Simulating Misinformation Spread

using Monte Carlo methods

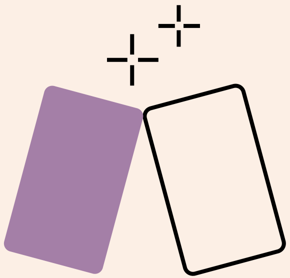
By Akanksha Agrawal and Sara Kiel

Misinformation Significance

- Misinformation could happen quickly and with no ill intent. In the hours after breaking news, many may draw conclusions or share information without enough context.

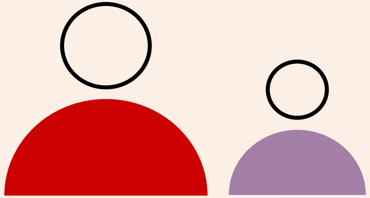
- Social media and digital news are where many people today get their news. It is increasingly difficult to discern authentic information from inaccurate information online.

Project Overview



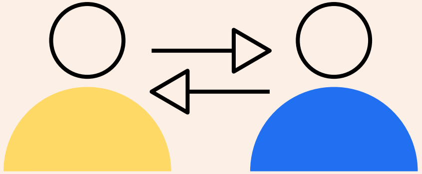
Simulation Goal

Analyze the dynamics of misinformation spread and contrast it with factual news propagation in synthetic social networks.



Agent-Based Model

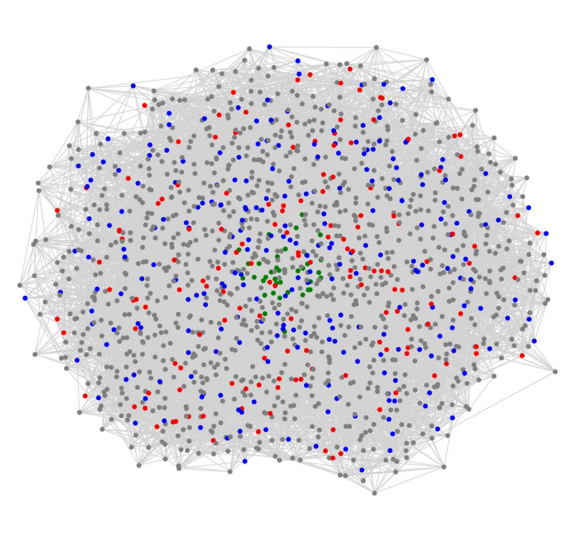
Each social media user is modeled as an agent with roles like influencer, fact-checker, or susceptible, with these roles impacting their sharing behaviors.



Network Construction

Hybrid network using Watts-Strogatz for local clustering of close-connections and Barabási-Albert for global influencer hubs, mimicking real-world social platforms.

Network Design



Role Assignment

Agents assigned as fact-checkers, normal, susceptible, or influencers based on degree and randomization. Influencers are also assigned 1 of the first 3 roles.

Susceptibility Types

Susceptible users are further categorized as normal, highly susceptible (5–10%), or super-spreaders (0.1%).

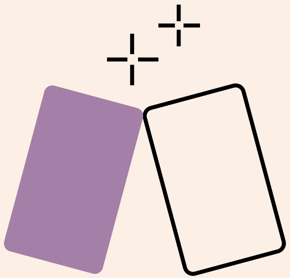
Community-Based Trust

Trust levels assigned to edges based on community proximity: higher within-cluster, lower across-cluster links.

Assumptions

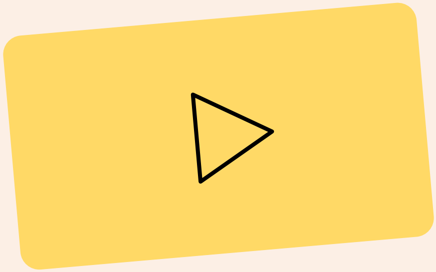
1. Content of the news
2. Trust with indirect connections
3. Relationship between the fake and real news
4. etc.

Code Architecture



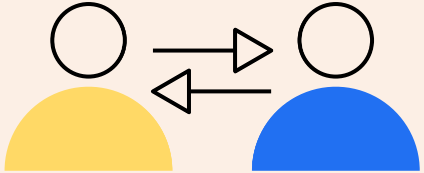
Modular Design

Divided into components: agents, network generator, news item classes, and configuration.



Configuration Management

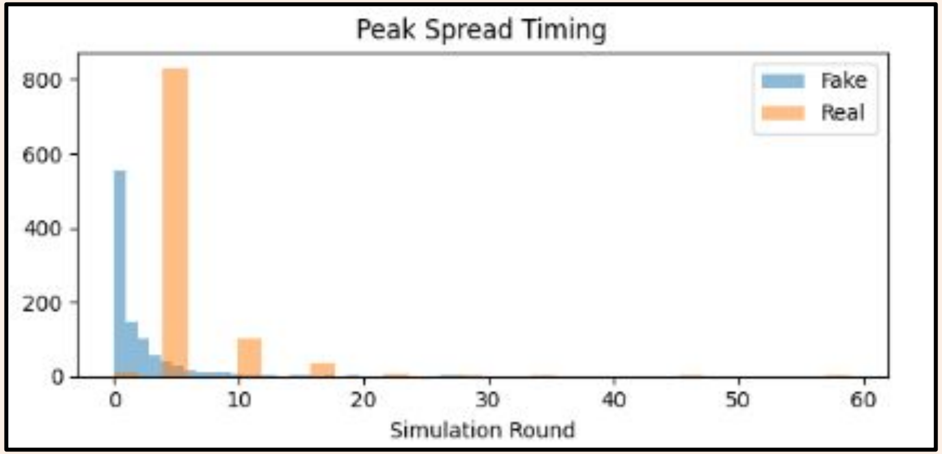
Parameter centralization in `config.py` enables easy tweaking for experiments and baselines.



Monte Carlo Engine

`main.py` runs large-scale simulations with randomized setups for statistical robustness.

Simulation Mechanics



Time-Stepped Spread

News diffuses in discrete rounds, with agents sharing based on probabilistic thresholds modulated by trust and role.

Delay Distributions

Fake news shares follow short delays (mostly 1 step), while factual news delays are significantly longer (up to 18 steps).

Trust-Weight Transmission

Sharing probability is scaled by the trust level on each edge, initially assigned according to intra- vs inter- community connections, simulating personalized credibility filters.

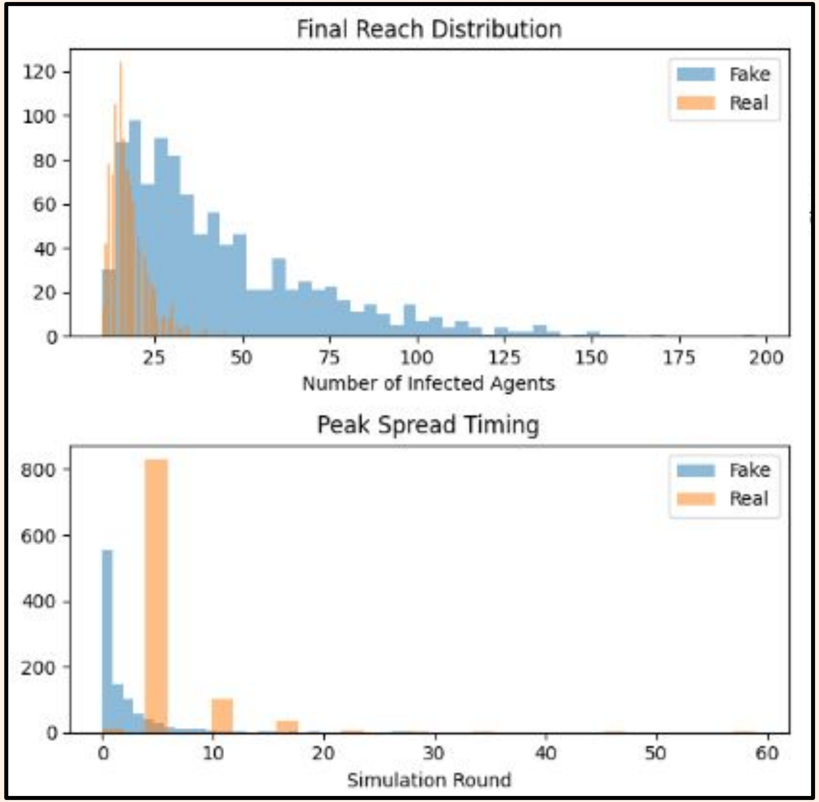
Fact-Checker Intervention

Fact-Checkers have a chance of labelling a fake news item as fake, impacting their neighbors' probabilities of sharing.

Baseline Results

```
Baseline Statistics (1,000 runs):
Fake News - Avg Reach: 43.7 ± 30.1
Real News - Avg Reach: 18.0 ± 6.1
Fake Peak Round: 0.0 (IQR 0.0-2.0)
Real Peak Round: 5.0 (IQR 5.0-5.0)
Fake Final Believers: 43.6 ± 30.1
Real Final Believers: 18.0 ± 6.1
```

MIT Study Claim	Our Results	Validation Status
Fake news spreads 6× faster	Fake peaks 8× earlier than real	Confirmed
"P_fake ≈ 1.7× P_real	P_fake (0.11–0.22) vs P_real (0.06–0.09) ≈ 1.7–2.4× ratio	Aligns with MIT's 1.7× ratio
Fake news reaches broader audiences	Fake reach 3× higher than real	Confirmed
Unpredictable "bursts" in misinformation	High standard deviation in fake news reach is observed	Confirmed - real-world unpredictability



Hypotheses

Fact-Checker Efficacy

H0: Adjusting the proportion of fact-checkers in the network has no significant effect on the final spread of misinformation.

Influencer Dynamics

H0: Influencers' sharing behavior has no special impact on outcomes.

1. Influencer controlled seeding
2. Influencer delay boost
3. Influencer trust boost

Competing News

H0: Introducing related factual news and incorporating competitive interference into a network has no impact on the final spread of misinformation.

Experimental Insights

Hypothesis 1 Results:

Fact-Checkers %	Avg Fake Reach	Avg Real Reach
10%	43.0 ± 29.4	18.0 ± 5.9
30%	46.4 ± 31.4	17.4 ± 5.4
50%	42.8 ± 30.5	17.6 ± 5.7
80%	44.3 ± 31.1	18.1 ± 6.0

```
Baseline Statistics (1,000 runs):
Fake News - Avg Reach: 43.7 ± 30.1
Real News - Avg Reach: 18.0 ± 6.1
Fake Peak Round: 0.0 (IQR 0.0-2.0)
Real Peak Round: 5.0 (IQR 5.0-5.0)
```

```
Influencer Counts Fake: 80.6 ± 38.2
Influencer Counts Real: 22.1 ± 8.7
Influencer Total Fake: 80.6 ± 38.2
Influencer Total Real: 22.1 ± 8.7
```

Experimental Insights

Hypothesis 3 Statistics (1,000 runs):

Fake News - Avg Reach: 44.2 ± 31.6

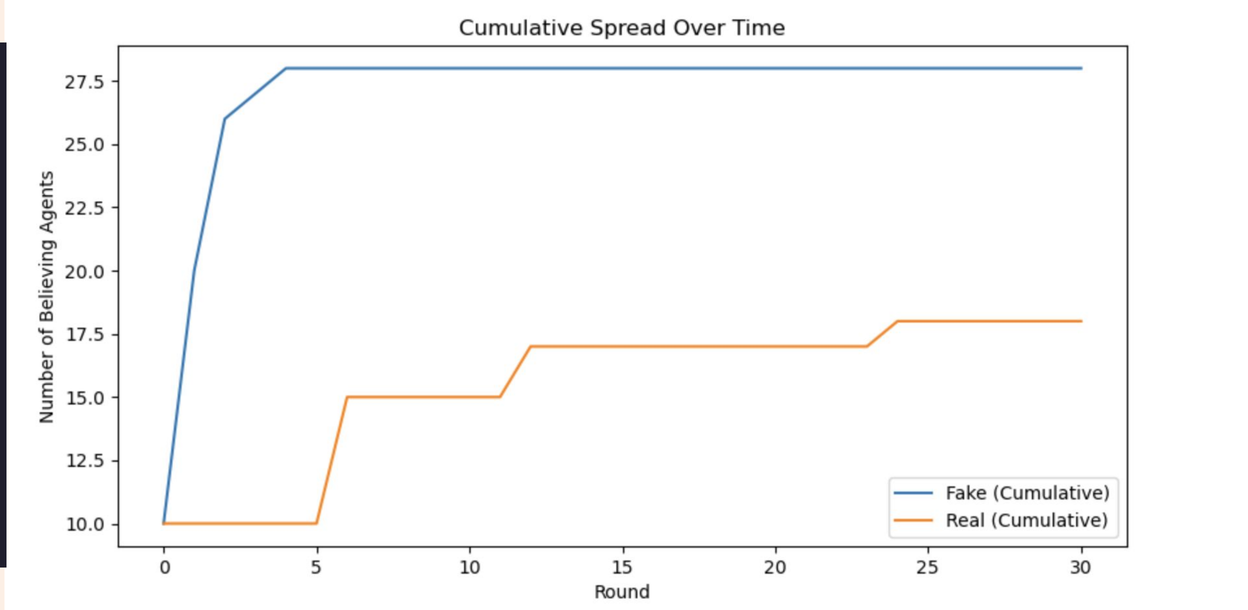
Real News - Avg Reach: 18.0 ± 5.6

Fake Peak Round: 0.0 (IQR 0.0-2.0)

Real Peak Round: 5.0 (IQR 5.0-5.0)

Fake Final Believers: 44.1 ± 31.6

Real Final Believers: 18.0 ± 5.6



Next Steps

Fix Hypothesis Testing Logic

1. Review code and parameters for logical errors
2. Consider other metrics not currently being measured to see if statistical significance is observed

Code Quality Review

1. Program Quality and Code Reviews document as a guide
2. Add docstrings describing functions and doctests
3. Additional type hinting
4. Organize functions into appropriate files
5. etc.