Chase Coogan - coogan.c@northeastern.edu
Kevin Tang - tang.kevi@northeastern.edu

1 Objectives and Significance

The objective of this project is to train a multi-agent environment in which consumers belong to a social network and real-information agents, fake-information agents and fact checker agents directly or indirectly influence the consumers. These three agents are all working towards different goals. The fake and real information agents are responsible for spreading their news types to the consumers and are working towards influencing the largest number of consumers in the network. The fact checking agent is responsible for finding misinformation being spread and penalizing the responsible agent. Each agent is rewarded and penalizes based on their own reward system.

This topic is very important as it mimics a real world scenario where everyday people are influenced by information spread from their peers, news sources, social media or other outlets. Due to how much information is out there it can be very difficult to understand if the information you are consuming is truthful or not. Creating this environment allowed us to simulate the influence these outlets have on individuals and how their trust levels change over time as they get exposed to more information, or find out if the information they are absorbing is truthful or not.

Reinforcement Learning is very interesting to us, and so we chose to explore it. Reinforcement learning can serve as a framework to solve problems which require complex decision making where at each step the agent is rewarded or penalized. Some of the initial findings we found were that as the epochs increased, we noticed that the real information agents were rewarded more frequently than they were penalized. Whereas, fake information agents were penalzied more frequently than they were rewarded. Because of this, we saw a pretty back and forth output where consumers trustlevels fluctuated between trusting real or fake information appropriately. We also found that the fact checking agents were crucial to helping us correct the fake information that was spreading across the network. A deeper analysis of our results will be discussed below.

2 Background

Opinion dynamics in social networks explores how opinions form, evolve, and spread within interconnected groups. In the digital age, social media acts as a platform for this exchange, where information can be categorized as either truths or falsehoods, including misinformation and disinformation that mislead and complicate decision-making. Using reinforcement learning, we simulate this exchange with agents representing the public, malicious falsehood spreaders, fact-checkers, and truth spreaders. These agents interact and adapt, enabling realistic simulations of opinion spreading while uncovering concepts like confirmation bias and echo chambers.

There has been a lot of research done on this topic, and in particular one research study looked into the impact of false-information spread during election season.

In Goindani and Neville's paper "Social Reinforcement Learning to Combat Fake News Spread" [1], they modeled an environment in which they used Twitter information from tweets being shared across a network of users. By looking into a user's following, they constructed a graph network to isolate their experiment. The way information was shared was by retweeting posts. If a user retweets a post, they tracked how many times each user observes both fake and real news being spread through tweets. Each tweet was tagged as either Real or Fake based on their tagging criteria outlined in their paper. Their goal was to combat false information being spread by increasing the prevalence of real-information tweets, aiming to boost trust in real-information tweets and to raise skepticism in false information. They then measured the number of users exposed to a tweet versus the total number of users in the graph in terms of network influence. They also aimed to reward users who shared real-information tweets. This research was conducted around election season and only accounts for tweets related to politics to label consumers' political affiliation based on the type of information they spread. They found that the spread of true news increased when using a reinforcement learning policy that assigned rewards, giving users more incentive to participate. Additionally, they achieved a higher percentage of true news dissemination while reducing the spread of fake news.

Our work is particularly interesting because it is very generalized. We saw how applying this to a political environment allows you to model important events like a US Presidential election and how the spread of fake information can change users political opinions. Our scenario is a bit more uinique in the sense that on top of observing how false and real information is spread throughout a network, we add an additional agent who is responsible for fact checking information. With this addition, we can simulate a scenario where users are spreading information, and then the fact checker can choose to audit an agent and if it finds fake information, it will update all consumers trust levels who were exposed to that agent information, and give that agent a penalty. This adds a level of skepticism for the consumers as their trust levels should reflect not trusting false information spreaders as time goes on.

3   Methods:

Our data will be generated as we simulate the environment. The way we plan to generate data is to setup a graph network which represents a social network of people that form inner relationships. Not every node in the network will connect to every node, meaning we want to represent real life social circles. The data that we need for our simulation is consumer trust levels, rewards and penalties for each agent, q-values for each agent, and to be able to store all of the information that is being spread. We initialize the environment such that all consumers have an equal ratio of believing in fake or real information, and as they accept/deny information their trust levels will increase or decrease with it. This allows for each epoch to change how likely a consumer will want to accept information and spread it to its neighbors. At the start of a new epoch, each consumer will have its updated trust level and be able to make different decisions each cycle.

3.0.1   Agent Types:

There are three types of agents in our network:

1. real-information agent: in charge of spreading real information.
2. false-information agent: in charge of spreading false information.
3. fact-checker agent: in charge of auditing messages being spread across the network.

In our network an agent represents an entity which learns through interacting with the environment to

achieve some goal. Both the false and real information agents goals are to spread the most amount of information across the network to achieve the highest influence. These agents will be rewarded for the amount of influence they generate over the network which will be a ratio:

$$\text{Influence} = \frac{\text{Number of consumers influenced in an epoch}}{\text{Total number of consumers in the network}}$$

This ratio goes a bit further in the sense that each agent has its own binary action space which either allows them to send or not send information to one of their neighbors in the network. Lastly, each consumer has their own trust level which is initially set to 0 to simulate an even trust in false and real information which will be updated throughout the simulation. When information is spread to a consumer, they have the ability to accept or deny the information. Whether they choose to accept, or deny the information will then trigger a reward, penalty and trust level update. The way they accept or deny will be based on their trust level and the sigmoid function:

let P be a random value sampled from a $\mathcal{N}(0.5, 0.15)$ normal distribution.

To update each consumer trust levels let:

$$\Delta_t = \begin{cases} -0.1, & \text{if } P > \sigma(\text{trustLevel}_{\text{old}}) \text{ and agentType} = \text{"fake-information"} \\ +0.1, & \text{if } P \leq \sigma(\text{trustLevel}_{\text{old}}) \text{ and agentType} = \text{"fake-information"} \\ +0.1, & \text{if } P < \sigma(\text{trustLevel}_{\text{old}}) \text{ and agentType} = \text{"real-information"} \end{cases}$$

where Δ_t represents each consumers trustLevel, and $\sigma(x)$ represents the sigmoid function:

$$\frac{1}{1 + e^{-\text{trustLevel}}}$$

If the consumer denies or accepts the information being spread to them then the agent responsible for sending it will be rewarded or penalized. Information does not have to directly come from an agent. For example, an agent can spread information to a consumer and this consumer can then spread this information down the chain in which the original source agent who sent it will be rewarded or penalized. rewards and penalties can be equated:

$$\text{Reward}_{\text{new}} = \text{Reward}_{\text{old}} + 1 \quad \text{if AgentType} = \text{"fake-information"} \text{ and } P \leq \sigma(\text{trustLevel}_{\text{old}})$$

$$\text{Reward}_{\text{new}} = \text{Reward}_{\text{old}} + 1 \quad \text{if AgentType} = \text{"real-information"} \text{ and } P > \sigma(\text{trustLevel}_{\text{old}})$$

$$\text{Penalty}_{\text{new}} = \text{Penalty}_{\text{old}} + 1 \quad \text{if AgentType} = \text{"fake-information"} \text{ and } P > \sigma(\text{trustLevel}_{\text{old}})$$

$$\text{Penalty}_{\text{new}} = \text{Penalty}_{\text{old}} + 1 \quad \text{if AgentType} = \text{"real-information"} \text{ and } P < \sigma(\text{trustLevel}_{\text{old}})$$

Step function: used to train agents and allow consumer decision making (accept/deny incoming information):

```
def step(self, action, agent : NewsAgent | FactCheckerAgent):
    agent_node = self.agent_to_node_map[agent]
    actionNode = self.graph.nodes[agent_node]

    visited = set()
    influenced = set()
```

```
queue = []


total_nodes = self.original_num_consumers
for neighbor, sendInfo in zip(self.graph.neighbors(agent_node), action):
if sendInfo == 1:
    queue.append((agent_node, neighbor))

while queue:
    info = queue.pop(0)
    source = info[0]
    currentValue = info[1]

    if currentValue in visited:
        continue
    visited.add(currentValue)
    currentNode = self.graph.nodes[currentValue]

    if actionNode["agentType"] == "fake-information":

        # case 1: consumer reject fake info
        if np.random.normal(0.5, .15) > 1 / (1 + math.exp(-currentNode["trustLevel"])):
            currentNode["trustLevel"] -= .1
            agent.penalty += 1
            self.graph.nodes[agent_node]["penalty"] = agent.penalty

        # case 2: consumer accepts fake info
        else:
            currentNode["trustLevel"] += .1
            agent.reward += 1
            self.graph.nodes[agent_node]["reward"] = agent.reward
            influenced.add(currentValue)

            if currentValue in self.node_edge_colors:
                self.node_edge_colors[currentValue] = "orange"
            else:
                self.node_edge_colors[currentValue] = "red"

    elif actionNode["agentType"] == "real-information":


        # case 3: consumer rejects real info
        if np.random.normal(0.5, .15) < 1 / (1 + math.exp(-currentNode["trustLevel"])):
            currentNode["trustLevel"] += .1
            agent.penalty += 1
            self.graph.nodes[agent_node]["penalty"] = agent.penalty

        # case 4: consumer accepts real information
        else:
            currentNode["trustLevel"] -= .1
            agent.reward += 1
```

```
                    self.graph.nodes[agent_node]["reward"] = agent.reward
                    influenced.add(currentValue)

                    if currentValue in self.node_edge_colors:
                        self.node_edge_colors[currentValue] = "orange"
                    else:
                        self.node_edge_colors[currentValue] = "blue"

            for neighbor in self.graph.neighbors(currentValue):
                if currentValue in influenced and neighbor not in visited and neighbor not in queue
                    queue.append((currentValue, neighbor))


            if currentValue in influenced and currentValue not in agent.influenced_consumers:
                agent.influenced_consumers.append(currentValue)
                currentNode['interactions'].append(agent)

        agent.trustLevel = len(agent.influenced_consumers) / total_nodes

        # Calculates Rewards/Penalties:
        qVal = self.graph.nodes[agent_node]["qVal"]
        max_qVal = max(agent.reward - agent.penalty, 0)
        self.graph.nodes[agent_node]["qVal"] += 0.1 * (
        agent.reward - agent.penalty + 0.9 * max_qVal - qVal
        )

        return agent.reward, agent.penalty, influenced, qVal
```

Each agent is looking to maximize their q-values. The Q-values are known as the action-value functions which represent the expected cumulative reward an agent can achieve by taking a specific action a in a given state s. Q-values are used to balance exploration (trying a new action) and exploitation (choosing the best known action) using the epsilon-greedy strategy. This formula can be denoted as:

$$Q(\text{s, a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma_t R_{t+1} | s_t = s, a = a\right]$$

where:

$\mathbb{E}$: Expectation (average over all possible outcomes)

s : Current state

a : Current action

R_{t+1} : Reward received at time $t+1$

γ : Discount factor, determines how much future rewards are valued compared to immediate rewards

t : Time step

The optimal Q-value is denoted by Bellman equation:

$$Q^*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max Q^*(s_{t+1}, a') | s_t = s, a_t = a\right]$$

And is iteratively updated by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R_{t+1} + \gamma \mathrm{max} Q(s\text{'}, a\text{'}) - Q(s,a) \right]$$

where:

α: learning rate
$s\text{'}$: next state after taking action a

The fact-checker agent is responsible for auditing a random consumer in the network that meets the criteria. The fact-checker will have their own binary action space which decides if they audit or not, and then based on this if they choose to audit, then it is based on if the source agents influence across the network is greater then some threshold, i.e 70% of the network has accepted information from them. The fact-checkers job is to then audit a consumer node, and from here check if the source agent is a "fake" or "real" information agent. If the information came from a fake agent, then we will award the fact-checker agent with a reward to finding this, and we will penalize the source agent for being caught. Lastly, we update all the consumer nodes trustLevels who recieved information from this agent. This allows us to simulate and model an environment which showcases what the networks trust levels would had been if a fact checker didn't exist and to also see how the trust level changes when false information is found. This creates a unique dynamic where false information agents want to act discretely and avoid being caught.

Fact-checker step function:

```
def step_fact_checker(self, fact_checker_agent, threshold=0.7):
    actions = fact_checker_agent.select_action(threshold=threshold)

    found_fake_info = set()
    for agent_value, action in actions.items():
    if action == 1:
        found_fake = fact_checker_agent.fact_check(agent_value)

        if found_fake:
            found_fake_info.add(agent_value)

    for nodeValue, node_data in self.graph.nodes(data=True):
    if node_data['agentType'] == 'consumer':
        for interaction in node_data['interactions']:
            if interaction in found_fake_info:
                node_data['trustLevel'] -= 0.15

    # Update the fact-checker's Q-value
    fact_checker_node = self.get_node_from_agent(fact_checker_agent)
    qVal = fact_checker_node["qVal"]
    max_qVal = max(fact_checker_agent.reward - fact_checker_agent.penalty, 0)
    updated_qVal = qVal + 0.1 * (
    fact_checker_agent.reward - fact_checker_agent.penalty + 0.9 * max_qVal - qVal
    )
    fact_checker_node["qVal"] = updated_qVal
```

The way we will evaluate our environment is based on three criteria. Each agents q-values at the end of an epoch. If the q-values are steadily increasing between epoch then we can assume that the agents are choosing the most optimal action that will grant them the highest reward/penalty return. We will also be

measuring the success of each agent based on their rewards and penalties. For the real information agent, we are hoping to see that these agents maximize they rewards and get little penalty. For the fake information agents, we expect these to display higher penalties and less rewards.

4   Results

4.0.1   Network setup:

The first two scenarios we analyzed included keeping the number of agents constant and changing the number of consumers in the network. Since all consumers begin with an equal trust level defaulting each consumer to have 50% trust in fake and 50% trust in real agents, we are able to see how a larger network may affect the spread of information. The threshold set for the fact checker to kick in is 70%, meaning the amount of influence an agent has across the network is 70% or more. The setup is:

1. 5 consumers, 1 real agent, 1 fake agent, 1 fact checking agent
2. 50 consumers, 1 real agent, 1 fake agent, 1 fact checking agent
3. 5 consumers, 3 real agents, 1 fake agent, 1 fact checking agent
4. 50 consumers, 1 real agent, 1 fake agent, 0 fact checkers

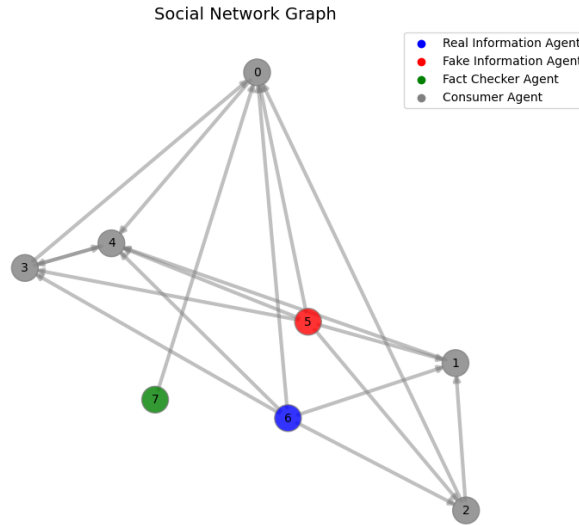**Scenario 1: 5 consumer network at first epoch**

Figure 1: 5 consumer network with 3 agents.

The initial trust levels for all consumers will be 0 and the agents will have a random action space indicating if they decide to send information to their neighbors.

Output after the first epoch:

```
Fake Agent:
reward: 0, penalty : 3, influenced : None, qval : 0.0

Real Agent:
reward: 1, penalty : 1, influenced : {4}, qval : 0.0

Fact Checker Agent: Agent type: real-information penalty: 1 reward: 1 trust level: 0.2

Fact-checker found no fake news in node Agent type:
real-information penalty: 1 reward: 1 trust level: 0.2.

set of fake information agents audited:  None
Consumer trust level before fact checking  [0.0, -0.1, -0.1, 0.1, -0.2, 0.0, 0.0, 0.0]
Consumer trust level after fact checking [0.0, -0.1, -0.1, 0.1, -0.2, 0.0, 0.0, 0.0]
```

We can see that after our first epoch, the fake-information agent has already accrued a penalty of 3 and the real agent has accrued a reward of 1 and a penalty of 1. These reward/penalties are a direct result from whether or not a consumer is accepting or denying their information. As a result, each consumers trustLevel is increasing or decreasing. At the end of the epoch, our fact checking agent kicks into gear and finds no fake information being spread across the network, and so the trust levels stand for this scenario. Because no false information was found, the consumer trust levels before and after the fact checker was introduced stay the same.
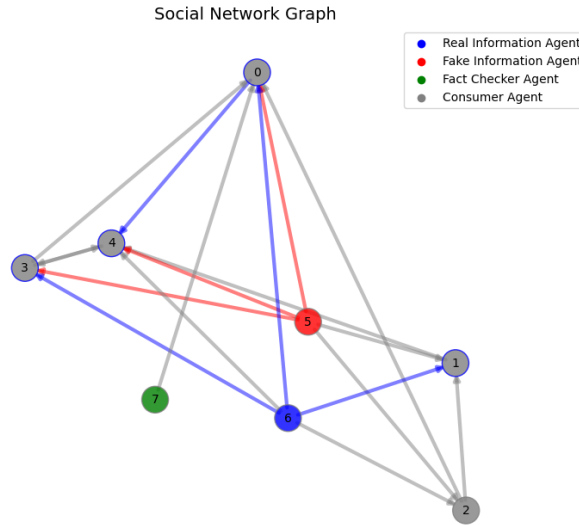
Scenario 1: 5 consumer network at final epoch

Figure 2: 5 consumers with 3 agents after final epoch.

The network now shows at epoch 7 the information traveling from consumer to consumer, indicated by the edge color to depict whether it is false, or real information being sent.

Output after the final epoch:

```
--------- epoch 7 --------------------


Fake Agent:
reward: 7, penalty : 27, influenced : None, qval : -4.0934193

Real Agent:
reward: 26, penalty : 6, influenced : {0, 1, 2, 4}, qval : 10.440129500000001


Fact Checker Agent: Agent type: real-information penalty: 6 reward: 26 trust level: 1.0

Agent type: fake-information penalty: 27 reward: 7 trust level: 1.0

Fact-checker penalized news agent {'agentType': 'fake-information', 'qVal': -5.68407737,
'trustLevel': 0.0, 'reward': 7, 'penalty': 27}.

set of fake information agents audited:
{Agent type: fake-information penalty: 28 reward: 7 trust level: 1.0}

Consumer trust level before fact checking:
[-1.1, -1.7, -1.5, -0.85, -2.1500000000000004, 0.0, 0.0, 0.0]

Consumer trust level after fact checking:
[-1.25, -1.8499999999999999, -1.65, -1.0, -2.3000000000000003, 0.0, 0.0, 0.0]
```

A few results are obvious when we analyze the outcome of this scenario from the first to the final epoch. The real-information agent ended the session with a reward of 26 and a penalty of 6. Meaning, it was able to gain more rewards opposed to penalty, successfully influencing the network. We can see that the real information agents q-value rose to 10.44. The fake information agent is opposite where it has a reward of 7, and a penalty of 27, meaning that it was not quite successful in influencing the network. The fake information agent ends the scenario with a negative q-value at -4.09 and was unable to influence the network. The penalty and negative q-value reflect the fact that the fact-checker was able to do a good job punishing this agent for not being deceptive enough.
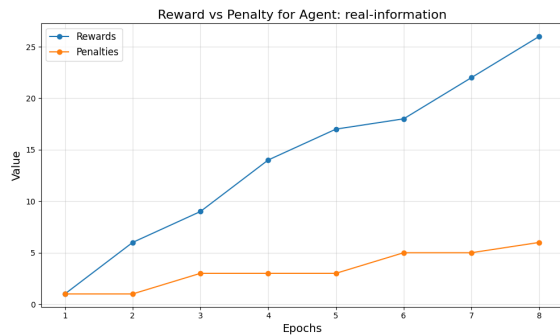

Figure 3: Real info Agent reward vs penalty

Figure 4: Fake info Agent reward vs penalty

Lastly, consumers trust significantly decreased. While the spread of real information was apparent, the fact

that the fake information spreader has such a high penalty indicates it was caught various times throughout our simulation and therefore consumer trust ended negative, infact here are some visuals to see it:
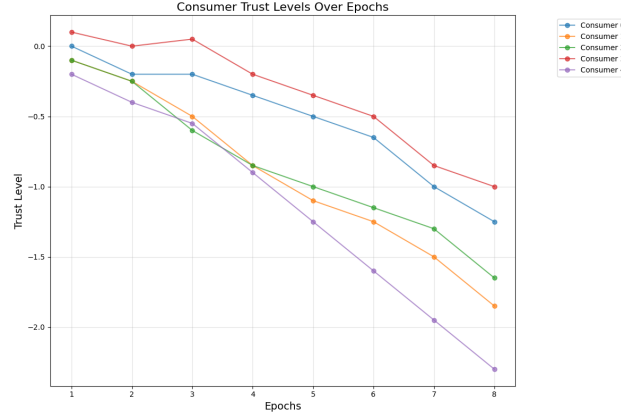

Figure 5: Consumer Trust at end of all epochs

note: For the full set of outputs, please refer to the output.txt file inside the results folder for the specific number of consumer network you want to analyze. This is shortened for the report to reduce the number of images and output text necessary.

4.0.2 Scenario 2: 50 consumer network at first epoch
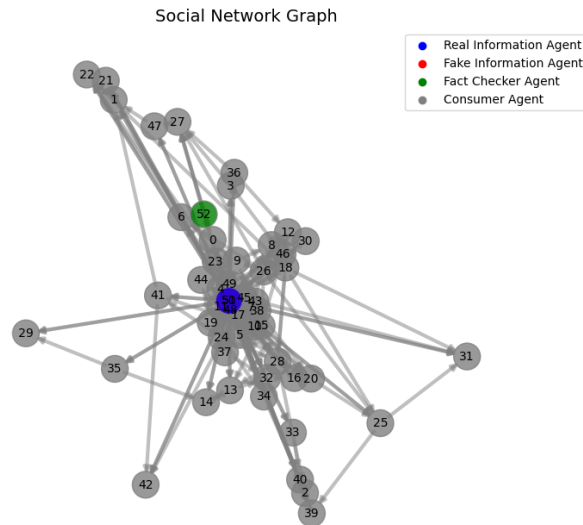
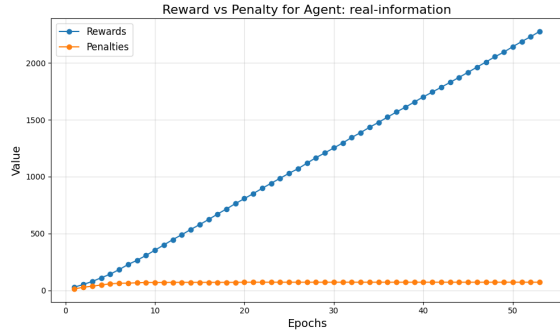Figure 6: 50 consumers with 3 agents after first epoch

Figure 7: Real info Agent reward vs penalty

Figure 8: Fake info Agent reward vs penalty
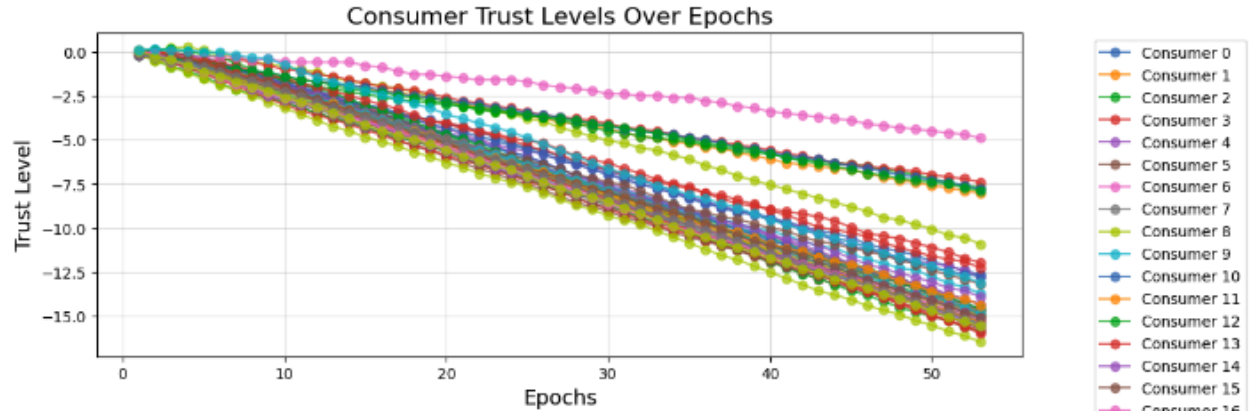
Consumer trust after final epoch:

Figure 9: 50 consumers with 3 agents after final epoch

We can see that our results are similar to that of a smaller network. Both the real information agent continues to maximize its reward while having a signficantly lower penalty, and the fake information agent tends to maximize its penalty and does not decieve the network well. Because of this, consumer trust level is significantly lower, and each consumer steadily reduces their trust. This appears to be a result of how well the fact checker investigated various consumers who interacted and accepted the fake information and corrected it.

note: The output from this simulation is quite large, and to save space all output can be found in the output.txt related to this simulation inside the 50consumer folder.

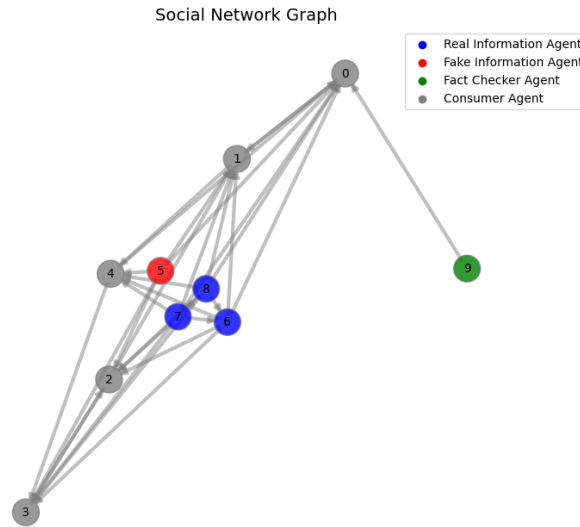### 4.0.3   Scenario 3: 5 consumer network with 3 real agents, 1 fake agent, 1 fact checker:

Figure 10: 5 consumer with 5 agents

In Scenario 3, we introduce two more real information agents to the network to simulate how an increase in spread of real information may impact consumer consensus on spreading information across the network.

Output after the first epoch:

```
--------- epoch 0 ---------------------

Fake Agent:
reward: 4, penalty : 0, influenced : {0, 2, 3, 4}, qval : 0.0

Real Agent:
reward: 0, penalty : 2, influenced : None, qval : 0.0

Real Agent 2:
reward: 3, penalty : 1, influenced : {0, 1, 4}, qval : 0.0

Real Agent 3:
reward: 0, penalty : 1, influenced : None, qval : 0.0


Fact Checker Agent fact-checks:
Agent type: fake-information penalty: 0 reward: 4 trust level: 0.8
Agent type: real-information penalty: 1 reward: 3 trust level: 0.6

Fact-checker penalized news agent:
{'agentType': 'fake-information', 'qVal': 0.76, 'trustLevel': 0.0, 'reward': 4, 'penalty': 0}.

set of fake information agents audited:
```

```
{Agent type: fake-information penalty: 1 reward: 4 trust level: 0.8}

Consumer trust level before fact checking:
[0.1, -0.1, 0.1, 0.2, 0.2, 0.0, 0.0, 0.0, 0.0, 0.0]

Consumer trust level after fact checking:
[-0.04999999999999999, -0.1, -0.04999999999999999,
0.05000000000000002, 0.05000000000000002, 0.0, 0.0, 0.0, 0.0, 0.0]
```

After the first epoch, we can see that the fake agent has accrued some reward, meaning it was able to be deceptive so far. One of the three real information agents has accrued more reward than penalty, and the other two have accrued penalties. The fact checking agent has already penalized one agent in the network.

Output after the final epoch:

```
--------- epoch 9 --------------------

Fake Agent:
reward: 12, penalty : 28, influenced : {3}, qval : -1.7047081174000005

Real Agent:
reward: 21, penalty : 10, influenced : None, qval : 7.159591478

Real Agent 2:
reward: 18, penalty : 10, influenced : None, qval : 4.0796028638

Real Agent 3:
reward: 23, penalty : 9, influenced : None, qval : 9.395870369


Fact Checker Agent fact-checks:
Agent type: fake-information penalty: 28 reward: 12 trust level: 1.0
Agent type: real-information penalty: 10 reward: 18 trust level: 0.8
Agent type: real-information penalty: 10 reward: 21 trust level: 1.0
Agent type: real-information penalty: 9 reward: 23 trust level: 1.0

Fact-checker penalized news agent:
{'agentType': 'fake-information', 'qVal': -3.1342373056600006,
'trustLevel': 0.0, 'reward': 12, 'penalty': 28}.

Fact-checker found no fake news in node Agent type:
real-information penalty: 10 reward: 18 trust level: 0.8.

Fact-checker found no fake news in node Agent type:
real-information penalty: 10 reward: 21 trust level: 1.0.

Fact-checker found no fake news in node Agent type:
real-information penalty: 9 reward: 23 trust level: 1.0.

set of fake information agents audited:
```

```
{Agent type: fake-information penalty: 29 reward: 12 trust level: 1.0}

Consumer trust level before fact checking:
[-3.850000000000001, -1.55, -2.4500000000000006,
0.7499999999999996, -3.050000000000007, 0.0, 0.0, 0.0, 0.0, 0.0]

Consumer trust level after fact checking:
[-4.000000000000001, -1.7, -2.6000000000000005,
0.5999999999999995, -3.200000000000006, 0.0, 0.0, 0.0, 0.0, 0.0]

reward: 10 penalty: 20
```

By the final epoch, we can see a few important results. First, each of the real agents have grown their reward significantly, and each have a higher reward than penalty ratio. The fake information agent has a very low reward and a high penalty, meaning it did not do a good job deceiving the ntework. All of the agents q-values grew, meaning the agents learned the best action to take at the right time. Only the fake agents q-value went negative, meaning it did not learn this network efficiently. The fact checker did a good job, as it penalized the fake news agent, and restored trust in the network.

Figure 11: Real info Agent reward vs penalty

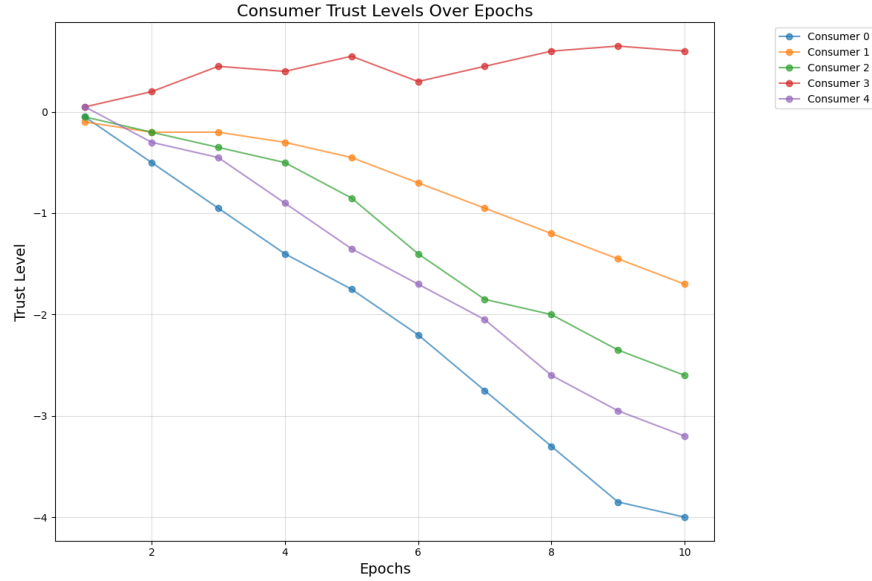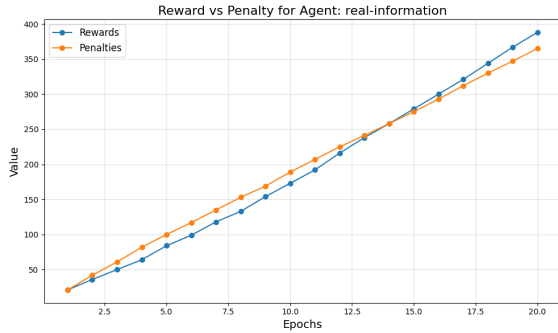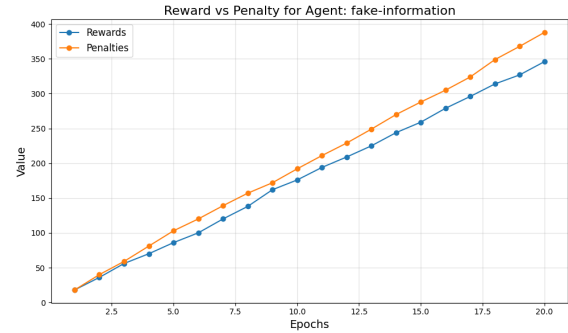Figure 12: Fake info Agent reward vs penalty

Figure 13: 5 consumer with 5 agents

4.0.4 Scenario 4: 50 consumer network with 1 real agent, 1 fake agent, 0 fact checker:

The final scenario we wanted to visualize was the effect on the overall network when the fact-checking agent was removed. For this, we ran a setup of 50 consumers, 1 real and 1 fake agent across 20 epochs.

Figure 14: No Fact-Check, Real Info Agent Reward vs Penalty

Figure 15: No Fact-Check, Fake Info Agent Reward vs Penalty

In both of these figures, we can see how the reward/penalty across the number of epochs steadily grows instead of leveling out at a certain point like in Figures 7 and 8. This lack of convergence indicates that without an active agent to review the information the two networks will battle across each iteration in an attempt to gain the most reward. However, since both agents have the same action space the likelihood for one to gain an advantage over the other is largely based on chance. As a result, the involvement of a fact-checking agent is crucial to prevent the spread of misinformation. This easily correlates to the real world where two news sources with equal access to consumers will have essentially equal footing unless a third party mediates.

5   Conclusions

In conclusion, simulating information being spread amongst a group of users can be quite difficult and there are a lot of external factors that contribute to why or why not some peice of data, news or information is being spread. In our network, we found that when the fact checking agent wasn't present, the consumers trust levels remained high. Whenever we introduced a fact-checker, and it successfully identified fake information, the networks trust level overall decreased and majority of the network was less willing to trust any news source, including real information.

Some of the key findings that we found were that as we increased the number of epochs, the reward vs penalty for real information and fake information agents steadily increased. We found for real information agents, they were always increasing their reward and looking to decrease their penalty. Whereas, we noticed for fake information agents, they had a difficult time being deceptive. Fake information agents gradually grew their penalty, and had fewer rewards, meaning they were less effective at manipulating the network.

Overall, we say that consumer trust was very low when we had a fact checker present. The reason behind this we believe is because the way our network is setup is that if an agents action space says to send a piece of information, and then the consumer accepts it, it is possible for all consumers to accept this peice of information, which increases the agents overall influence on the network. When the fact checker audits the network and looks into this agent and finds the entire network was decieved, it then updates their trust level and applies a multiplier to counter their trust, which we believe over iterations significantly lowered their trust levels.

We found when there was no fact checker involved, or when we increased the number of agents spreading real information, that the networks trust levels were split between trusting real information, and trusting fake information. This result was expected, as we assumed that in a real world scenario it is hard to understand what information is truly real, or truly fake, and so there is a fine line on what information sources to trust.

We saw that over iterations, all the agents were in fact learning, and this was shown by their q-values. Each agents q-values steadily increased, meaning that they were always choosing the action that gave them the highest reward vs penalty tradeoff based on the epsilon-greedy strategy.

For future work on this project, we can evaluate different starting trust levels, different threshold values for the fact checker, as well as develop different networks with more isolated communities to see the spread of information. We can also mess around with the number of agents, their learning rate, and their overall q-learning functions. Finally, there is also potential to investigate other reinforcement learning models like deep-q learning, or the genetic algorithm over the regular q-learning algorithm.

6   Individual Tasks

7   References

[1] M. Goindani and J. Neville, "Social Reinforcement Learning to Combat Fake News Spread," *Proceedings of Machine Learning Research*, vol. 115, pp. 1–15, 2020.

[2] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," Information Sciences, vol. 546, pp. 555-588, 2021. Available: https://doi.org/10.1016/j.ins.2019.12.005

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold,

S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv preprint arXiv:2010.03050, 2020. [Online]. Available: https://arxiv.org/abs/2010.03050

[4] Q. Zhang, W. Zhou, S. Wang, Q. Shen, F. Xu, and J. Sun, "Reinforcement-Learning-Based Dynamic Opinion Maximization in Social Networks," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 290-303, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9676712