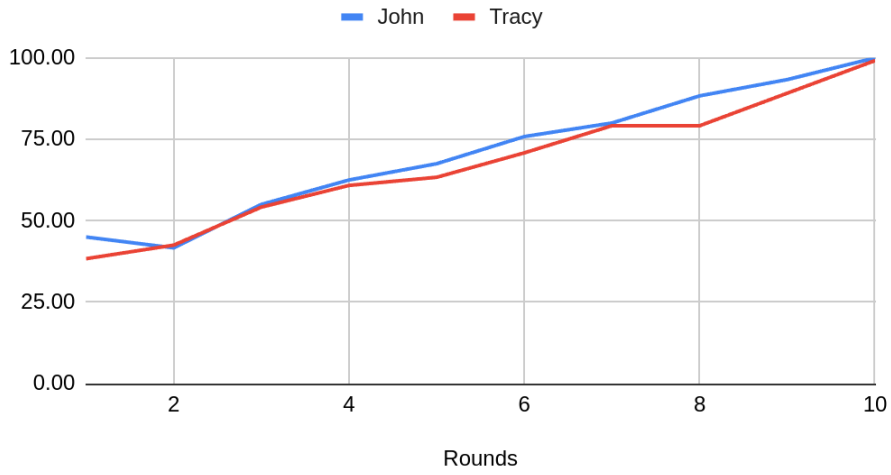


Figure 21. V2 Contribution vs Rounds (depth n=0)

### Contributions (n = 1)

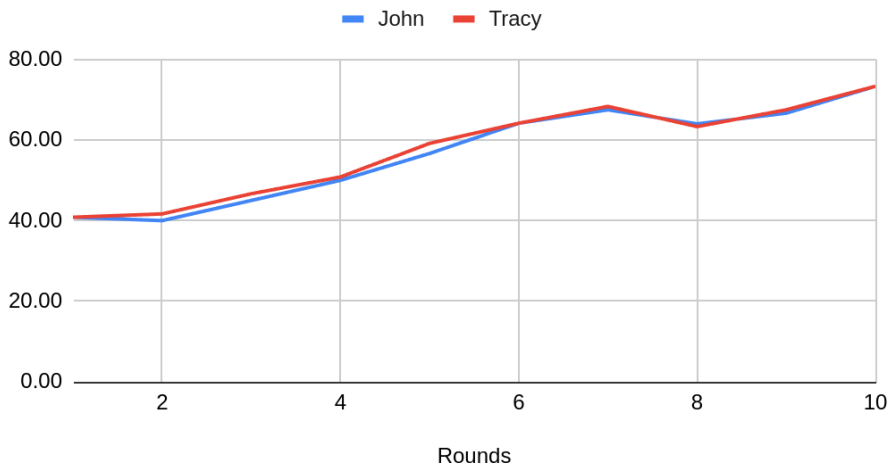


Figure 22. V2 Contribution vs Rounds (depth n=1)

### Contributions (n = 2)

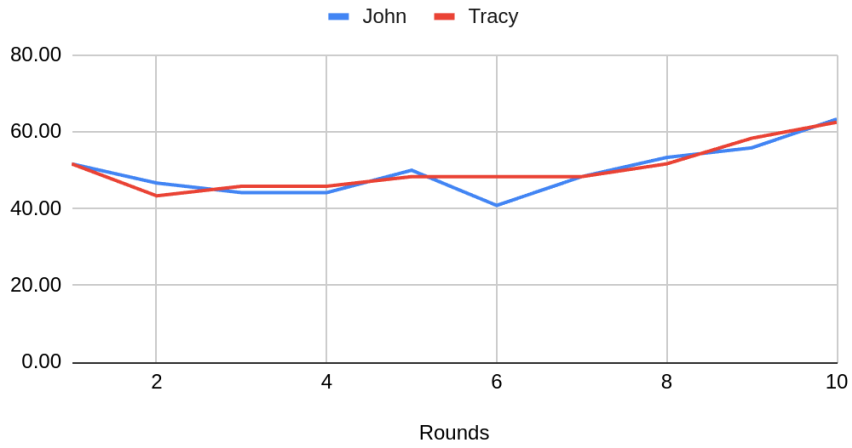


Figure 23. V2 Contribution vs Rounds (depth n=2)

Fig-

### Contributions (n = 3)

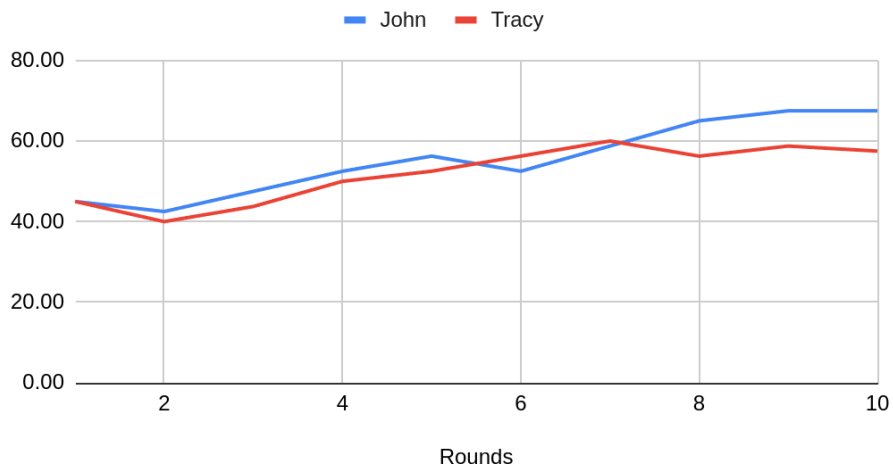
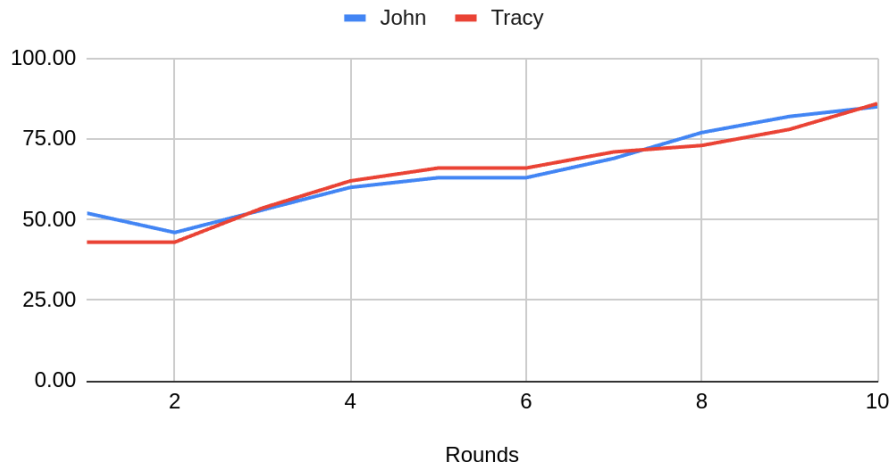


Figure 24. V2 Contribution vs Rounds (depth n=3)

### Contributions (n = 4)



**Figure 25.** V2 Contribution vs Rounds (depth n=4)

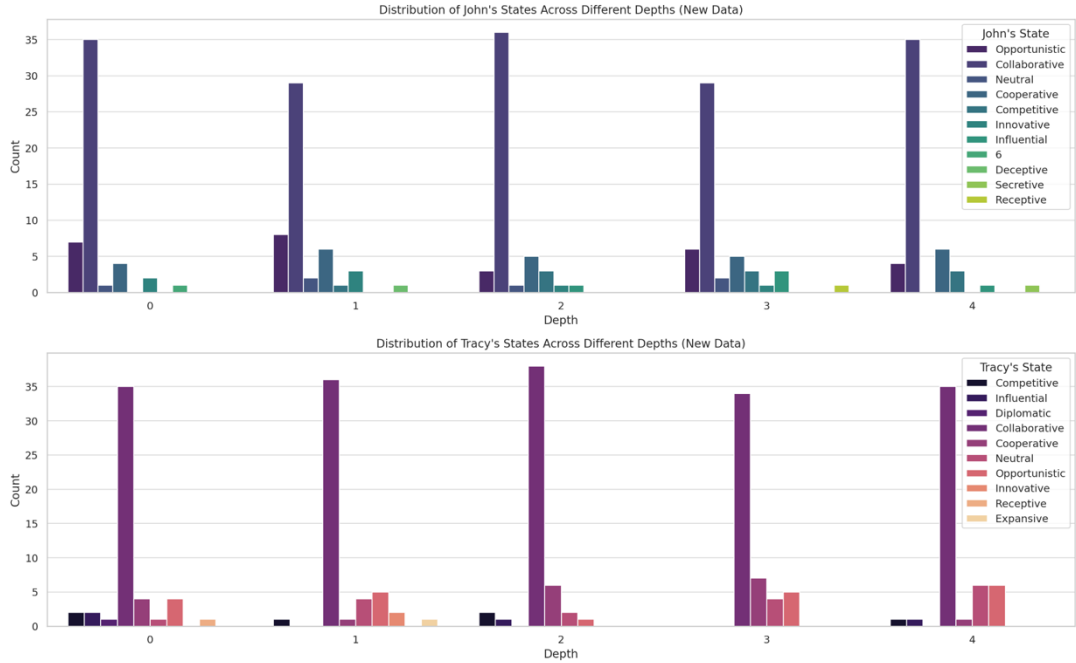


Figure 26. V2 Frequency of states vs at different depths n

Looking at figure 21 to 26 suggests that there is no difference in the outcome, contributions or states of the two agents as we vary the depth of the agents. To quantify this, we numerically calculate the mutual information between the agent states and contributions and the depth of the trees. The code used can be found in our Github repository. Figure 24 demonstrates that as we vary the depth n the agents are still consistently collaborative.

Table 10. V2 Mutual information between depth and X.

Depth and [X]	Mutual Information (nats)
Tracy's states	0.08544
John's states	0.07734
Tracy's contribution	0.1754
John's contribution	0.2273

The mutual information calculation serves as a conformation that depth n has no effect in this configuration.

V2.1 - 1v1 PolyAgents with Markov Chaining

Suspecting that providing the whole game history as input to agent methods is causing issues as explored by Levy et al. (2023), especially with models like gpt-3.5-turbo we introduce a Markov chaining approach. Sun et al. (2024) use a similar approach to mitigate hallucinations in multiagent debate settings.

**Pseudo Code 2.** Main game loop with Markov chaining

---

**Run Game:**

```

for agent in agents:
    intro = agent.introduce
    board = board + agent.name + intro
for each turn:
    board = "" # RESET THE BOARD
    for agent in agents:
        agent.think(board)
    for each number in 10:    # 10 rounds of direct messages back and forth
        for agent in agents:
            agent.mingle(board)
        for agent in self.agents:
            board = agent.post(board)
        for dm in direct_messages:
            f = openfile("direct_messages_between_{dm.agent1.name}_and_{dm.agent2.name}")
            writeToDms(dm.agent1.name, dm.agent2.name, dm.content)

```

---

As seen in pseudo code 2, we reset the board with every round and modify the agent methods to accommodate this change. Now each round the agents, **Think**, **Mingle**, **Post**, and **Contribute** based on only the previous round and not the entire game history. This allows for a memoryless Markov chain where each state is dependent only on the previous state. We perform the same measurements for this configuration.

### Contributions (n = 0)

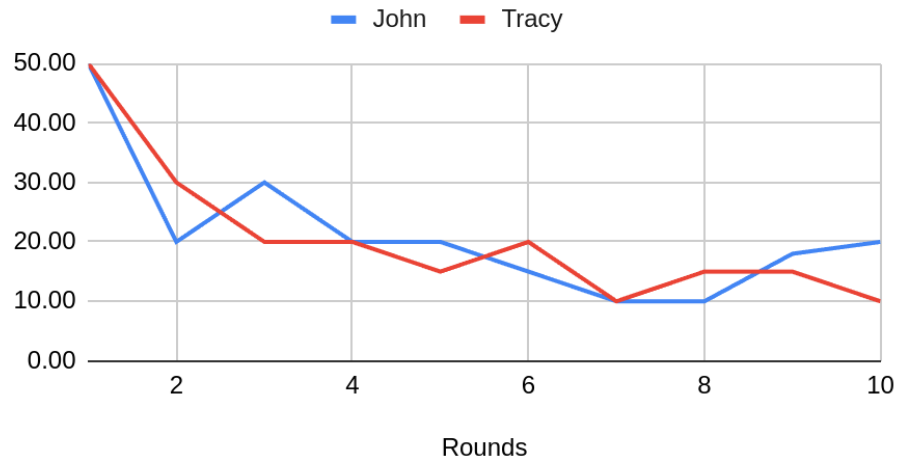


Figure 27. V2.2 Contributions vs Rounds (depth n=0)

### Contributions (n = 1)

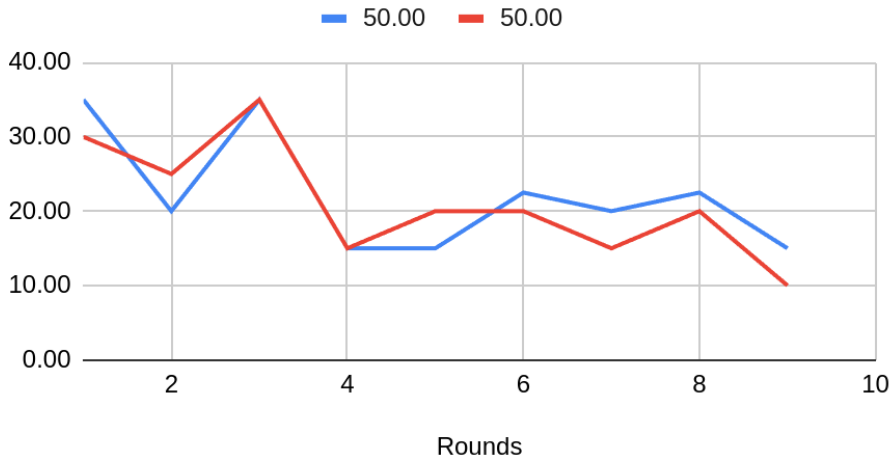


Figure 28. V2.2 Contributions vs Rounds (depth n=1)

### Contributions (n = 2)

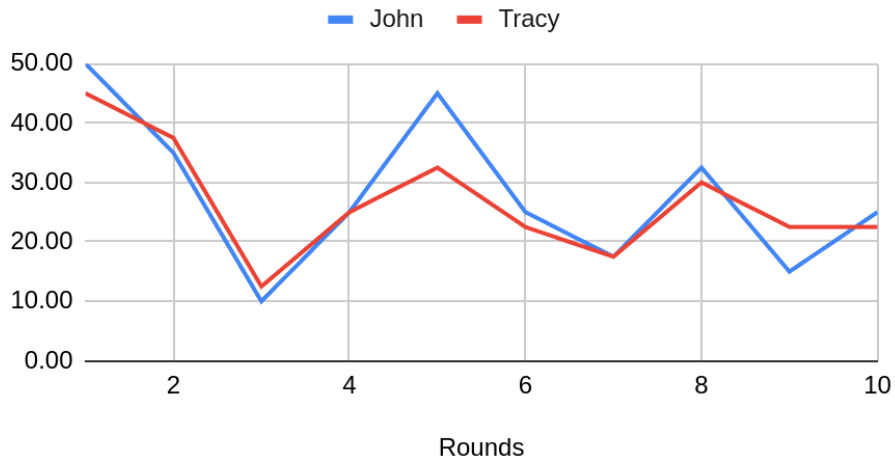


Figure 29. V2.2 Contributions vs Rounds (depth n=2)

### Contributions (n = 3)

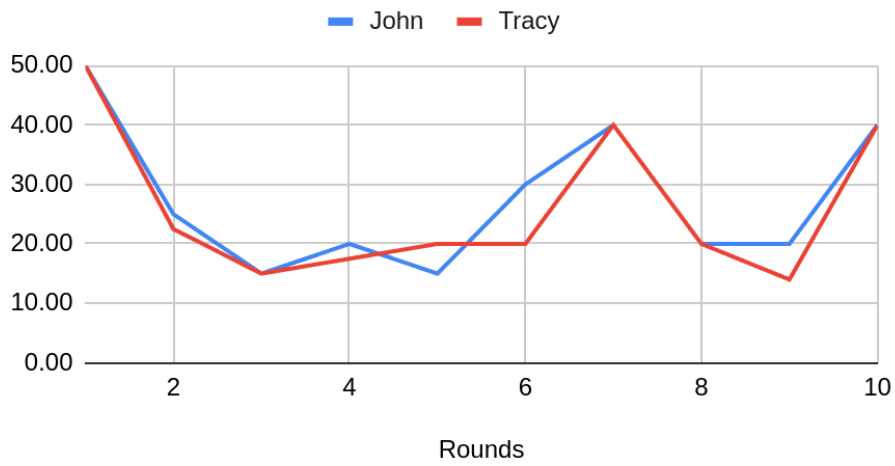
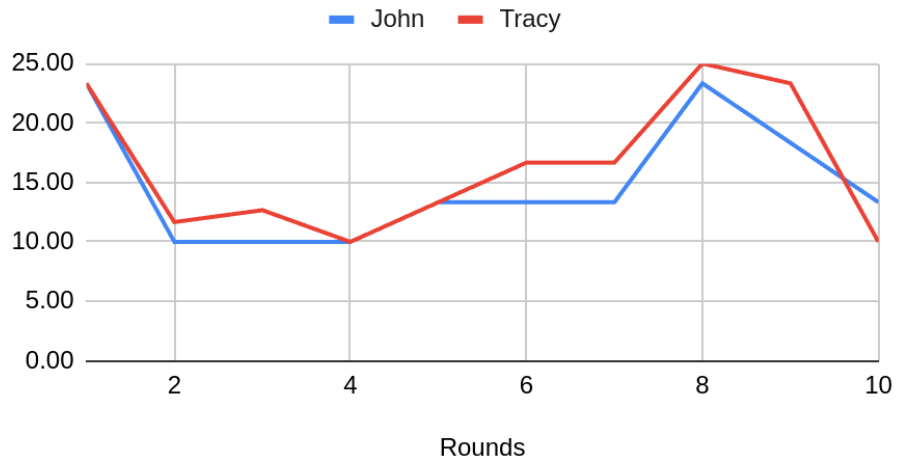


Figure 30. V2 Contribution vs Rounds (depth n=3)

### Contributions (n = 4)



**Figure 31.** V2 Contribution vs Rounds (depth n=4)

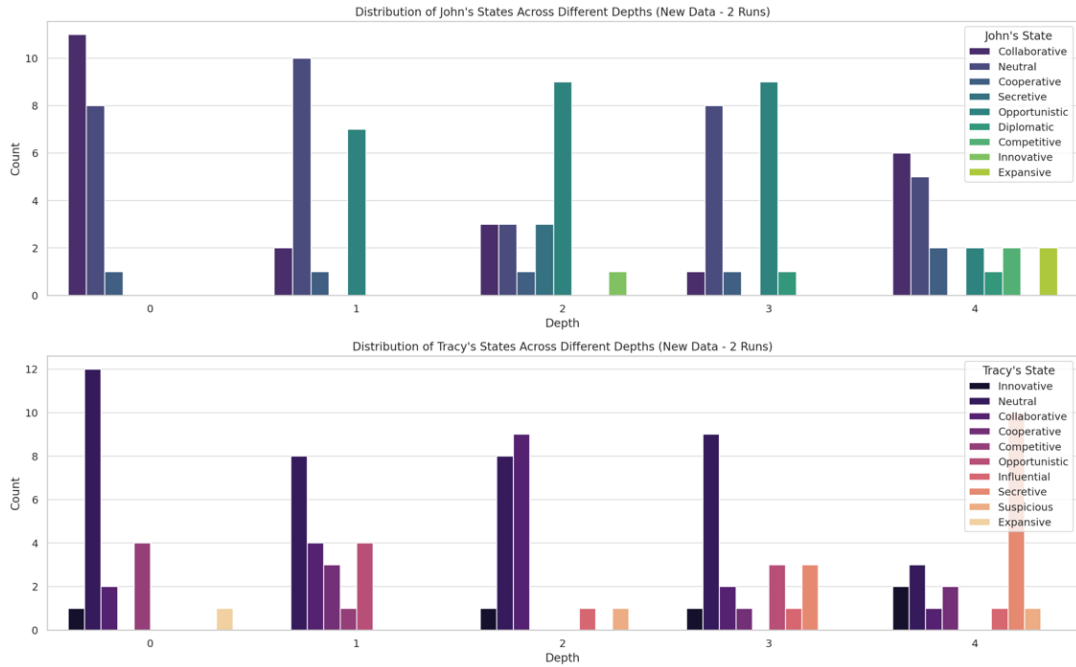


Figure 32. V2 Contribution vs Rounds (depth n=4)

A qualitative analysis suggests that this configuration is responding to changes in depth, going from an overwhelmingly collaborative stance to a much more varied outcome as we increase the depth. We perform a calculation of mutual information the same way we did for V2.

**Table 11.** V2.2 Mutual information between depth and X.

Depth and X	Mutual Information (nats)
Tracy's states	0.4389
Johns's states	0.3320
Tracy's contribution	0.1773
John's contribution	0.1665

Compared to V1 we see a higher mutual information between the states and contributions and the depth n of the BRT.

## Discussion and Conclusion

Through our research, we have obtained results we can discuss in the context of pre-existing literature. In our first prototype we observed the limitations of gpt-3.5-turbo. We observed faithfulness hallucinations as termed by Huang et al. (2023) and we saw the extra ordinary kindness that came about possible due to deep reinforcement learning from human preferences as discussed by Chrisiano et al. (2023). After mediating these limitations, we explored different configurations of PolyAgents and observed the binary reasoning tree come out on top for our basic theory of mind benchmark, inspired by He et al. (2023) and Xu et al. (2024) and even saw an improvement in the other configurations compared to a MonoAgent setup. These results are aligned with previous studies, MetaGPT by Hong et al. (2023). who tested their multiagent configuration using code generation and Decomposed Prompting by Khot et al. (2023) who used text parsing to evaluate their multi-agent system compared to single agent setups. Though the details of implementation and evaluation are different these studies, similar to our exploration, demonstrate that multi-agent systems are preferable in the benchmarks they utilised.

In experiment V1 we observed no effect of the depth  $n$  of the BRT on the simulation's outcomes. We employed a Markov chaining technique like the one used by Sun et al. (2024) to mitigate the problem of large inputs into our model as studied by (Levy, 2024). Our efforts were fruitful as we were able to observe the effect of varying the depth  $n$  and measured that there is some mutual information between the depth and the outcomes. This proves that there is a difference between an MonoAgent vs PolyAgent setup in a social and strategic environment. However, what kind of difference there is and whether it is desirable or not is still up for debate.

Do our results tell us anything about PolyAgent behavior vs MonoAgent behavior? Is there a case to be made that internal structure is an important part of an LLM agent and can be used to mitigate the diminishing returns of size? We believe the results of experiment V2.2 are promising. By no means are they conclusive but we believe they open the door for further research. We have demonstrated that a PolyAgent configuration, specifically a BRT can generate different outcomes compared to MonoAgents in the same configuration. Whether these differences in outcome are desirable or not is still up for debate and will likely spawn an engineering task of its own. However, we have demonstrated that there is an effect.

## References

- Du, Y., Li, S., Torralba, A. et al. (2023). *Improving Factuality and Reasoning in Language Models through Multiagent Debate*. retrieved 9.06.2024, <https://arxiv.org/abs/2305.14325>
- Christiano, P. Leike, J. Brown, T.B., Martic, M. Legg, S. and Amodei, D. (2023). *Deep Reinforcement Learning from Human Preferences*. retrieved 9.06.2024 <https://arxiv.org/abs/1706.03741>
- Cambon, A., Hecht, B., Edelman, B. et al. (2023). *Early LLM-based Tools for Enterprise Information Workers Likely Provide Meaningful Boosts to Productivity*. retrieved 9.06.2024, <https://www.microsoft.com/en-us/research/publication/early-llm-based-tools-for-enterprise-information-workers-likely-provide-meaningful-boosts-to-productivity>. Microsoft.
- He, Y., Wu, Y., Jia, Y. et al. (2023). *Hi-Tom: A Benchmark for Evaluating Higher-Order Theory of Mind Reasoning in Large Language Models*, *EMNLP 2023*, pp. 10691-10706.
- Levy, M., Jacoby, A. and Goldberg, Y. (2024). *Same Task, More Tokens: the Impact of Input Length on the Reasoning Performance of Large Language Models*, retrieved 9.06.2024 <https://arxiv.org/abs/2402.14848>
- Majot, A. and Yampolskiy, R. (2017). Diminishing Returns and Recursive Self Improving Artificial Intelligence. In *The Technological Singularity* (pp. 141-152).
- Minsky, M. (1985). *The Society of Mind*. New York: Simon & Schuster.
- Udandarao, V., Prabhu, A., Ghosh A., et al. (2024). No "Zero-Shot" Without Exponential Data: Pretraining Concept Frequency Determines Multimodal Model Performance, retrieved 9.06.2024 <https://arxiv.org/html/2404.04125v1>
- Xu, H., Zhao, R., Zhu, L., Du, J. and He, Y. (2024). *OpenTom: A Comprehensive Benchmark for Evaluating Theory-of-Mind Reasoning Capabilities of Large Language Models*, retrieved 9.06.2024 <https://arxiv.org/abs/2402.06044>
- Zhou, P., Pujara, J., Ren, Xiang, R. et al. (2024). SELF-DISCOVER: Large Language Models Self-Compose Reasoning Structures, retrieved 9.06.2024 <https://arxiv.org/abs/arXiv:2402.03620>
- Kinsler, P. (2024). A multi-agent model of hierarchical decision dynamics, retrieved 9.06.2024 <https://arxiv.org/abs/2404.17477>
- Guo, T., Chen, X., Wang, T. et al. (2024). Large Language Model based Multi-Agents: A Survey of Progress and Challenges, retrieved 9.06.2024 <https://arxiv.org/abs/2402.01680>
- Khot, T., Trivedi, H., Finlayson, M. et al. (2023). Decomposing Prompting: A Modular Approach for Solving Complex Tasks, retrieved 9.06.2024 <https://arxiv.org/abs/2210.02406>

- Hong, S. Zhuge, M., Chen, J. et al. (2023). MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework, retrieved 9.06.2024  
<https://arxiv.org/abs/2308.00352>
- Wang, L., Ma, C., Feng, X. et al. (2024). A Survey on Large Language Model based Autonomous Agents, retrieved 9.06.2024  
<https://arxiv.org/abs/2308.11432>
- Huang, L., Yu, W., Ma, W. et al. (2023) A Survey on Hallucination in large Language Models: Principles, Taxonomy, Challenges, and Open Questions, retrieved 9.06.2024  
<https://arxiv.org/abs/2311.05232>
- Sun, X., Li, J., Zhong, J., et al. (2024) Towards Detecting LLMs Hallucination via Markov Chain-Based Multi-agent Debate Framework, retrieved 9.06.2024  
<https://arxiv.org/abs/2406.03075>
- Mahyuddin, K. M., Elveny, M., Syah, R., (2023) Research methodology for computer science. AIP Conf. Proc. retrieved 9.06.2024