



UNIVERSITY OF MILAN - BICOCCA
Scuola di Scienze
Dipartimento di Informatica, Sistemi e Comunicazione
Corso di Scienze Informatiche

Simulating Rogers' Innovation Diffusion through GABMs

Report of:

Sanvito Marco 886493

Academic year 2024-2025

ABSTRACT

This project investigates the simulation of innovation diffusion through Generative Agent-Based Modeling (GABM), based on Everett Rogers' Diffusion of Innovations theory, focusing on the decision stage and with some simplifications. By employing advanced AI agent frameworks, particularly AutoGen, the system enables agents to simulate human-like reasoning, adaptive decision-making and socially responsive behavior. The simulation models the spread of innovation across social networks, incorporating rich behavioral profiles to reflect individual and group dynamics.

The focus of the project is not only trying to simulate the diffusion of innovation but also to explore how generative agents can be used to model complex social phenomena and decision-making processes.

The report begins with an overview of AI agent frameworks and their capabilities, followed by an explanation of the GABM approach and its relevance to modeling complex social phenomena. Rogers' theoretical model is then examined, highlighting key adopter categories and the mechanisms of diffusion. Finally, the implementation of the simulation is presented, including system architecture, agent design and network configuration.

It then concludes with a discussion of the results obtained and future development directions, focusing on the possibilities offered by the AutoGen framework to enhance agent autonomy, multi-agent collaboration and scalability in simulating increasingly complex innovation ecosystems.

The repository containing the complete simulation results and additional materials can be found at GitHub - GABMs-Rogers-Innovation-Diffusion.

Contents

1. Introduction to AI Agents	1
1.1. AI Agents as Practical Task Automation Tools	1
1.1.1. Contemporary AI Agent Applications for Task Automation	1
1.1.2. Limitations and Challenges	2
1.2. The AutoGen Framework: Advanced Multi-Agent Orchestration	2
1.2.1. AutoGen Architecture: Core vs. AgentChat	2
1.2.2. Agent Types and Capabilities	3
1.2.3. Tool and MCP server Integration	5
1.2.4. AutoGen Design Patterns	6
1.2.5. Advanced AutoGen Capabilities	7
1.2.6. AutoGen Extensions Ecosystem	7
1.3. Comments	7
2. Generative Agent-Based Modeling (GABM): From Task Automation to Simulation	8
2.1. Evolution from Traditional AI Agents to Generative Models	8
2.2. The Emergence of Generative Agent-Based Models	8
2.3. Technical Architecture of GABM	8
2.3.1. Memory and Reflection Systems	8
2.3.2. Behavioral Architecture	9
2.4. The Concordia Framework for Advanced GABM	9
2.5. Some examples of GABMs	9
2.5.1. Complex Networks and Social Dynamics in GABM	9
2.5.2. GABM Applications in Game Theory and Strategic Interaction	10
2.5.3. Epidemic Modeling and Public Health Applications	10
2.5.4. GABM Applications in Social Simulation	11
2.6. Methodological Advantages and Challenges	11
2.6.1. Enhanced Realism and Behavioral Authenticity	12
2.6.2. Technical and Computational Limitations	12
2.6.3. Methodological and Reproducibility Concerns	12
2.6.4. Interpretability and Control Limitations	12
2.7. Validation Challenges of GABMs	13
2.7.1. Established Validation Frameworks	13
2.8. Simplified GABM Approach for Innovation Diffusion	13
3. Rogers' Diffusion of Innovations Theory: A Comprehensive Framework	14
3.1. Theoretical Foundation: Rogers' Diffusion of Innovations	14
3.1.1. The Four Key Elements of Diffusion	14
3.1.2. The Innovation-Decision Process	16
3.1.3. Adopter Categories: Profiles and Characteristics	17
3.2. Rogers's Curve	19
3.3. Social Pressure and Network Influence in Innovation Diffusion	20
3.3.1. The Dynamics of Social Pressure	20
3.3.2. Network Topology and Diffusion Patterns	21
3.3.3. Agent-Based Modeling Insights	21
3.4. The Importance of Modeling the Decision Phase	22

3.4.1. Project Path	23
4. Generative Agent-Based Modeling of Innovation Diffusion: Implementation and Methodology	24
4.1. Simplification of Rogers' Theory	24
4.2. Core Architecture	24
4.2.1. Simulation Flow	24
4.2.2. Decision-Making Mechanism	24
4.2.3. Theoretical Integration	24
4.2.4. State Management Protocol	25
4.2.5. Core Decision-Making Process	25
4.3. Agent-Based Decision-Making Architecture	25
4.3.1. Prompt Engineering and LLM Integration	25
4.3.2. Agent Behavioral Profiles and Category Implementation	27
4.3.3. Prompt Characteristics	30
4.4. NetworkX - Network Implementation	31
4.5. Implementation Challenges and Solutions	31
4.5.1. LLM Response Reliability	31
4.5.2. Prompt Engineering Challenges	32
4.6. Technology Stack and LLM Choice	33
4.6.1. Hardware Constraints	33
4.7. Configuration Management and Simulation Control	33
4.7.1. Comprehensive Configuration	33
4.7.2. Predefined Configuration Scenarios	34
4.7.3. Streamlit Application Interface	34
4.8. Agent Response Capture and Data Collection	34
4.8.1. Comprehensive Decision Recording	34
4.9. Validation Framework	34
4.9.1. Theoretical Consistency Validation	35
4.9.2. Methodological Limitations	35
5. Simulation Results and Analysis	36
5.1. Premises	36
5.1.1. Simulation Scope and Design Choices	36
5.1.2. Prompt Engineering and Decision Consistency	36
5.1.3. Response inconsistencies	37
5.1.4. Computational Constraints	37
5.1.5. Network and Global Influence Levels	37
5.2. Devil's Advocate Prompt Evaluation	37
5.2.1. Experimental Design	38
5.2.2. Findings and Implications	38
5.2.3. Decision Impact	39
5.3. Categories Initial Responses	40
5.4. Network Structure and Simulation Environment	40
5.5. Simulation of Unsuccessful Innovations	41
5.6. Comparative Analysis	42

5.6.1. Methodology	42
5.6.2. Cross-Run Analysis and Results Summary	43
5.6.3. Successful Innovation Runs (S1-S3)	44
5.6.4. Balanced Innovation Runs (B1-B3)	45
5.6.5. Category-Specific Adoption Patterns	46
5.6.6. Agents Response Analysis	49
5.6.7. Comparison	51
5.6.8. Network Positioning Effects	52
5.7. Detailed Single-Run Analysis	52
5.7.1. Successful Innovation Case Study (Run S1)	53
5.7.2. Balanced Innovation Case Study (Run B1)	61
6. Conclusions and Future Developments	71
6.1. Evaluation of Results	71
6.1.1. GABM Methodology Validation and Advantages	71
6.1.2. Innovation Attribute Impact and Decision Patterns	72
6.1.3. Key Empirical Insights	72
6.1.4. Limitations and Boundary Conditions	72
6.2. Future Developments Directions	73
6.2.1. Hardware Considerations	73
6.2.2. Experimenting with Alternative LLMs	73
6.2.3. Exploring Alternative Network Structures	73
6.2.4. Comprehensive Implementation of Rogers' Five-Stage Model	73
6.2.5. Refinement of Devil's Advocate Prompting Strategy	74
6.2.6. Systematic Prompt Engineering Research	74
6.2.7. Integration of Media Influence Mechanisms	74
6.2.8. Advanced Cognitive Modeling Through Society of Mind Agent	75
6.2.9. Addressing LLM Hallucination	75
6.3. Conclusions	75
Bibliography	77

Chapter 1

Introduction to AI Agents

Artificial Intelligence (AI) agents are software systems that incorporate machine learning, natural language processing, and decision-making capabilities. These features allow AI agents to operate in complex and uncertain environments with reduced human intervention.

While AI agents offer new possibilities for automation and problem-solving, their effectiveness depends on careful design, appropriate training and clear understanding of their limitations. They are tools that can support or enhance human work, but they require oversight and thoughtful integration into existing processes.

1.1. AI Agents as Practical Task Automation Tools

Over the course of the years, AI agents have evolved from experimental research prototypes into powerful, highly capable systems capable of scoping, planning and executing complex projects using a wide array of integrated tools, with minimal human oversight. These agents represent a significant advancement beyond traditional automation, functioning as intelligent collaborators that can reason, adapt and act independently.

1.1.1. Contemporary AI Agent Applications for Task Automation

1.1.1.1. Business Processes Automation

- **Data Analysis and Reporting:** Analyzing data, drafting emails and pulling reports, all on its own
- **Customer Service:** Automated response systems with contextual understanding
- **Project Management:** Breaking down complex projects into manageable subtasks
- **Quality Assurance:** Automated testing and compliance monitoring

1.1.1.2. Development and Technical Tasks

- **Code Generation:** GitHub Copilot has revolutionized code generation by providing context-aware suggestions and completing code snippets based on natural language prompts S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer [1]
- **Software Development:** Devin AI, as an example, has emerged as a powerful AI pair programmer, assisting developers in writing code more efficiently
- **Documentation Generation:** Automated technical writing and maintenance
- **Testing and Debugging:** Automated unit testing and code analysis

1.1.1.3. Productivity and Workflow Management

- **Calendar Optimization:** AI-powered scheduling that adapts to priorities and constraints

- **Email Management:** Automated sorting, drafting and response generation
- **Research and Information Gathering:** Web scraping and data synthesis
- **Content Creation:** Multi-modal content generation for various purposes

1.1.2. Limitations and Challenges

Despite their impressive capabilities, AI agents are not magic wands. Their effectiveness depends heavily on several key factors:

- **Quality of Implementation:** Poorly designed or inadequately trained agents can produce inaccurate, inefficient or even harmful outcomes
- **Clarity of Objectives:** AI agents perform best when given well-defined goals and structured contexts. Ambiguity or vague instructions can hinder performance
- **Integration with Tools and Environments:** An agent's ability to operate depends on access to compatible tools, APIs, databases and infrastructure
- **Human Oversight and Control:** Even the most advanced systems benefit from human supervision, especially in critical domains like healthcare, finance or security

AI agents mark a transformative shift in how automation and human-machine collaboration are approached. However, their impact is shaped by how they are designed, deployed and integrated into workflows. They are powerful tools, but not infallible and like any technology, they require expertise, vision and responsibility to unlock their full potential.

1.2. The AutoGen Framework: Advanced Multi-Agent Orchestration

AutoGen represents a significant advancement in AI agent orchestration, providing a comprehensive framework for building sophisticated multi-agent systems Q. Wu *et al.* [2]. Developed by Microsoft Research, AutoGen enables the creation of conversational AI agents that can collaborate, reason and execute complex tasks through structured interactions, with the framework utilizing a layered and extensible design with clearly divided responsibilities.

1.2.1. AutoGen Architecture: Core vs. AgentChat

The current AutoGen ecosystem is structured into two primary layers, each serving different levels of abstraction and use cases:

1.2.1.1. AutoGen Core (autogen-core)

AutoGen Core provides the foundational layer of the framework, implementing low-level primitives for multi-agent systems. The Core API implements message passing, event-driven agents and local and distributed runtime for flexibility and power, with support for cross-language compatibility including .NET and Python. Key features include:

- **Asynchronous Message Passing:** Native support for asynchronous communication patterns
- **Event-Driven Architecture:** Agents respond to events rather than following synchronous conversation flows
- **Distributed Runtime:** Support for scaling across multiple processes and machines
- **Cross-Language Interoperability:** Seamless integration between Python and .NET implementations

- **Low-Level Control:** Fine-grained control over agent behavior and message routing

1.2.1.2. AutoGen AgentChat (autogen-agentchat)

AutoGen AgentChat builds upon the Core layer to provide high-level abstractions specifically designed for conversational multi-agent scenarios. This layer is particularly suitable for the project requirements as it offers:

- **Simplified Agent Creation:** High-level APIs for rapid agent development
- **Conversation Management:** Built-in conversation flow control and context management
- **Tool Integration:** Streamlined integration with external tools and APIs
- **Human-in-the-Loop:** Native support for human participation in agent conversations
- **Pre-built Agent Types:** Common agent patterns implemented as reusable components

For this project, I utilize AutoGen AgentChat (version 0.7.2) due to its higher-level abstractions and simplified development experience, which align well with the objective of creating accessible multi-agent workflows without requiring deep expertise in distributed systems architecture, while I don't utilize all functionalities, they could be incorporated in future developments for the project.

Other alternatives considered were:

- Langchain - a framework similar to AutoGen but focused on document-based workflows
- Concordia - a framework for building and simulating complex multi-agent systems

1.2.2. Agent Types and Capabilities

AutoGen provides a rich ecosystem of agent types tailored for multi-agent systems. While the foundational agents in Autogen Core require manual setup, the AgentChat module introduces specialized agents with distinct roles. These are further extended in the `autogen-ext` layer, offering even more flexibility and functionality.

1.2.2.1. Assistant Agent

This Agent is designed for task completion and problem-solving. It is typically powered by a large language model and excels in generating technical solutions, writing code and providing expert-level knowledge across various domains. This agent type is ideal for collaborative problem-solving, as it can interact with other agents to tackle complex challenges. It can also refine responses within a conversation via feedback [3].

This type of Agent will be used for the current project due to its ability to define a system prompt, manage conversation flow and avoid the complexity associated with other agent types.

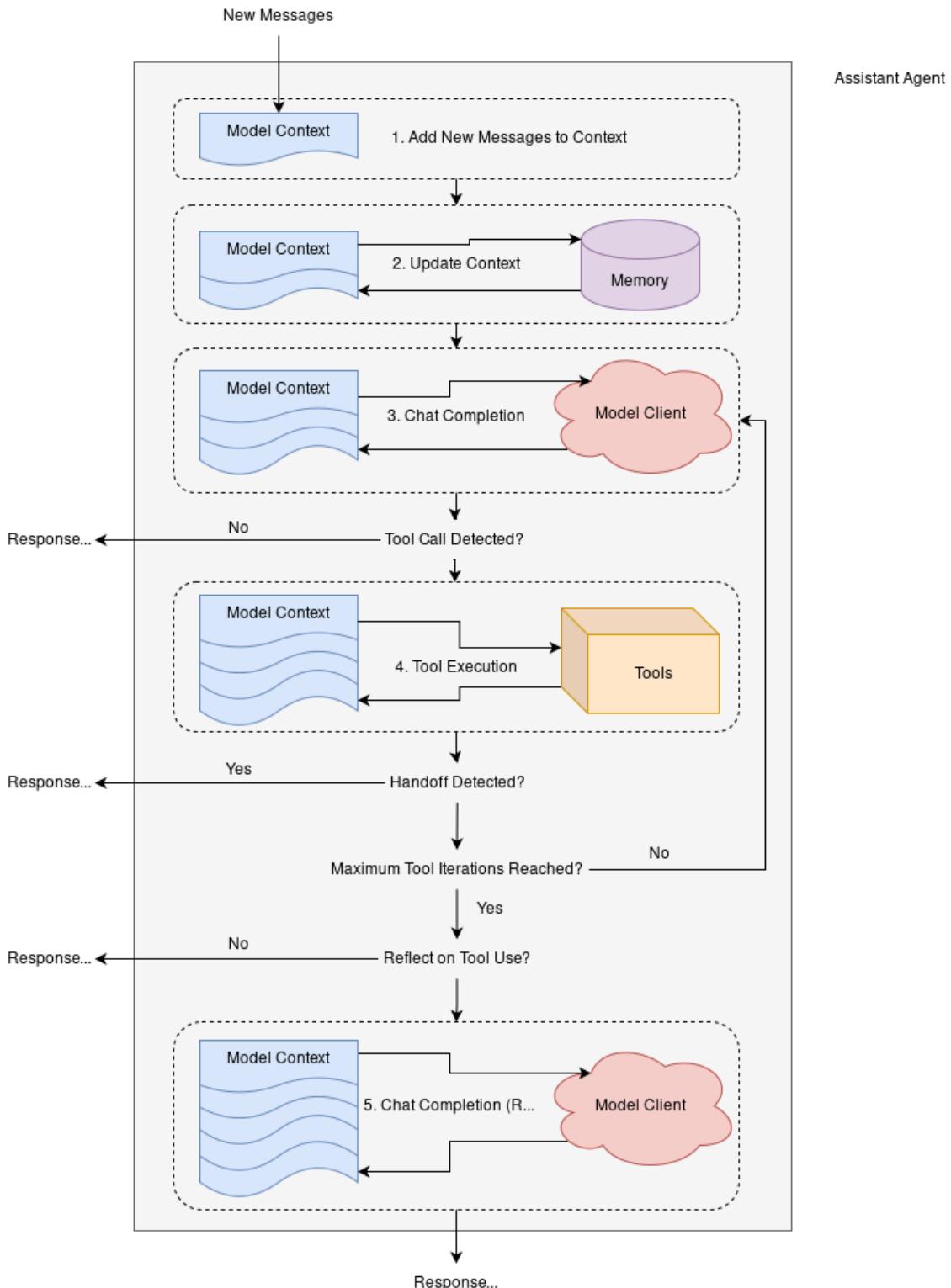


Figure 1: Assistant Agent Architecture [4]

1.2.2.2. User Proxy Agent

The User Proxy Agent, as the name suggests, serves as a bridge between human users and the agent system. It enables human oversight and intervention, allowing users to manually approve or reject agent actions before they are executed. This agent also facilitates the collection of human feedback, which can be used to improve agent behavior over time. In scenarios involving complex decisions, the User Proxy Agent can escalate tasks to human operators for resolution.

1.2.2.3. Society Of Mind Agent

An Agent that operates as a collective of sub-agents working together to generate responses. Each sub-agent may represent a different perspective, skill or strategy and their interactions simulate a form of internal dialogue. This architecture allows for more nuanced reasoning and decision-making, making it particularly useful in tasks that require multi-faceted analysis or creativity.

1.2.2.4. Message Filter Agent

The Message Filter Agent acts as a wrapper that intercepts and processes incoming messages before they reach the core agent. This filtering mechanism can be used to remove irrelevant content, apply specific rules or pre-process messages in multi-agent environments. It ensures that only relevant and well-structured inputs are passed to the main agent, allowing for improved overall system efficiency and coherence.

1.2.3. Tool and MCP server Integration

AutoGen's tool integration allows agents to call Python functions as tools and interact with MCP (Model Context Protocol) servers for external capabilities.

Agents can use Python functions as tools, making them available during conversations:

```

1  # Define a tool using a Python function.                                     Python
2  async def web_search_func(query: str) -> str:
3      """Find information on the web"""
4      return "AutoGen is a programming framework for building multi-agent
5      applications."
6
7  # This step is automatically performed inside the AssistantAgent if the tool is a
8  # Python function.
9  web_search_function_tool = FunctionTool(web_search_func, description="Find
10 information on the web")
11 # The schema is provided to the model during AssistantAgent's on_messages call.
12 web_search_function_tool.schema
13
14 agent = AssistantAgent(name="Agent With Tool", model_client=client,
15 tools=[web_search_function_tool], reflect_on_tool_use=True)

```

Besides Python functions as tools, it supports MCP servers, which are external services that can be called by agents to perform specific tasks. These servers can be used to integrate with existing systems or provide additional functionality.

An important aspect of tool integration is the ability to reflect on tool use, which allows agents to evaluate the effectiveness of their tool calls and adapt their strategies accordingly. This is particularly useful in complex workflows where multiple tools may be required to achieve a goal.

What happens is that the agent uses the tool's description to understand its capabilities and limitations, enabling it to make informed decisions about when and how to use each tool effectively.

1.2.4. AutoGen Design Patterns

The AutoGen framework suggests multiple agents design patterns. These patterns provide structured approaches to common multi-agent system challenges:

1.2.4.1. Sequential Workflow Pattern

Agents communicate in a predetermined sequence, with each agent building upon the previous agent's output:

- **Use Cases:** Code review workflows, document editing pipelines, multi-step analysis tasks
- **Implementation:** Agent A processes input → Agent B refines output → Agent C validates results
- **Benefits:** Clear accountability, predictable flow, easy debugging
- **Example:** A data analyst agent extracts insights → A visualization agent creates charts → A report agent generates final presentation

1.2.4.2. Group Chat Pattern

Multiple agents participate in a dynamic conversation, with a manager agent coordinating the discussion:

- **Use Cases:** Brainstorming sessions, complex problem-solving, multi-perspective analysis
- **Implementation:** Manager agent orchestrates conversation flow and determines speaking order
- **Benefits:** Rich interaction, diverse perspectives, emergent solutions
- **Example:** Product development team with designer, engineer, marketer and project manager agents

1.2.4.3. Handoffs

Agents transfer control of a task or conversation to another agent using a tool call, allowing for specialization and focused expertise:

- **Use Cases:** Task delegation, expertise sharing, collaborative problem-solving
- **Implementation:** Agent A completes initial analysis → Hands off to Agent B for deeper investigation
- **Benefits:** Efficient use of specialized skills, reduced cognitive load, improved outcomes
- **Example:** Research agent gathers background information → Subject matter expert agent provides in-depth analysis

1.2.4.4. Multi-Agent Debate

Agents engage in structured debates to explore different perspectives and reach consensus:

- **Use Cases:** Policy discussions, ethical dilemmas, strategic planning
- **Implementation:** Moderator agent facilitates debate, ensuring all voices are heard
- **Benefits:** Diverse viewpoints, critical thinking, well-rounded conclusions
- **Example:** Ethics committee with legal, technical and social impact agents

1.2.4.5. Reflection Pattern

Agents engage in self-evaluation and iterative improvement of their outputs:

- **Use Cases:** Quality assurance, creative writing, complex reasoning tasks
- **Implementation:** Primary agent generates initial output → Critic agent evaluates → Primary agent refines
- **Benefits:** Higher quality outputs, self-correction capabilities, continuous improvement
- **Example:** Writer agent creates content → Editor agent provides feedback → Writer agent revises

1.2.5. Advanced AutoGen Capabilities

1.2.5.1. Memory and Context Management

AutoGen includes experimental Canvas Memory functionality, providing shared “whiteboard” memory for agents to collaborate on common artifacts such as code, documents or illustrations. This enables:

- **Shared Workspace:** Common memory space accessible to all agents in a group
- **Artifact Collaboration:** Joint editing and refinement of documents and code
- **Context Persistence:** Maintaining conversation state across extended interactions
- **Version Control:** Tracking changes and maintaining history of shared artifacts

1.2.5.2. Extensibility and Customization

The modular architecture supports extensive customization:

- **Custom Agent Types:** Creating specialized agents for domain-specific tasks
- **Plugin System:** Third-party extensions for additional functionality
- **Model Integration:** Support for various LLM providers and local models
- **Runtime Environments:** Flexible deployment options from local to cloud-scale

1.2.6. AutoGen Extensions Ecosystem

The AutoGen ecosystem includes extensions that add specialized agents (e.g., MultimodalWebSurfer for web browsing, FileSurfer for documents, VideoSurfer for videos). Another extension, models, enables seamless integration of various LLM providers and local models.

1.3. Comments

While AI agents have proven effective in automating tasks, their potential extends far beyond operational efficiency. The emergence of Generative Agent-Based Modeling (GABM) marks a paradigm shift. Where agents not only act, but simulate, reason and reflect within complex social systems. The following chapter explores this evolution.

Chapter 2

Generative Agent-Based Modeling (GABM): From Task Automation to Simulation

Building on orchestration frameworks like AutoGen for multi-agent collaboration, GABM integrates LLM-driven cognition, memory and social interaction to study emergent behavior.

2.1. Evolution from Traditional AI Agents to Generative Models

After the introduction of AI agents powered by large language models (LLMs), the landscape of artificial intelligence began to shift dramatically. These advanced models enabled agents to not only understand and generate human-like text but also to engage in more complex forms of reasoning and interaction.

2.2. The Emergence of Generative Agent-Based Models

Generative Agent-Based Modeling (GABM) represents a novel approach that integrates large language models (LLMs) with traditional agent-based modeling methodologies. This fusion creates agents capable of exhibiting human-like reasoning, creativity and social behavior, offering potential possibilities for simulating complex social systems and human interactions as demonstrated in the paper J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein [5], where a social simulation environment was created and agents powered by LLMs simulated believable human behavior.

2.3. Technical Architecture of GABM

Recent research has introduced a variety of generative agent-based modeling (GABM) approaches, each with its own architectural choices and implementation details J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein [5], A. S. Vezhnevets *et al.* [6], Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno [7]. While there is no single standard architecture for GABMs, most share common elements such as the integration of large language models (LLMs).

2.3.1. Memory and Reflection Systems

- **Episodic Memory:** Storing and retrieving specific interaction experiences with temporal and contextual tags
- **Semantic Memory:** Maintaining general knowledge and learned patterns about the world and other agents
- **Reflection Mechanisms:** Periodic self-evaluation and behavior adjustment based on past experiences

- **Importance Weighting:** Prioritizing memories based on relevance, emotional significance and recency
- **Memory Consolidation:** Converting short-term experiences into long-term behavioral patterns

2.3.2. Behavioral Architecture

- **Goal-Directed Behavior:** Pursuing both short-term objectives and long-term aspirations
- **Social Cognition:** Understanding social hierarchies, relationships and cultural contexts
- **Emotional Processing:** Generating and responding to emotional states and social dynamics
- **Personality Consistency:** Maintaining stable personality traits while allowing for growth and change
- **Adaptive Learning:** Modifying behavior based on social feedback and environmental changes

2.4. The Concordia Framework for Advanced GABM

Google DeepMind's Concordia framework represents one of the recent approaches to generative agent-based modeling, providing a comprehensive library for creating, configuring and studying generative agents in grounded physical, social or digital environments A. S. Vezhnevets *et al.* [6]. Concordia enables researchers and developers to build agents that can participate in complex social simulations with high realism and sophistication.

While this framework could have been used for the current project, because it is still under active development and not yet fully mature, I opted for a more established solution.

2.5. Some examples of GABMs

Since this project will focus on decision-making, it is essential to explore existing GABMs that address these themes and use them as a foundation for this work.

While the paper J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein [5] is one of the biggest example of GABMs focusing on social interactions and decision-making processes, it is not the only one and due to its complexity, I decided to search for additional relevant studies.

2.5.1. Complex Networks and Social Dynamics in GABM

Recent research has demonstrated how GABMs can model complex network formation and social dynamics. An example is the paper that explored the self-organization of generative agents in forming complex network structures, where nodes represented generative agents whose behavior was controlled by GPT-3.5-turbo. The agents were initialized using specific prompts, simulating the growth of an online social network G. D. Marzo, L. Pietronero, and D. Garcia [8].

The LLM created a network with a hub-and-spoke structure that does not resemble the classical results obtained from preferential attachment algorithms. Interestingly, the researchers found that this was a consequence of a bias in the selection of nodes by the LLM that depended on

their name. This study highlights both the potential and limitations of these models, demonstrating the importance of careful experimental design and bias mitigation.

Similarly, in another paper, 10 artificial agents based on the LLM Claude-2.1 were deployed and allowed to freely interact without specific priors. The agents showed a tendency to interact repeatedly with the same peers rather than exploring new connections and an analysis of their conversations indicated homophily, a common characteristic of human social networks S. Lai, Y. Potter, J. Kim, R. Zhuang, D. Song, and J. Evans [9].

2.5.2. GABM Applications in Game Theory and Strategic Interaction

One of the most compelling applications of GABM has been in the study of game theory and strategic interaction, where LLM-driven agents can mimic intricate internal features of human cognition. Researchers have successfully deployed GABMs in various classic game theory scenarios, revealing both similarities and differences with human behavior patterns Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno [7].

2.5.2.1. Game Theory Experiments with LLMs

LLM-driven agents have been extensively tested in various economic games, including the Dictator Game, Ultimatum Game, Prisoner's Dilemma and Public Goods Games. These experiments illustrate the feasibility of simulating individuals with a wide range of characteristics and traits, unlike traditional rule-based agents.

In the Ultimatum Game, LLMs demonstrated behavior that closely aligned with human decision trends, but also revealed biases related to gender and social roles. For instance, agents with male identifiers were more likely to accept unfair offers from agents with female identifiers, while female agents were less inclined to accept unfair offers from male agents G. Aher, R. I. Arriaga, and A. T. Kalai [10].

The Prisoner's Dilemma experiments showed that LLMs exhibited cooperation rates of 65.4% on average, significantly higher than the 37% found in meta-analyses of human participants. This suggests that LLMs may have different risk and cooperation preferences compared to humans, potentially reflecting their training on cooperative and helpful interactions Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno [7].

2.5.2.2. Strategic Behavior and Adaptation

Research has shown that this type of modeling can be conditioned to follow certain strategic behaviors, modifying their cooperative profiles based on specific prompts. However, this conditioning revealed the complexity of prompting these models, as some initial hypotheses had to be discarded. For instance, prompting for "selfish behavior" sometimes led to more cooperation than competitive scenarios, indicating the nuanced relationship between prompt design and behavioral outcomes Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno [7].

2.5.3. Epidemic Modeling and Public Health Applications

One of the most promising applications of GABM lies in epidemic modeling, where traditional approaches struggle to capture the complexity of human behavior during outbreaks. Current epidemic models, even those using traditional agent-based modeling, must make certain

assumptions about how humans react during an outbreak. GABMs can transfer the decision-making process directly to LLMs without having to introduce these assumptions.

Williams et al. explored these possibilities using a simple GABM epidemic model, simulating virus propagation in a population where ChatGPT decided whether individual agents would exit home at each timestep. The agents received varying levels of information: baseline scenarios with no virus information, self-health feedback scenarios including symptom information and full feedback scenarios including information about the virus and infection rates in the community R. Williams, N. Hosseinichimeh, A. Majumdar, and N. Ghaffarzadegan [11].

The results demonstrated the potential of GABMs for epidemic modeling:

- **Baseline Model:** Reproduced SIR-like model results with all agents exiting homes daily
- **Self-Health Feedback:** Agents with symptoms usually decided to stay home
- **Full Feedback:** Even asymptomatic agents decided to stay home when informed about community infection rates, greatly diminishing outbreak size

2.5.4. GABM Applications in Social Simulation

Generative agent-based models are opening new possibilities in social simulation, helping researchers explore how people behave in complex, dynamic environments. Unlike traditional rule-based systems, these models can reflect the unpredictability and nuance of real human interactions.

2.5.4.1. Information Propagation in Social Networks

Gao et al. created a simulated social media platform to study how ideas and rumors spread. Each agent powered using ChatGLM made independent choices: whether to share a post, write something new or stay quiet, based on patterns observed in real-life behavior. By tuning the model with data on sharing habits, follower networks and attention spans, they were able to recreate familiar patterns of information spread. They also pinpointed key moments when a message shifts from grassroots sharing to viral influence driven by popular accounts C. Gao *et al.* [12].

2.5.4.2. Simulating Collective Decision-Making with Multi-Agent Debates

Liang et al. explored group cognition by orchestrating debates among heterogeneous LLM agents: GPT-4, Vicuna and GPT-3.5. They found that isolated self-reflection often led individual models into degenerative loops of circular reasoning. In contrast, structured argumentation among multiple agents produced richer and more accurate conclusions, reducing logical errors and enhancing consistency. This multi-LLM debate paradigm illustrates how GABMs can emulate real-world deliberative forums, providing a sandbox for testing policy proposals, design concepts or strategic plans under diverse social dynamics T. Liang *et al.* [13].

2.6. Methodological Advantages and Challenges

While the precedent sections explored some examples of GABMs, it is important to acknowledge the methodological advantages and challenges that come with this approach. GABMs offer a unique blend of capabilities that traditional agent-based models (ABMs) struggle to achieve, but they also introduce new complexities that must be addressed.

2.6.1. Enhanced Realism and Behavioral Authenticity

- **Cognitive Biases and Heuristics:** Realistic decision-making flaws including confirmation bias, anchoring effects and availability heuristics
- **Cultural Competency and Context:** Understanding and expressing cultural norms, values and communication styles
- **Individual Personality Variation:** Diverse personalities, backgrounds and capabilities that reflect real human variation
- **Moral and Ethical Reasoning:** Complex ethical decision-making processes that reflect human moral reasoning

2.6.2. Technical and Computational Limitations

- **Computational Resource Requirements:** Large-scale GABM simulations require substantial computational resources, including powerful GPUs and significant memory capacity
- **Scalability Constraints:** Hardware requirements for running multiple agents simultaneously can be prohibitive, limiting the number of agents that can be effectively simulated
- **Model Validation Complexity:** Difficulty in validating agent behavior against real-world data, particularly for novel social scenarios where ground truth is unavailable
- **Behavioral Consistency Issues:** Maintaining consistent agent behavior across different contexts and extended time periods remains challenging

2.6.3. Methodological and Reproducibility Concerns

A main concern is the reproducibility of GABM simulations, which can be affected by various factors Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno [7]:

- **Prompt Sensitivity and Engineering:** LLMs are highly sensitive to word ordering, formatting and non-semantic features of prompts, making results difficult to replicate
- **Training and Fine-Tuning Biases:** Models may propagate biases from training data and posterior fine-tuning processes, affecting simulation validity
- **Hallucination and Factual Accuracy:** Models may generate plausible but factually incorrect responses that compromise simulation integrity
- **Cross-Model Behavioral Variation:** Different LLMs exhibit distinct behavioral patterns and preferences that can significantly affect study outcomes
- **Version Dependency:** Different versions of the same LLM may produce varying outputs, complicating longitudinal studies and comparisons across different simulation runs

2.6.4. Interpretability and Control Limitations

- **Black Box Decision-Making:** The reasoning process behind agent decisions is often opaque, making it difficult to understand why specific behaviors emerge
- **Limited Behavioral Control:** There is minimal control over specific agent responses, unlike traditional ABMs where behaviors are explicitly programmed
- **Emergent Behavior Unpredictability:** Complex interactions between agents can produce unexpected emergent behaviors that are difficult to anticipate or control

2.7. Validation Challenges of GABMs

The validation of GABMs presents significant methodological challenges that distinguish them from traditional agent-based models. Unlike conventional ABMs with transparent rule-based behaviors, GABMs operate as black boxes, making it difficult to trace the reasoning behind agent decisions and assess model reliability.

The opacity of LLM-driven decision-making processes creates a fundamental validation problem: how can researchers verify that agent behaviors accurately represent the phenomena being studied when the underlying reasoning mechanisms are not directly observable? This challenge is compounded by the stochastic nature of LLM outputs and their sensitivity to prompt variations.

2.7.1. Established Validation Frameworks

Research investigating whether GABMs address the traditional limitations of agent-based modeling, the lack of realism in social systems, has identified five primary validation approaches currently employed M. Larooij and P. Törnberg [14]:

- **Human-Centered Validation:** Leveraging human evaluators or LLMs themselves to assess the realism and plausibility of agent behaviors
- **Theory-Based Validation:** Comparing emergent agent behaviors against established social theories, psychological frameworks and empirically-documented patterns
- **Comparative Model Validation:** Benchmarking GABM outputs against results from traditional agent-based models and other approaches to identify convergent or divergent patterns
- **Empirical Data Validation:** Utilizing real-world datasets—from surveys, experiments or observational studies, to validate agent behaviors against documented human decision-making patterns
- **Internal Consistency Validation:** Ensuring logical coherence and consistency in agent responses by using for example sensitivity analysis

2.8. Simplified GABM Approach for Innovation Diffusion

Using as a foundation the examples discussed in this chapter, I will develop a simplified GABM that retains the essential elements of innovation diffusion while minimizing complexity. This focused approach enables investigation into how LLM-driven agents interpret and enact adoption decisions, providing insights into both the capabilities and limitations of generative models in simulating social diffusion dynamics.

Chapter 3

Rogers' Diffusion of Innovations Theory: A Comprehensive Framework

The diffusion of innovations theory, developed by Everett Rogers in 1962, provides a comprehensive framework for understanding how new ideas, technologies and practices spread through social systems E. Rogers [15]. This work has been instrumental in explaining the adoption patterns of innovations across various domains, from agricultural practices to digital technologies. The theory offers a robust theoretical foundation that continues to influence research in sociology, marketing, technology adoption and computational social science.

3.1. Theoretical Foundation: Rogers' Diffusion of Innovations

Rogers' diffusion of innovations theory defines diffusion as “the process in which an innovation is communicated through certain channels over time among the members of a social system” E. Rogers [15]. This foundational definition highlights the four critical elements that determine the success and trajectory of innovation diffusion: the innovation itself, communication channels, time and the social system.

3.1.1. The Four Key Elements of Diffusion

3.1.1.1. Innovation Characteristics

The perceived attributes of an innovation significantly influence its rate of adoption. Rogers identified five key attributes that determine adoption rates:

Relative Advantage: The degree to which an innovation is perceived as superior to existing alternatives. This advantage may be economic, social or related to convenience and satisfaction E. Rogers [15]. Innovations with clear relative advantages typically achieve faster adoption rates, though the perception of advantage varies among different adopter categories.

Compatibility: The extent to which an innovation aligns with existing values, past experiences and needs of potential adopters E. Rogers [15]. Innovations that are incompatible with prevailing social norms face significant resistance and slower adoption rates. Compatibility encompasses both practical considerations and cultural alignment.

Complexity: The perceived difficulty of understanding and using an innovation E. Rogers [15]. More complex innovations generally exhibit slower adoption rates, particularly among later adopter categories who may lack technical expertise or willingness to invest learning time. The complexity factor often creates barriers that prevent widespread adoption.

Trialability: The degree to which an innovation can be experimented with on a limited basis E. Rogers [15]. Innovations that can be tested without significant commitment typically achieve faster adoption rates as they reduce uncertainty for potential adopters. Trialability allows individuals to learn by doing rather than relying solely on others' experiences.

Observability: The extent to which the results of an innovation are visible to others E. Rogers [15]. Highly observable innovations stimulate peer discussion and facilitate social proof mechanisms that accelerate diffusion. Observable benefits create demonstration effects that influence potential adopters.

3.1.1.2. Communication Channels

The means by which information about an innovation flows between individuals fundamentally shapes diffusion patterns E. Rogers [15]. Rogers distinguished between different types of communication channels:

Mass Media Channels: Effective for creating awareness and knowledge about innovations, particularly in the early stages of diffusion. These channels can reach large audiences rapidly but are less effective for persuasion. Mass media plays a crucial role in the knowledge stage of the adoption process.

Interpersonal Channels: More influential in persuading individuals to adopt innovations, especially when sources are similar to receivers. Face-to-face communication allows for two-way interaction and clarification of doubts, making it more persuasive than mass media.

Homophilous Communication: Occurs between similar individuals and is more comfortable but less effective for diffusion. While homophilous communication faces fewer barriers, it may not introduce truly novel ideas or perspectives.

Heterophilous Communication: Involves dissimilar individuals and is more effective for diffusion but may face barriers. These communications often introduce innovations across social boundaries but require greater effort to establish effective exchange.

3.1.1.3. Time Dimension

Time manifests in diffusion research through three primary mechanisms:

Innovation-Decision Process Time: The length of time required for an individual to pass through the five stages of the adoption process. This temporal aspect varies significantly among adopter categories, with innovators typically requiring less time than laggards.

Innovativeness: The earliness of adoption relative to other members of the social system, which forms the basis for Rogers' adopter categories. This relative measure creates the foundation for understanding adoption timing patterns.

Rate of Adoption: The speed with which an innovation is adopted by members of a social system, typically following an S-curve pattern.

3.1.1.4. Social System

The social system provides the context within which diffusion occurs and establishes the boundaries for innovation spread. Key characteristics include:

Social Structure: The patterned arrangements of units in a system that affect diffusion through established communication patterns. Social structure influences who communicates with whom and determines the flow of information and influence.

System Norms: The established behavioral patterns for members of a social system that can either facilitate or hinder diffusion. Norms create expectations about appropriate behavior and can create resistance to incompatible innovations.

Opinion Leadership: The degree to which an individual influences others' attitudes or behaviors in a desired way. Opinion leaders serve as intermediaries between change agents and the general population, often determining the success or failure of diffusion efforts.

Change Agents: Individuals who influence innovation decisions by promoting adoption. Change agents often come from outside the social system and work to introduce innovations, though their effectiveness depends on their credibility and relationship with potential adopters.

3.1.2. The Innovation-Decision Process

Rogers conceptualized the innovation-decision process as consisting of five sequential stages:

3.1.2.1. Knowledge Stage

The individual becomes aware of the innovation and gains understanding of its functions. This stage involves exposure to the innovation's existence and developing comprehension of how it works. Three types of knowledge are relevant:

- **Awareness-knowledge:** Information about the innovation's existence
- **How-to-knowledge:** Information about proper use of the innovation
- **Principles-knowledge:** Information about underlying principles explaining how and why the innovation works

3.1.2.2. Persuasion Stage

The individual forms a favorable or unfavorable attitude toward the innovation. This stage is more affectively oriented, with individuals actively seeking information about the innovation and mentally applying it to their present and anticipated future situations. The five perceived attributes of innovations are particularly important during this stage.

3.1.2.3. Decision Stage

The individual engages in activities that lead to adoption or rejection of the innovation. This stage represents the choice between adoption and rejection.

An important aspect of this stage is that the decision to adopt or reject is not final, individuals may reverse their decisions based on new information or experiences. Additionally, the decision-making process can be influenced by social pressures, personal values and perceived risks associated with the innovation.

3.1.2.4. Implementation Stage

The individual puts the innovation to use. This stage involves the actual utilization of the innovation and may require acquiring new skills or modifying existing practices. Implementation may lead to reinvention as users adapt the innovation to their specific circumstances.

3.1.2.5. Confirmation Stage

The individual seeks reinforcement of the innovation decision. This stage involves ongoing evaluation of the decision's consequences and may result in continued adoption, discontinuance or modification of the innovation. Confirmation helps resolve any cognitive dissonance that may arise from the adoption decision.

3.1.3. Adopter Categories: Profiles and Characteristics

Rogers identified five categories of adopters based on their timing of adoption relative to the overall population. Each category exhibits distinct characteristics that influence their propensity to adopt innovations:

3.1.3.1. Innovators (2.5% of population)

Innovators are characterized by their venturesome nature and willingness to take risks. They possess substantial financial resources, cosmopolite social relationships and demonstrate scientific orientation. Innovators exhibit:

- **Venturesome personality:** High tolerance for uncertainty and willingness to take calculated risks
- **Financial resources:** Substantial liquidity to absorb potential losses from unsuccessful innovations
- **Technical competence:** Ability to understand and apply complex technical knowledge
- **Cosmopolite orientation:** Social networks extending beyond local boundaries to national and international levels
- **Values and norms:** Orientation toward science, technology and rational decision-making
- **Risk tolerance:** Comfort with high degrees of uncertainty and potential setbacks
- **Control of resources:** Access to financial, technical and informational resources necessary for innovation adoption

The essential role of innovators is to launch innovations into the social system by importing them from external sources and serving as gatekeepers between the system and its environment.

3.1.3.2. Early Adopters (13.5% of population)

Early adopters serve as opinion leaders and are characterized by their integration into the local social system while maintaining progressive attitudes. They demonstrate:

- **Opinion leadership:** Typically the highest degree of influence within their social system
- **Local integration:** Well-integrated into the social system while maintaining progressive outlook
- **Respected status:** Earned reputation and trust among potential adopters
- **Judicious decision-making:** Careful evaluation that maintains their credibility as successful adopters

- **Change agent contact:** Frequent interaction with change agents and external information sources
- **Financial resources:** Above-average economic status enabling adoption of innovations
- **Educational attainment:** Higher levels of formal education compared to later adopters

Early adopters are crucial for triggering critical mass as they serve as role models and reduce uncertainty for subsequent adopters.

3.1.3.3. Early Majority (34% of population)

The early majority represents adopters who follow after careful deliberation but before the system's average adoption time. Their characteristics include:

- **Deliberate decision-making:** Thorough evaluation process requiring extended interaction time
- **Social integration:** Above-average social participation and network connectivity
- **Moderate opinion leadership:** Some influence within their social circles but not system-wide leaders
- **Economic security:** Above-average socioeconomic status providing resources for adoption
- **Information seeking:** Active pursuit of information about innovations before adoption
- **Risk aversion:** Preference for established innovations with demonstrated success
- **Peer consultation:** Reliance on trusted peers and opinion leaders for adoption decisions

The early majority's adoption decisions are critical as they represent the largest single group before the average adoption time.

3.1.3.4. Late Majority (34% of population)

The late majority approaches innovations with skepticism and adopts primarily due to economic necessity or increasing network pressure. They exhibit:

- **Skeptical attitude:** Cautious and doubtful approach to new innovations
- **Below-average resources:** Limited financial means making innovation failure costly
- **Network pressure sensitivity:** Adoption often driven by increasing peer pressure
- **Economic motivation:** Focus on economic necessity rather than status or novelty
- **Traditional orientation:** Preference for established practices and conventional approaches
- **Limited opinion leadership:** Minimal influence on others' adoption decisions
- **Support requirements:** Need for extensive assistance and encouragement during adoption

The late majority's adoption often occurs when network effects create pressure to conform and non-adoption becomes economically or socially disadvantageous.

3.1.3.5. Laggards (16% of population)

Laggards are the most traditional adopters who resist innovations until adoption becomes unavoidable. Their characteristics include:

- **Traditional values:** Strong orientation toward past practices and conventional approaches

- **Lowest socioeconomic status:** Most limited financial resources and tend to have the lowest social status
- **Isolated networks:** Social relationships primarily with individuals sharing traditional values
- **Suspicious attitude:** Skeptical view of innovations, change agents and scientific approaches
- **Localized orientation:** Focus on local rather than cosmopolite information sources
- **Economic constraints:** Severe resource limitations making innovation failure particularly problematic
- **Advanced age:** Typically older individuals with established routines and practices

Laggards represent the final group to adopt innovations, often doing so only when traditional alternatives are no longer viable or available.

3.2. Rogers's Curve

The diffusion of innovations typically follows an S-shaped curve, illustrating the cumulative adoption over time. While this curve could be used as a validation mechanism, this project focuses on the individual decision-making process rather than the overall diffusion pattern, meaning it's a possibility to obtain completely different results.

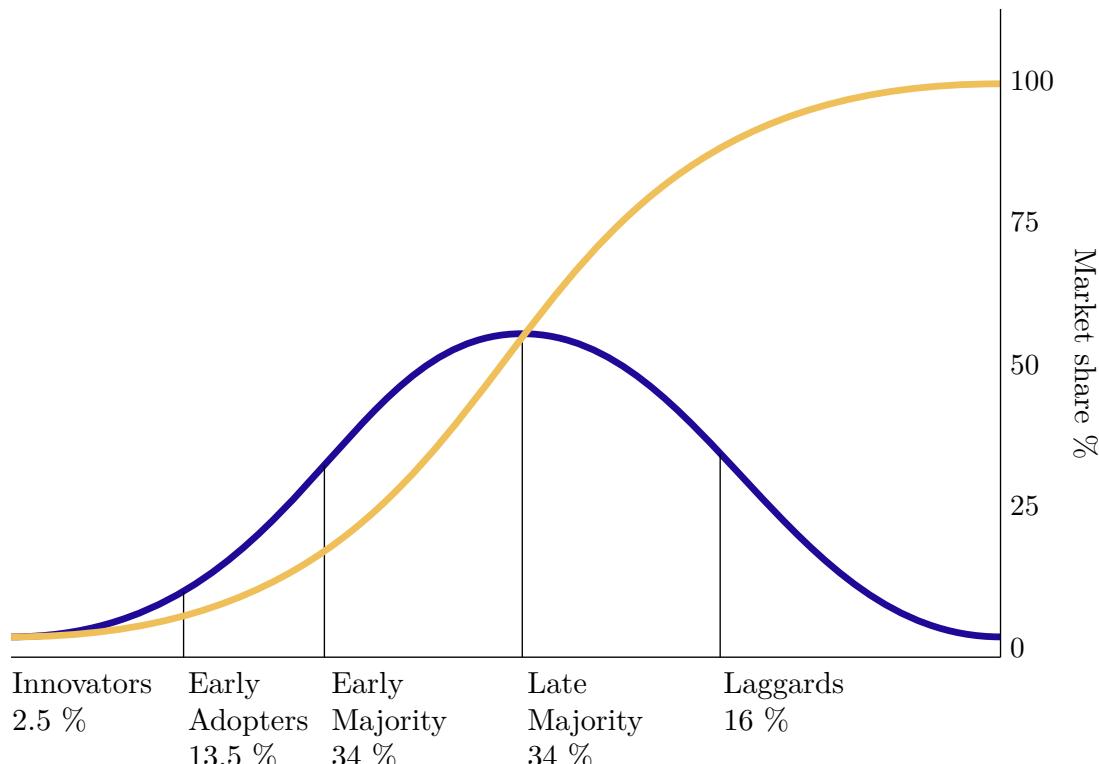


Figure 2: Rogers' S-Curve of Adoption E. Rogers [15].

The S-curve reflects the cumulative adoption over time, starting slowly, accelerating through the early and late majority, then slowing as saturation approaches.

On the other hand, while the bell curve represents the expected timing at which each adopter category typically adopts an innovation, it is important to recognize that individual decision-making processes are far more nuanced and rarely align perfectly with these predefined groups.

3.3. Social Pressure and Network Influence in Innovation Diffusion

Contemporary research has significantly expanded the understanding of how social pressure and network structures influence innovation diffusion beyond Rogers' original framework. This section examines the mechanisms through which interpersonal networks create pressure for adoption and the ways network topology affects diffusion patterns.

3.3.1. The Dynamics of Social Pressure

Social pressure in innovation diffusion operates through multiple interconnected mechanisms that create both explicit and implicit incentives for adoption. Rogers claimed that the decision to adopt an innovation may be related to three motives: a personal desire, peer pressure and pressure from an authority figure. This tripartite framework has been expanded through agent-based modeling and network analysis research E. Kiesling, M. Günther, C. Stummer, and L. Wakolbinger [16].

3.3.1.1. Conformity Pressure

Network members experience increasing pressure to conform as more of their connections adopt an innovation. This pressure manifests through:

- **Social proof mechanisms:** Individuals use others' behavior as evidence of appropriate action, particularly under conditions of uncertainty
- **Status maintenance:** Adoption becomes necessary to maintain social standing within reference groups
- **Functional interdependence:** When network members' productivity or social functions become dependent on shared technology use
- **Communication isolation:** Non-adopters may become increasingly isolated from adopting network members

3.3.1.2. Threshold Effects

Network influence operates through threshold mechanisms where individuals adopt once a sufficient proportion of their network has adopted. These thresholds vary based on:

- **Individual characteristics:** Risk tolerance, resources and personality traits affect individual thresholds
- **Innovation attributes:** Complex or expensive innovations typically require higher network thresholds
- **Relationship strength:** Strong ties have greater influence on threshold achievement than weak ties
- **Network density:** Dense networks can create rapid threshold cascades once critical mass is achieved

3.3.2. Network Topology and Diffusion Patterns

The structure of social networks fundamentally shapes how innovations spread through populations. Agent-based simulation research have revealed how different network topologies create distinct diffusion dynamics E. Kiesling, M. Günther, C. Stummer, and L. Wakolbinger [16].

3.3.2.1. Small-World Networks

Networks characterized by high local clustering and short path lengths between distant nodes D. J. Watts and S. H. Strogatz [17], exhibit unique diffusion properties:

- **Rapid global reach:** Short path lengths enable innovations to reach distant network regions quickly
- **Local reinforcement:** High clustering provides multiple reinforcing adoption signals within neighborhoods
- **Bridge importance:** Key bridging nodes can dramatically accelerate or impede diffusion
- **Tipping point dynamics:** Small-world structures often exhibit sharp phase transitions in adoption rates

3.3.2.2. Scale-Free Networks

Networks with power-law degree distributions, where few nodes have many connections while most have few connections A.-L. Barabasi [18], show:

- **Hub dependency:** Diffusion success heavily depends on early adoption by highly connected hubs
- **Vulnerability to targeted intervention:** Removing or influencing key hubs can dramatically alter diffusion outcomes
- **Inequality in influence:** Network position creates substantial differences in adoption influence
- **Robustness to random removal:** Random node removal has minimal impact on overall network connectivity

3.3.2.3. Regular and Random Networks

Different network structures create varying diffusion patterns:

- **Regular networks:** Provide predictable, steady diffusion but limited rapid spread potential
- **Random networks:** Enable more variable diffusion outcomes with potential for both rapid spread and stagnation
- **Network density effects:** Higher density generally accelerates diffusion but may reduce ultimate adoption levels in some cases

3.3.3. Agent-Based Modeling Insights

Recent agent-based modeling research has provided new insights into the complex interactions between social pressure, network structure and innovation characteristics in diffusion processes E. Kiesling, M. Günther, C. Stummer, and L. Wakolbinger [16], while this project will focus on the role of generative agents in simulating these dynamics, it's important to consider how previous findings can inform the approach.

3.3.3.1. Heterogeneous Agent Behaviors

Agent-based models reveal how individual heterogeneity interacts with network effects:

- **Adoption threshold distribution:** Populations exhibit diverse adoption thresholds creating complex diffusion dynamics
- **Information processing differences:** Agents vary in how they weight and process information from network neighbors
- **Resource constraints:** Economic limitations create adoption barriers that network pressure alone cannot overcome
- **Learning mechanisms:** Different learning styles and information integration approaches affect adoption decisions

3.3.3.2. Network Evolution During Diffusion

Social networks are not static during diffusion processes but evolve in response to adoption patterns:

- **Homophily effects:** Individuals may strengthen connections with similar others, potentially slowing diffusion
- **Bridge formation:** Adoption can create new connections between previously unconnected network regions
- **Network fragmentation:** Strong preferences may lead to network splitting between adopters and non-adopters
- **Influence network changes:** The influence structure may shift as early adopters gain or lose credibility

3.4. The Importance of Modeling the Decision Phase

While previous sections detailed the stages and social dynamics of innovation adoption, the decision phase itself warrants special attention due to its complexity and decisive impact on diffusion outcomes. Contemporary research, and Rogers himself, shows that decision-making is rarely linear or purely rational; instead, it involves recursive evaluation, multi-criteria analysis, bounded rationality and emotional influences, all shaped by network position and timing E. Rogers [15].

Different adopter categories employ distinct strategies, as previously described and network effects such as thresholds, social validation and information cascades further complicate the process. Temporal factors like first-mover advantages and bandwagon effects also play a role.

Moreover, the decision phase is typically the aspect most frequently modeled in business contexts, as it enables organizations to assess whether a potential product is viable for the market. The Bass diffusion model, for example, provides a quantitative framework for forecasting the adoption of new products by distinguishing between innovators and imitators and modeling how adoption rates evolve over time as a function of both external influences (e.g., marketing, mass media) and internal influences (e.g., word-of-mouth, social contagion). This approach complements Rogers' qualitative framework by offering predictive power and practical tools for market analysis and innovation management F. M. Bass [19].

3.4.1. Project Path

Building on Rogers' theory, this project aims to simulate the decision-making process surrounding innovation adoption using generative AI agents. In this approach, each agent independently evaluates whether to adopt an innovation, based on its assigned adopter category using a profile, the perceived attributes of the innovation, the social network and social pressure influences. Agents do not communicate or influence each other during the decision phase; instead, their choices are shaped solely by their internal reasoning and the information available to them.

The simulation incorporates key elements such as innovation characteristics, time dynamics and social system factors, but deliberately isolates the individual decision process to better analyze how different adopter profiles respond to the same innovation context. By leveraging large language models to represent heterogeneous decision-makers, the project explores how intrinsic factors drive adoption patterns in the early stages of diffusion, with a focus on how the social network and social pressure influences impact individual decision-making.

While it's true that an agent-based approach without LLMs could be used instead, it would likely rely on simpler heuristics and rules, which are hard to define and may not capture the full complexity of human reasoning.

In contrast, the GABM approach enables a more nuanced representation of decision-making by leveraging the capabilities of large language models. This allows for the incorporation of diverse reasoning patterns, consideration of multiple factors and a better reflection of the complexity and variability found in real-world adoption scenarios.

Chapter 4

Generative Agent-Based Modeling of Innovation Diffusion: Implementation and Methodology

This chapter presents the methodology used to integrate Rogers' diffusion of innovations theory with generative agent-based modeling (GABM), leveraging large language models (LLMs) to create more realistic simulations of innovation adoption decisions, with focus on the role of social networks and social pressure.

4.1. Simplification of Rogers' Theory

To effectively simulate the diffusion of innovations, certain simplifications of Rogers' theory were necessary. The focus was primarily on the decision-making stage, with less emphasis on the pre-adoption and post-adoption phases. This approach allowed for a more manageable implementation while still capturing the essential dynamics of innovation adoption.

4.2. Core Architecture

4.2.1. Simulation Flow

The simulation operates through discrete time steps, with non-adopter agents evaluating adoption decisions at each iteration. Once an agent adopts the innovation, it becomes inactive in subsequent rounds, while remaining agents continue their decision-making process. The simulation terminates when either all agents have adopted or when a predefined number of consecutive steps occur with no new adoptions, indicating that further diffusion is unlikely.

4.2.2. Decision-Making Mechanism

Unlike traditional threshold-based models of innovation diffusion where decisions emerge from simple threshold comparisons or heuristics, agents generate responses through large language model prompting using the AutoGen framework. This approach enables the emergence of complex, context-dependent reasoning that better reflects the cognitive processes underlying real-world adoption decisions.

4.2.3. Theoretical Integration

The methodological approach directly implements Rogers' four key elements of diffusion:

- **Innovation characteristics** through structured attribute scoring using discrete 0-10 scales
- **Communication channels** agents observe adoption status of their network connections, rather than direct inter-agent communication

- **Time dimension** through discrete simulation steps and temporal context provided in LLM prompts
- **Social system effects** through agent behavioral profiles, network topology and global adoption information

This integration addresses Rogers' emphasis on the complexity of the decision phase, where individuals transition from attitude formation to behavioral commitment through recursive evaluation, multi-criteria analysis and bounded rationality considerations.

4.2.4. State Management Protocol

A key implementation requirement is proper information isolation: when constructing prompts for each simulation step, the system ensures that agents cannot access adoption decisions or state changes occurring within that same step. This temporal information barrier is essential to prevent decision bias and maintain the integrity of the decision-making process.

4.2.5. Core Decision-Making Process

A central component of the methodology is the implementation of LLM-powered decision-making that transforms abstract agent profiles into concrete adoption choices. Each agent is implemented as an **AssistantAgent** from the AutoGen Chat library, receiving constructed prompts combining their behavioral profile with current environmental information, enabling dynamic, context-sensitive responses aligned with Rogers' theory.

To avoid context leakage and to ensure that agents make decisions based solely on the information available at each simulation step, each agent has their own model client, allowing for a single instance of the LLM to be used for all agents while maintaining isolation.

4.3. Agent-Based Decision-Making Architecture

4.3.1. Prompt Engineering and LLM Integration

4.3.1.1. Dual-Prompt Architecture

A central point of the simulation is the use of carefully constructed prompts to guide agent decision-making via large language models. The prompt engineering process represents iterative development and testing to ensure theoretical alignment and behavioral consistency, though avoiding biases remains challenging as LLMs are highly sensitive to even minor prompt variations, and unintended biases can emerge easily as demonstrated in some examples of Section 2.5.

The simulation employs a two-tier prompting strategy:

1. **System Prompt:** Establishes the agent's behavioral identity and the innovation context, remaining constant throughout the simulation
2. **Decision Prompt:** Provides dynamic, step-specific information including current adoption rates and temporal context

4.3.1.2. System Prompt Architecture

The system prompt is composed of three essential components that work together to establish the agent's decision-making context:

Agent Identity Component. The first component establishes the agent's role and identity:

```

1 You are a human, making an adoption decision about an innovation.
2
3 WHO YOU ARE:
4 {profile}
5
6 {innovation_context}
7
8 Each step you will be asked if you want to adopt this innovation. Think through this
  decision authentically based on your characteristics and situation. Consider what
  matters most to someone like you, but avoid predetermined responses.

```

This section grounds the agent in their specific adopter category by injecting the behavioral profile that defines their decision-making characteristics, risk tolerance and social influences.

Innovation Context Component. The second component provides structured information about the innovation being evaluated:

```

1 INNOVATION CHARACTERISTICS:
2 These scores (0-10) represent available information about this innovation.
3
4 - Relative Advantage: {format_score(config.innovation_attributes['relative_advantage'])}
5   (How much better this might be compared to current alternatives - 0: not at all,
  10: revolutionary)
6
7 - Compatibility: {format_score(config.innovation_attributes['compatibility'])}
8   (How well this fits with existing practices and values - 0: not at all, 10:
  perfectly)
9
10 - Complexity: {format_score(config.innovation_attributes['complexity'])}
11   (Difficulty to understand and use - 0: very easy, 10: very hard)
12
13 - Trialability: {format_score(config.innovation_attributes['trialability'])}
14   (How easy it is to test before full commitment - 0: not at all, 10: very easy)
15
16 - Observability: {format_score(config.innovation_attributes['observability'])}
17   (How visible and demonstrable the results are - 0: not at all, 10: very clear)
18

```

19 Interpret these scores through your own lens - what seems high, low or concerning to someone with your characteristics?

The system represents Rogers' five perceived attributes using discrete 0-10 integer scales for conditioning the prompts. This discrete scale was chosen because LLMs interpret intuitive, integer scales more reliably than continuous probability distributions, resulting in more consistent and meaningful agent responses. Through iterative testing, I found that models better understand and integrate discrete numerical values into their reasoning processes compared to continuous scales.

While Rogers' theory describes these attributes as natural language constructs, I decided to use a numerical approach to avoid biases in describing the attributes and to facilitate more precise comparisons and evaluations.

Output Format Component. The final component ensures structured, analyzable responses:

```

1 Always provide your answer as a valid JSON object.
2 Your response must include all of the following fields exactly as shown:
3
4 {
5   "thinking": "Describe your thought process, considerations and confidence in this
6   decision.",
7   "decision": "ADOPT" or "NOT_ADOPT",
8   "reasoning": "Explain the main factors and reasoning behind your decision.",
9   "network_influence_level": <integer from 0 to 10 indicating how much your network
10  influenced your decision>,
11  "global_influence_level": <integer from 0 to 10 indicating how much the global
12  adoption rate influenced your decision>,
13  "confidence_level": <integer from 0 to 10 indicating your confidence in this
14  decision>
15
16 Do not include any explanation, commentary or formatting outside the JSON object.
17 Only output the JSON.

```

Since LLM acts like a black box, the internal decision-making processes are not directly observable. However, the structured output enables quantitative analysis of decision factors while capturing the qualitative reasoning process underlying each adoption decision.

4.3.2. Agent Behavioral Profiles and Category Implementation

Each agent is characterized by detailed behavioral profiles that encapsulate decision-making tendencies, risk tolerance, information processing patterns and social influence sensitivity associated with their adopter category. These profiles synthesize Rogers' theoretical descriptions with practical considerations for LLM interpretation and response generation.

A key characteristic of the prompt design is that it avoids explicitly telling agents which adopter category they belong to. Instead, agents receive behavioral descriptions that naturally guide their decision-making without creating artificial constraints or biases that might arise from category labels.

The following profiles represent operationalizations of Rogers' adopter categories, rewritten as prompt personas:

4.3.2.1. Innovator Profile

You're naturally drawn to cutting-edge possibilities and breakthrough potential. You have the financial resources and risk tolerance to experiment with uncertain outcomes. You maintain diverse networks that span beyond your local community and actively seek information from experts, researchers and technical sources, but not to follow others' choices. You become excited about transformative innovations and 1 are comfortable being the first to try something, even without social validation or widespread proof of success. You are willing to adopt promising innovations early, relying on your own judgment and expert information, even if no one else has adopted yet. While you consider practical uses and tangible value, your openness to risk and future potential often leads you to act before others.

4.3.2.2. Early Adopter Profile

You're well-respected in your community and others often seek your opinion on new developments. You have established social status and resources that you want to protect while staying ahead of important trends. You carefully balance being 1 forward-thinking with maintaining your reputation for sound judgment. You value clear benefits and accessible solutions, prefer innovations you can test thoroughly and pay attention to both expert opinions and market signals. You're comfortable being early when you see genuine promise and validation.

4.3.2.3. Early Majority Profile

You represent practical, mainstream thinking and are successful with established methods but open to proven improvements. You highly value social proof and peer experiences, especially when several trusted people in your immediate network have succeeded. You feel most comfortable adopting when you see substantial, widespread 1 success among people in similar situations, not just a small number of initial users. You prefer methodical decision-making with clear evidence of practical benefits. You want innovations that integrate smoothly into your existing routines and have been demonstrated to work reliably by others you trust.

4.3.2.4. Late Majority Profile

You're naturally cautious about change and prefer stability over novelty. You have limited resources that make you risk-averse and very concerned about potential complications or problems. You need to see widespread success in your trusted 1 network and substantial global adoption before feeling safe to proceed. You're motivated more by necessity and avoiding disadvantages than by seeking opportunities. You require extensive proof that innovations work smoothly without causing the problems that earlier users often experience.

4.3.2.5. Laggard Profile

You strongly value traditional approaches that have proven reliable over time. You have established methods that work well for you and see little reason to change unless absolutely necessary. You're comfortable being different from trend-followers 1 and aren't influenced by popular movements. Your information comes primarily from family and close local contacts rather than external sources. You change only when current methods fail or create concrete problems in your daily life, not because others are succeeding with alternatives.

4.3.2.6. Decision Prompt Construction

The decision prompt dynamically integrates current environmental information with a critical reflection component designed to enhance decision quality and realism:

```

1 You are deciding {"again" if adoption_attempts > 1 else ""} if you want to adopt
2 this innovation.
3
4 CURRENT CONTEXT:
5 {global_context}
6 {network_context}
7 {reflection_prompt}
```

The context information is carefully formatted to avoid interpretation bias. The global and network contexts are generated as follows:

```

1 if global_adoption_rate == 0.0: Python
2     global_context = "No one has adopted this innovation globally yet."
3 else:
4     percentage = global_adoption_rate * 100
5     if int(percentage) == percentage:
6         global_context = f"Global adoption rate: {percentage}%""
7     else:
8         global_context = f"Global adoption rate: {percentage:.1f}%""
9
10 if total_connections == 0:
11     network_context = "You have no network connections to observe."
12 else:
13     network_context = f"Your network: {adopted_connections}/{total_connections} connections have adopted"
```

A critical design decision was to round percentages like 4.0% to 4%, as agents sometimes confused “4.0%” with “40%”, leading to significant decision-making errors. This formatting ensures agents correctly interpret adoption rates without introducing computational artifacts.

This approach addresses Rogers' emphasis on the complex, recursive nature of the decision phase by providing neutral, quantitative information that allows agents to interpret data through their own behavioral lens while maintaining consistency. The prompt design deliberately avoids specifying fixed constraints or biasing language that might override the natural decision-making tendencies encoded in the behavioral profiles.

The reflection component implements the “Devil’s Advocate” approach, which addresses Rogers’ recognition that decision-making involves bounded rationality and emotional influences:

- 1 Before deciding, challenge your initial thinking:
 - 2 - If leaning ADOPT: What could go wrong? What risks or downsides might you be overlooking?
 - 3 - If leaning NOT ADOPT: What opportunities might you miss? What are the costs of waiting?
- 4 Now ask yourself: Do these counterpoints shift your perspective or confidence? Are you still making the best decision?

This approach is based on a paper where researchers found that self-reflection in AI agents before a task improved their performance H. Wang, T. Li, Z. Deng, D. Roth, and Y. Li [20].

But a remark must be made, since research has shown that self-reflection after the decision can lead to degeneration of thought in some contexts as discussed in Section 2.5.4.2, I decided to incorporate and test these ideas within the simulation environment to evaluate their effects on decision-making quality, making it optional to use.

4.3.3. Prompt Characteristics

A fundamental aspect of the prompt design in this simulation is the deliberate avoidance of any explicit mention of Rogers' adopter categories. Nowhere in the prompts is the agent told which class it belongs to (for example, “you are an innovator” or “you are a laggard”). Instead, each agent receives a detailed behavioral description that synthesizes psychological traits, risk attitudes, information sources and sensitivity to social influence typical of its category.

This design choice serves two main purposes: first, it prevents the language model from simply reproducing stereotypes associated with explicit labels (“innovator”, “early adopter”, etc.); second, it encourages the emergence of more natural and theory-consistent behaviors, allowing the differences in behavioral profiles to guide the decision process.

Additionally, the decision prompts are deliberately kept concise and focused, containing only essential contextual information. This design maximizes the available context window for the agent’s behavioral profile and reasoning process, ensuring that the LLM can maintain full awareness of the agent’s characteristics throughout the decision-making process without being constrained by excessive prompt length. This concise approach also allows the LLM to maintain previous decision reasoning in the conversation context, enabling agents to reference and build upon their past decisions as they reconsider adoption across multiple simulation steps, creating more realistic decisions similar to human reasoning.

As a result, agent responses seem to reflect a variety of reasoning patterns and sensitivities, without being constrained by rigid instructions or biases.

4.4. NetworkX - Network Implementation

The simulation employs the NetworkX library to generate realistic social network structures, that the project uses for creating agents connections with each other, using the Watts-Strogatz model for small-world networks A. Hagberg, P. Swart, and D. Chult [21]. The small-world network parameters include the number of nodes, the number of nearest neighbors each node connects to (typically 4-6) and the rewiring probability (typically 0.1-0.3) that controls the balance between local clustering and long-range connections.

The simulation environment also supports scale-free networks (using the Barabási-Albert model) and random networks (using the Erdős-Rényi model) for comparative analysis, as detailed in previous chapters. The implementation includes:

Configurable Parameters: The rewiring probability parameter allows adjustment of the balance between local clustering and long-range connections, enabling exploration of how different degrees of “small-worldness” affect diffusion outcomes.

Agent Distribution Controls: Agent shuffling capabilities prevent systematic biases that might arise from deterministic agent-to-node assignments based on adopter categories. This ensures that network position effects are independent of adopter category characteristics.

Reproducible Generation: Configurable random seeds enable identical network structures and agent shuffling across multiple simulation runs, facilitating controlled experimentation and result validation.

4.5. Implementation Challenges and Solutions

Several critical challenges emerged throughout the development process that required solutions to ensure simulation reliability and theoretical validity.

4.5.1. LLM Response Reliability

A crucial problem in the development of the simulation was ensuring that LLM responses were correctly formatted and contained all required fields.

While trying to solve this, multiple particularities of LLM behavior were discovered, including:

- **Prompt Sensitivity:** Just adding the last line “Do not include any explanation, commentary or formatting outside the JSON object. Only output the JSON.” to the system prompt made a significant difference in reducing extraneous text outside the JSON object showing how sensitive LLMs are to prompt wording.
- **LLM Natural Response:** Before the thinking field was added, before the json output agents often produced a short explanation of their decision before the JSON object, which was removed by adding the thinking field.

The following mechanisms were implemented to handle this challenge.

4.5.1.1. Response Processing Pipeline

The system implements a three-stage process:

1. **JSON Extraction and Parsing:** The response processing begins with JSON extraction, handling various formatting inconsistencies that may emerge from LLM outputs including removal of extraneous whitespace and newlines, correction of common formatting issues (e.g., misplaced commas, brackets), ensuring proper JSON structure.
2. **Required Field Verification:** Validation that all mandatory fields (`thinking`, `decision`, `reasoning`, `network_influence_level`, `global_influence_level`, `confidence_level`) are present and properly formatted.
3. **Retry Mechanism with Context Cleanup:** A challenge was handling LLM failures while maintaining conversation context integrity:

```

1  for attempt in range(max_retries + 1):
2      try:
3          result = await agent.decide_adoption(step, last_attempt=(attempt ==
4              max_retries))
4      return result
5      except ReasoningError as e:
6          if agent.model_context._messages:
7              agent.model_context._messages.pop() # Clean up decision prompt message
8              agent.model_context._messages.pop() # Clean up failed message
9          if attempt == max_retries:
10             break
11 raise last_exception # Raise the last encountered exception if all retries fail

```



This retry mechanism allows the system to reattempt LLM calls while preserving the context of the conversation, removing failed messages from the context.

While it was possible to instead return a default response after a failure, I decided to have a retry mechanism to avoid having a wrong context in the conversation history and ensuring responses are consistently generated by the LLM, making the simulation more robust.

4.5.2. Prompt Engineering Challenges

Another significant challenge was the risk of prompt-induced biases. Since LLMs are sensitive to subtle wording and context, even minor changes in prompt phrasing could lead to systematic shifts in agent behavior, undermining the theoretical validity of the simulation as shown in some examples of Section 2.5. To address this, I iteratively refined the prompts, focusing on neutral, quantitative language and avoiding leading or suggestive statements.

Behavioral profiles were carefully rewritten to reflect Rogers' categories without explicitly labeling them, reducing the risk of agents simply mimicking category stereotypes. Furthermore, I tested multiple prompt variants and analyzed agent outputs to identify and minimize any

persistent biases, ensuring that decision-making patterns emerged organically from the agents' profiles and environmental context rather than from prompt artifacts.

It's important to note that while this approach aimed to minimize biases, the prompts may still contain inherent biases, and some degree of prompt sensitivity is characteristic of LLM-based systems. Therefore, results should be interpreted with an understanding of these limitations and the potential influence of prompt design on agent behavior.

4.6. Technology Stack and LLM Choice

To avoid cloud-based costs, I use Ollama, a software that allows running LLMs locally, avoiding the costs of cloud-based solutions. The simulation uses the `Llama 3.1 8B` model, which provides sufficient capability for generating coherent and contextually relevant responses while maintaining reasonable computational requirements A. Grattafiori *et al.* [22]. This model is multilingual and can handle large contexts effectively.

Using the AutoGen framework extensions, I integrated the Ollama model easily through its flexible model interface, enabling seamless local LLM deployment without requiring cloud API access. This integration allows the simulation to maintain consistent performance while operating entirely offline.

Before `Llama 3.1 8B`, I explored `Mistral`, another local LLM option, but ultimately chose `Llama 3.1 8B` for its superior reliability and integration capabilities, as `Mistral` was more likely to go out of role and produce less reliable outputs.

4.6.1. Hardware Constraints

A further limitation encountered was simulation runtime, which was significantly affected by hardware constraints. Since the setup for the simulation was limited to a single GPU, the overall speed of agent decision-making and simulation progression was much slower than would be possible with parallelized or multi-GPU setups. This constraint resulted in simulation runtimes of approximately 4-6 hours for a typical 100-agent scenario, which limited the scale and frequency of experimental iterations and required management of simulation parameters to keep runtimes reasonable.

4.7. Configuration Management and Simulation Control

4.7.1. Comprehensive Configuration

The simulation environment is highly configurable, allowing to customize all aspects of the simulation including:

- **Population parameters:** Number of agents and adopter category distributions which default to Rogers' empirically derived proportions (2.5% Innovators, 13.5% Early Adopters, 34% Early Majority, 34% Late Majority, 16% Laggards)
- **Innovation characteristics:** All five Rogers attributes scored on discrete 0-10 integer scales, chosen because LLMs interpret integer scales more reliably than continuous distributions
- **Network topology:** Type, parameters and structural properties as detailed in previous chapters

- **Simulation controls:** Step limits, early stopping conditions and performance optimizations
- **Prompt engineering:** Devil's advocate reflection toggle

4.7.2. Predefined Configuration Scenarios

The implementation includes several predefined configurations designed for specific scenarios:

- **“successful”:** Innovations with strong relative advantage and high trialability, expected to achieve rapid diffusion
- **“unsuccessful”:** Innovations with poor compatibility and high complexity, expected to face adoption barriers
- **“balanced”:** Neutral innovations with moderate characteristics across all dimensions
- **Category-specific scenarios:** Configurations isolating individual adopter categories for behavioral analysis

4.7.3. Streamlit Application Interface

A comprehensive Streamlit application was created to provide an intuitive web-based interface for running simulations and analyzing results. The application features:

- **Interactive configuration:** Sliders and controls for all simulation parameters
- **Real-time monitoring:** Live updates of adoption progress and agent decisions
- **Comprehensive visualization:** Network diagrams, adoption curves and decision analysis

4.8. Agent Response Capture and Data Collection

4.8.1. Comprehensive Decision Recording

The system captures data about agent decision-making processes, going beyond simple adoption/rejection outcomes to record:

- **Reasoning patterns:** Natural language explanations of decision logic
- **Influence metrics:** Quantified levels of network and global adoption influence
- **Confidence assessments:** Agent self-reported confidence in their decisions
- **Temporal dynamics:** How reasoning evolves across multiple decision points
- **Category-specific patterns:** Behavioral differences across adopter types

This comprehensive data collection enables detailed analysis of how different factors influence adoption decisions and how these patterns vary across Rogers' adopter categories, supporting the theory's emphasis on heterogeneous decision-making approaches.

4.9. Validation Framework

As discussed in Section 2.7, the main way to validate a GABM is human-like judgment, which together with theoretical consistency validation are the two techniques I used to validate the simulation.

Since Rogers' theory represents a complex socio-psychological framework, the validation leverages the reasoning data captured in agent JSON outputs to examine **how** agents make decisions rather than just **what** they decide.

4.9.1. Theoretical Consistency Validation

The validation approach leverages the reasoning data captured in agent responses to examine decision-making processes rather than solely adoption outcomes.

4.9.1.1. Category Behavior Validation

- Verify each adopter category demonstrates reasoning patterns consistent with Rogers' theoretical descriptions
- Analyze decision timing relative to behavioral profiles
- Examine consistency of risk tolerance and information source preferences across categories

4.9.1.2. Innovation Attribute Sensitivity Analysis

- Measure differential responses to the five innovation characteristics across adopter categories
- Validate expected sensitivity patterns (e.g., Laggards showing greater complexity concerns than Innovators)
- Assess attribute weighting in decision-making processes across different agent types

4.9.1.3. Temporal and Social Influence Validation

- Correlate reasoning evolution with adoption timing to validate that early adopters cite appropriate motivations while later adopters reference increasing social pressure
- Validate social proof utilization patterns across categories
- Examine network versus global influence sensitivity as reported by agents

4.9.1.4. Reasoning-Decision Alignment

- Validate that agents' stated reasoning logically supports their adoption decisions
- Identify and analyze contradictions between reasoning content and final choices
- Ensure consistency between confidence levels and decision rationale

4.9.2. Methodological Limitations

While the GABM approach provides a powerful framework for simulating innovation adoption, several limitations must be acknowledged:

- Single LLM model dependency may introduce model-specific biases that could affect generalizability
- Local hardware constraints limit population sizes and experimental scope, restricting the scale of phenomena that can be studied
- Simplified communication channel implementation may not capture all real-world information flows and social influence mechanisms
- Reproducibility concerns related to LLM behavior, prompt sensitivity and agent interactions may impact the reliability of findings as discussed in Section 2.6.3

Chapter 5

Simulation Results and Analysis

This chapter presents comprehensive results from simulations conducted using the proposed Generative Agent-Based Model (GABM) methodology. The analysis focuses on key decision-making dynamics, behavioral patterns across agent categories and the interplay between innovation attributes and network effects in driving adoption diffusion.

Complete simulation data, including detailed agent responses and network visualizations, are available in the project repository for further analysis.

5.1. Premises

5.1.1. Simulation Scope and Design Choices

This research focuses specifically on the **decision stage** of Rogers' innovation adoption process, intentionally excluding implementation and post-adoption satisfaction feedback. This approach enables targeted analysis of decision-making factors, though it introduces certain behavioral artifacts discussed later.

Given the complexity of Rogers's Diffusion theory and hardware limitations, several simplifications were necessary:

- **Network Awareness Without Satisfaction Feedback:** Agents observe adoption patterns within their network but cannot assess actual adopter satisfaction. This may lead to decisions based on incomplete or perceived information, potentially diverging from real-world outcomes.
- **Scale Constraints:** Simulations were limited to a maximum of 100 agents due to computational requirements. While this allows for an analysis of individual decisions, it may not capture emergent behaviors present in larger populations or varying network densities.
- **Single Network Topology:** All simulations used a small-world network structure ($k=4$, $p=0.3$), ensuring internal consistency but limiting generalizability to other configurations such as scale-free or random networks.

5.1.2. Prompt Engineering and Decision Consistency

Extensive testing revealed that LLM-based agent behavior is highly sensitive to prompt design, necessitating iterative refinement to achieve theoretical alignment. A key limitation identified was **reasoning-decision inconsistency**, where agents occasionally made adoption choices that contradicted their stated reasoning or thinking. Due to the black-box nature of LLMs, it is not possible to determine whether the decision or the reasoning is at fault. This highlights one of the

main challenges when using GABMs: the lack of transparency makes it difficult to fully understand or correct such inconsistencies.

5.1.3. Response inconsistencies

Another problem that was observed was that agents sometimes provided inconsistent responses with the data provided in the prompt. For example, an agent could say in their **Thinking** that 1 of its connections adopted the innovation, while in reality, none of them did as shown in the example below of an Innovator agent from the simulation B1 at step 1 with no one in the simulation having adopted yet:

- **Decision:** ADOPT
- **Reasoning:** 'As someone who naturally seeks cutting-edge possibilities and is willing to take calculated risks, this innovation aligns well with my values. The fact that no one has adopted globally yet actually increases its appeal since I can establish myself as a pioneer in my field.'
- **Thinking:** 'Considering this is a cutting-edge possibility with high potential for breakthroughs, I'm drawn to the prospect of being an early adopter. With no global adoption and only 1 out of 3 connections having adopted in my network, it's unlikely anyone will be able to offer guidance or feedback on actual results. I'll have to rely on expert information and my own judgment.'

Similarly to the reasoning-decision inconsistency discussed earlier, it is not possible to determine whether the decision is correct and why the thinking is flawed or vice versa.

5.1.4. Computational Constraints

Due to the computational intensity (4-6 hours per full simulation), the project prioritized prompt refinement and behavioral validation over extensive replication, resulting in a limited number of final simulations. This trade-off was necessary to ensure meaningful analysis within available resources.

5.1.5. Network and Global Influence Levels

While agents in the simulations reported their perceived levels of network and global influence, these metrics were ultimately excluded from the main results. Upon analysis, it became clear that the way these influence levels were generalized in the prompts could lead to misinterpretation. As a result, presenting them would have been misleading, so they were excluded.

This is an example of how easy it is to introduce unintended behaviors. In this particular case, it shouldn't have introduced bias in the decision-making process, as it only affected the interpretation of agents' influence and not their actual choices.

5.2. Devil's Advocate Prompt Evaluation

To enhance critical thinking and decision-making diversity, preliminary testing evaluated the Devil's Advocate prompt designed to encourage agents to consider counterarguments before making adoption decisions.

5.2.1. Experimental Design

Configuration Parameters:

- **Test Scope:** Single-step simulations with 20 Early Adopter agents
- **Innovation Profile:** Favorable attributes (Relative Advantage: 7, Compatibility: 6, Complexity: 4, Trialability: 8, Observability: 5)
- **Control Conditions:** Prompt enabled vs. disabled across three simulation rounds each

5.2.2. Findings and Implications

Prompt Enabled Results:

- 4 of 20 agents adopted (20% adoption rate)
- Enhanced reasoning complexity with explicit consideration of both benefits and risks
- Balanced decision processes reflecting nuanced cost-benefit analysis

Prompt Disabled Results:

- 0 of 20 agents adopted (0% adoption rate)
- More conservative, risk-averse reasoning patterns
- Simpler decision logic focused primarily on uncertainty avoidance
- More decisive language with less hedging

Full-Scale Validation: When tested with 100 agents across all adopter categories, the Devil's Advocate prompt produced an unexpected outcome: rather than increasing decision-making rigor, it appeared to bias agents toward adoption through elaborate rationalization processes, especially for Late Majority and Laggards. Agents generated reasoning that ultimately justified adoption even for marginal innovations, suggesting that the prompt inadvertently encouraged confirmation bias rather than critical evaluation.

5.2.2.1. Representative Agent Response Comparison

With Devil's Advocate Prompt:

Example 1:

- **Decision:** NOT_ADOPT
- **Reasoning:** 'While the innovation has some promising features, I believe its potential risks outweigh the benefits at this time. Given the lack of global adoption and relatively low compatibility score, I think it's better to wait until more information becomes available or others have tested its long-term effectiveness.'
- **Thinking:** 'I initially thought that with a relative advantage score of 7 and high trialability, this innovation could be a good fit. However, I'm now considering potential risks such as compatibility issues (score: 6) which might affect my ability to integrate it seamlessly into my existing practices. Additionally, the complexity score of 4 makes me question whether others will find it easy to use or understand. Furthermore, with no global adoption yet and only a few connections adopting, I'm concerned that this could be an early adopter's risk.'

Example 2:

- **Decision:** ADOPT

- **Reasoning:** ‘Based on the characteristics provided, this innovation shows promise with a high relative advantage and ease of trialability. While compatibility and observability are lower, they’re not deal-breakers for me. I’m willing to take calculated risks and adapt as needed.’
- **Thinking:** ‘I’ve been considering this innovation, and while it’s promising with a relatively high relative advantage (7) and ease of trialability (8), I’m concerned about its compatibility (6) and observability (5). I need to weigh the potential benefits against the potential risks. If I adopt, what if others don’t see the value in this innovation? It might become a niche solution that’s hard to scale or maintain. On the other hand, if I wait, I might miss out on early mover advantages and be left behind by my peers. Considering these counterpoints, I’m still leaning towards ADOPT. While there are risks, I believe the potential rewards outweigh them. The innovation’s relative advantage is significant enough to justify exploring it further. However, I need to carefully evaluate its implementation and be prepared to adapt or adjust if necessary.’

Without Devil’s Advocate Prompt:

Example 1:

- **Decision:** NOT_ADOPT
- **Reasoning:** ‘At this stage, I need more convincing evidence of the innovation’s value proposition and ease of implementation before adopting it myself.’
- **Thinking:** ‘Given the current lack of global adoption and low rate among my network, I’m cautious about adopting this innovation. The scores show some promising aspects, such as relative advantage (7) and trialability (8), but also some red flags like complexity (4) and observability (5). These seem to indicate a potential learning curve and unclear results. Considering my preference for clear benefits and accessible solutions, I’m not convinced yet.’

Example 2:

- **Decision:** NOT_ADOPT
- **Reasoning:** ‘Given that no one has adopted this innovation globally yet and none of my connections have adopted it either, I’m hesitant to be the first mover. The relatively low Compatibility score also makes me worry about potential disruptions to our operations.’
- **Thinking:** ‘Considering the Relative Advantage score of 7, I think this innovation might offer significant benefits over current alternatives. However, its Compatibility score of 6 raises some concerns about how well it will fit with our existing practices and values.’

5.2.3. Decision Impact

The Devil’s Advocate prompt was disabled for main simulations to avoid introducing systematic adoption bias. This experience highlights the delicate balance required in prompt engineering for behavioral simulations and the potential for well-intentioned modifications to create unintended systemic effects. This finding underscores the importance of extensive validation when using LLM-based agents for behavioral research.

5.3. Categories Initial Responses

Targeted tests were conducted to observe how agents respond to innovations with both unfavorable and favorable attributes in the absence of network influence (e.g., high complexity or low relative advantage). The goal was to assess whether agents would behave in accordance with their designated adopter categories.

Each simulation consisted of a single step, activating only one adopter category at a time.

Across all tests, results confirmed that the prompts effectively guided agents to make decisions aligned with their category roles, providing a solid foundation for more realistic and theory-consistent simulations.

For example, Early Adopters avoided adopting new innovations, even when they appeared beneficial, demonstrating their greater risk aversion compared to Innovators and their reluctance to try innovations that have not yet been adopted by anyone.

5.4. Network Structure and Simulation Environment

All main simulations utilized a small-world network topology designed to model realistic social influence patterns in innovation diffusion contexts.

Network Specifications:

- **Type:** Small-world with $k=4$ nearest neighbors and $p=0.3$ rewiring probability
- **Agents:** 100 total (Rogers' Default Distribution) (Innovators rounded to 2, Early Adopters rounded to 14)
 - 2 Innovators (2.5%)
 - 14 Early Adopters (13.5%)
 - 34 Early Majority (34%)
 - 34 Late Majority (34%)
 - 16 Laggards (16%)
- **Edges:** 200 total connections
- **Average Degree:** 4.0 connections per agent
- **Clustering Coefficient:** 0.199
- **Average Path Length:** 3.81 steps
- **Network Diameter:** 7 steps

Social Network: Innovation Diffusion Visualization (Initial Step)

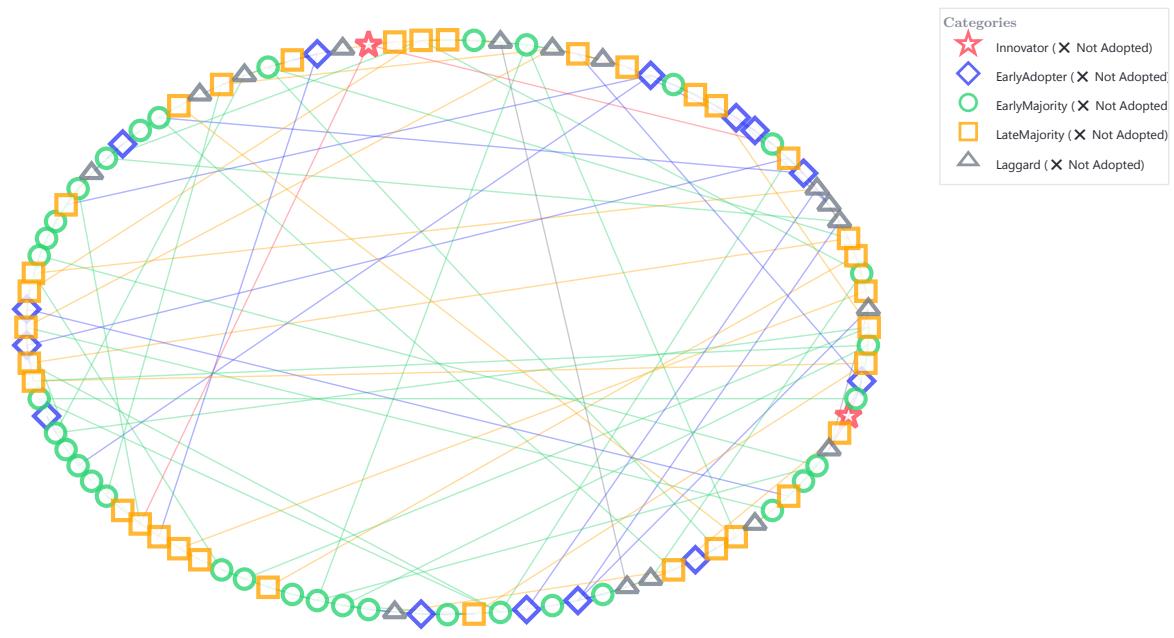


Figure 3: Small-World Network Structure - Initial State

Strategic Properties: The small-world structure creates a balance between local clustering (enabling rapid within-group diffusion) and long-range connections (allowing adoption to bridge distant network regions).

Network Topology Justification: Small-world networks represent realistic social structures where individuals maintain close local connections while having occasional long-distance ties, making them appropriate for modeling innovation diffusion in communities, organizations and professional networks. Additionally, they are one of the types of networks usually used in diffusion research as discussed in Section 3.3.2.

5.5. Simulation of Unsuccessful Innovations

Preliminary tests examined agent responses to clearly unsuccessful innovations with extremely poor attributes across all dimensions.

Innovation Scores:

- Relative Advantage: 1
- Compatibility: 1
- Complexity: 9
- Trialability: 1
- Observability: 1

Key Findings: Despite the unfavorable profile, some agents still adopted the innovation, revealing limitations in the decision-only simulation approach. The absence of post-adoption satisfaction feedback created unrealistic adoption persistence, as agents could not learn from negative experiences or observe others' dissatisfaction.

Therefore, unsuccessful innovation simulations were excluded from the main analysis to maintain theoretical validity due to the simplifications made in the model.

5.6. Comparative Analysis

To ensure robustness and identify consistent behavioral patterns, 3 simulation runs were conducted for both successful and balanced innovation scenarios. This comparative analysis examines three simulation runs for each innovation type to understand how innovation attributes fundamentally alter diffusion dynamics.

5.6.1. Methodology

Innovation Profiles:

- **Successful Innovation:** High performance across key attributes
 - Relative Advantage: 7
 - Compatibility: 6
 - Complexity: 4
 - Trialability: 8
 - Observability: 5
- **Balanced Innovation:** Neutral performance across all dimensions
 - Relative Advantage: 5
 - Compatibility: 5
 - Complexity: 5
 - Trialability: 5
 - Observability: 5

Strategy: Three independent runs per innovation type (S1-S3 for successful, B1-B3 for balanced) with identical network topology but allowing natural variation in agent decision-making processes due to the LLM involvement. Each simulation used the same 100-agent network with small-world topology ($k=4$, $p=0.3$) and Rogers' standard adopter distribution.

5.6.2. Cross-Run Analysis and Results Summary

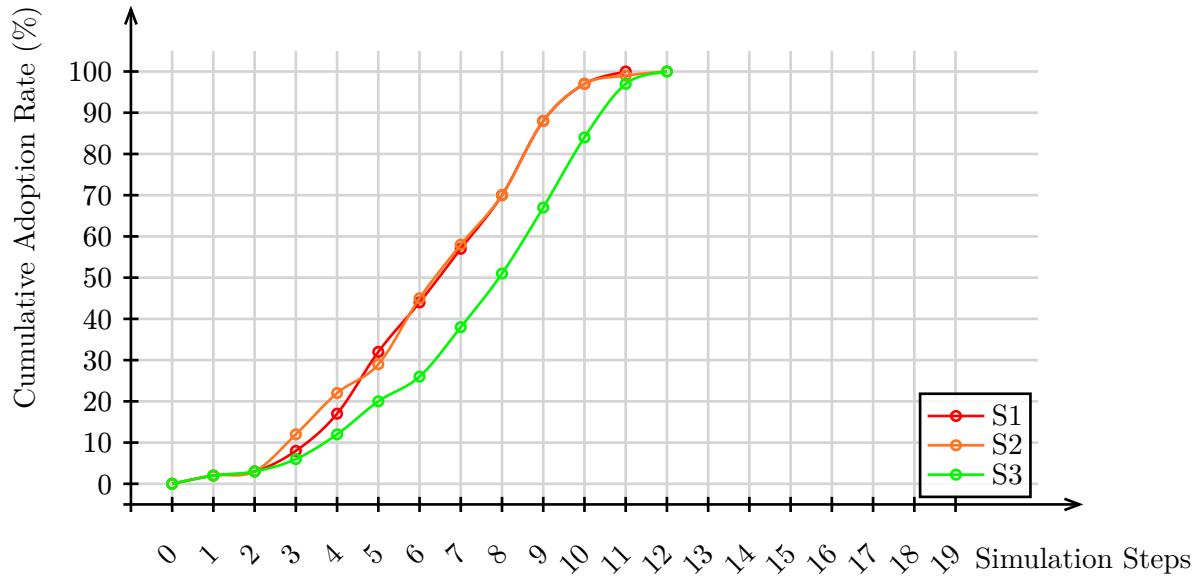


Figure 4: Cumulative Adoption Rate Comparison - Successful Runs

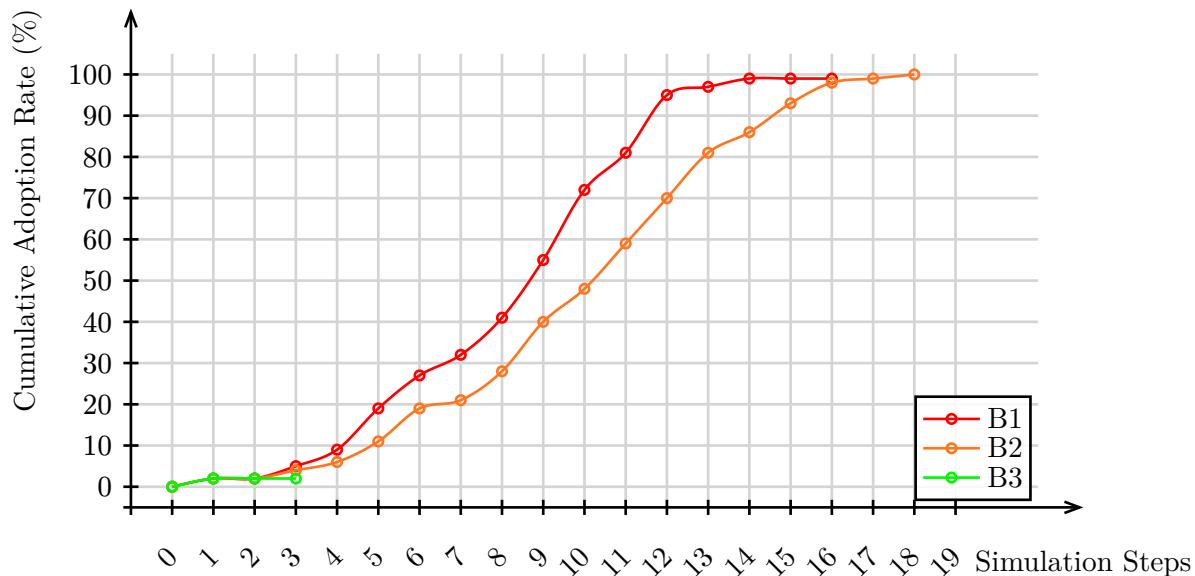


Figure 5: Cumulative Adoption Rate Comparison - Balanced Runs

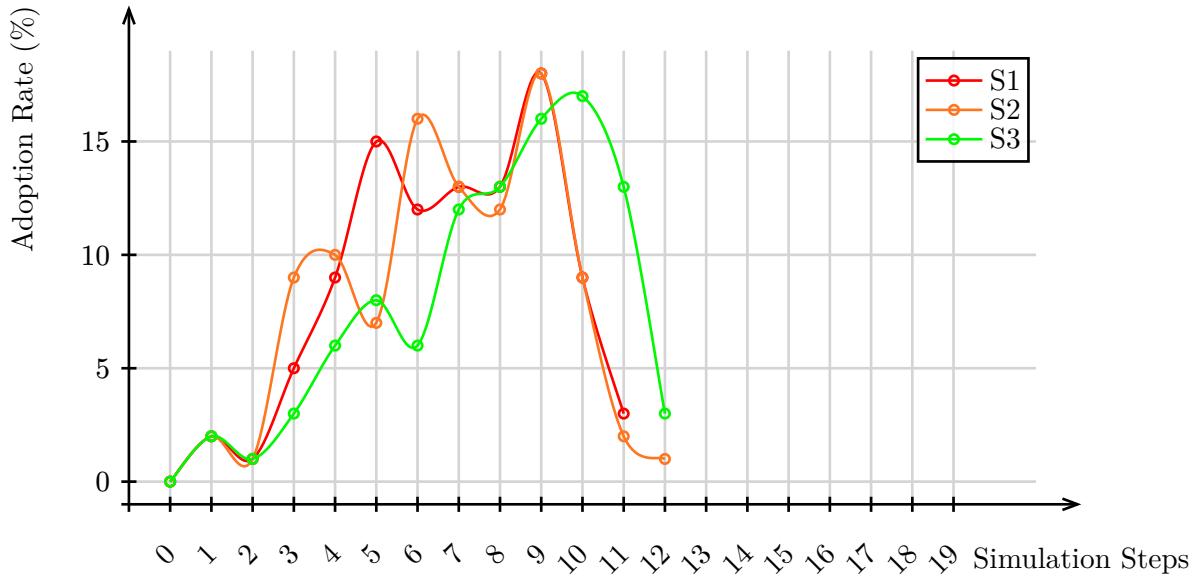


Figure 6: Adoption Rate Comparison - Successful Runs

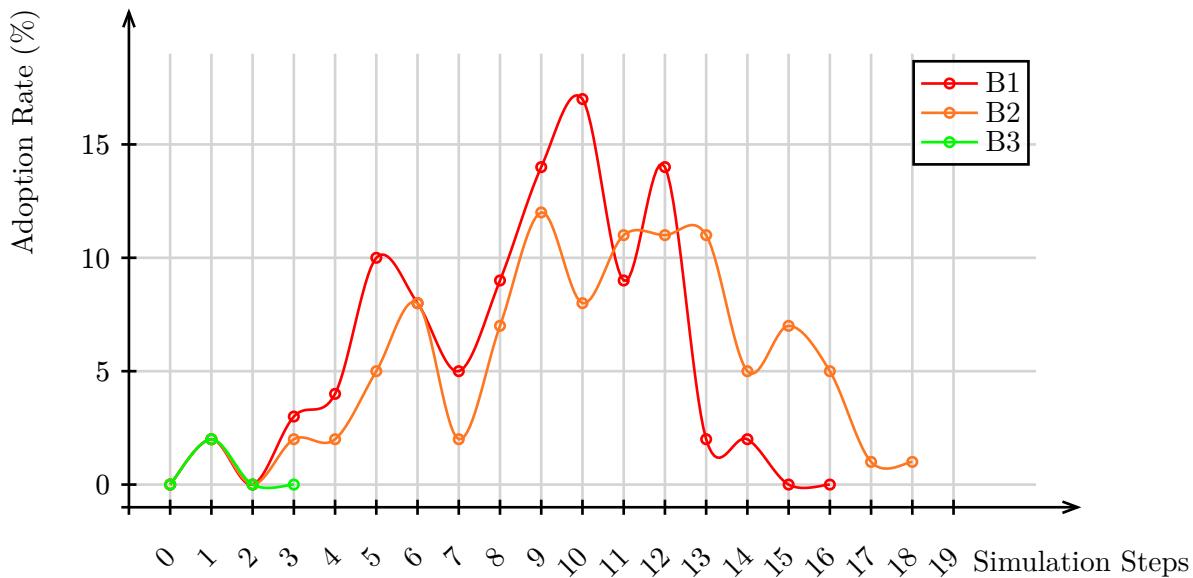


Figure 7: Adoption Rate Comparison - Balanced Runs

5.6.3. Successful Innovation Runs (S1-S3)

The three successful innovation simulations demonstrate high consistency in their overall diffusion patterns (Figure 4), though with some variation in acceleration phases:

- **Completion Time:** 11-12 steps across all runs (S1: 11 steps, S2: 12 steps, S3: 12 steps)
- **Final Adoption Rate:** 100% in all cases
- **Critical Mass Point:** Steps 4-5 for S1 and S2, step 6-7 for S3
- **Peak Adoption Rate:** S1 shows peaks around steps 5 and 9 (15-18%), S2 around steps 6 and 9, S3 more gradual with peak around step 10

The adoption rate analysis (Figure 6) reveals peak periods around steps 5, 6 and 9, where maximum single-step adoption occurs. This consistency suggests that when innovators and some early adopters adopt, it influences others to follow as social pressure and network effects come into play, though network position variations still play an important role. However, even successful innovations may find diffusion challenging, as demonstrated by S3.

The successful innovation adoption curves (Figure 4) appear to align with Rogers' classic S-curve model and bell shape (Section 3.2), exhibiting the characteristic three-phase progression (with S3 showing a slightly more gradual initiation phase):

Phase 1 - Slow Initial Growth (Steps 1-3): The curves show gradual initiation with 2-15% adoption, corresponding to Rogers' innovator and early adopter phases. This matches the theoretical expectation of limited initial uptake among risk-tolerant individuals.

Phase 2 - Rapid Acceleration (Steps 4-8): All three runs display steep acceleration phases reaching peak adoption rates of 15-18% per step, representing the critical mass breakthrough as early and late majority populations adopt. This phase closely aligns with Rogers' prediction of exponential growth once social proof mechanisms activate.

Phase 3 - Saturation and Plateau (Steps 9-12): The curves level off as remaining agents complete adoption, matching Rogers' saturation phase where diminishing returns occur as the innovation reaches market limits.

It's important to note that the fact that only the decision stage was considered may have influenced the observed diffusion patterns, as the initial stage could be slower due to the need for knowledge of the innovation's existence. Additionally, the missing media influence could have played a role in shaping the adoption dynamics.

5.6.4. Balanced Innovation Runs (B1-B3)

The balanced innovation scenarios exhibit significantly greater variability and extended timelines (Figure 5):

- **Completion Time:** 16 steps for B1 (early stopping), 18 steps for B2
- **Final Adoption Rate:** 99% for B1 (stopped at 16 due to no adoptions since step 14); 100% for B2; 2% for B3 (early stopping)
- **Critical Mass Point:** Steps 9-12 (requiring 40-50% adoption threshold)
- **Peak Adoption Rate:** 11-17% per step, occurring around steps 3 and 6-9 depending on the run

The early stopping of B3 at only 2% demonstrates that neutral innovations might face genuine diffusion risk. Unlike successful innovations that create intrinsic adoption motivation, balanced innovations rely heavily on network cascade effects, making them vulnerable to early adoption failures, particularly when considering only the decision stage and assuming everyone knows the innovation exists.

The Balanced Innovation runs exhibited greater variability and uncertainty, highlighting the challenges of achieving widespread adoption for innovations with neutral attributes.

The balanced innovation curves (Figure 5) reveal significant departures from Rogers' classic S-curve model (Section 3.2):

Flattened Initiation Phase (Steps 1-3): Extended slow-growth period with minimal adoption beyond innovators, suggesting that neutral innovation attributes create adoption barriers that delay the transition to rapid growth phases.

Multiple Acceleration Phases: Rather than a single steep acceleration, the curves show multiple smaller growth spurts (visible in B1 around steps 3 and 9, in B2 around steps 6-8), indicating that network effects must accumulate gradually when intrinsic innovation appeal is limited.

Extended Timeline with Incomplete Adoption Risk: The 16-18 step timelines represent 45-60% longer diffusion periods compared to successful innovations, with B3's complete failure demonstrating that not all innovations manage to achieve adoption.

Linear Progression Tendency: The balanced curves approximate linear growth more than exponential acceleration, suggesting that when innovation attributes are neutral, diffusion becomes primarily dependent on network cascade effects rather than intrinsic appeal (considering the simplifications made).

5.6.5. Category-Specific Adoption Patterns

Adoption Rate Over Steps by Category

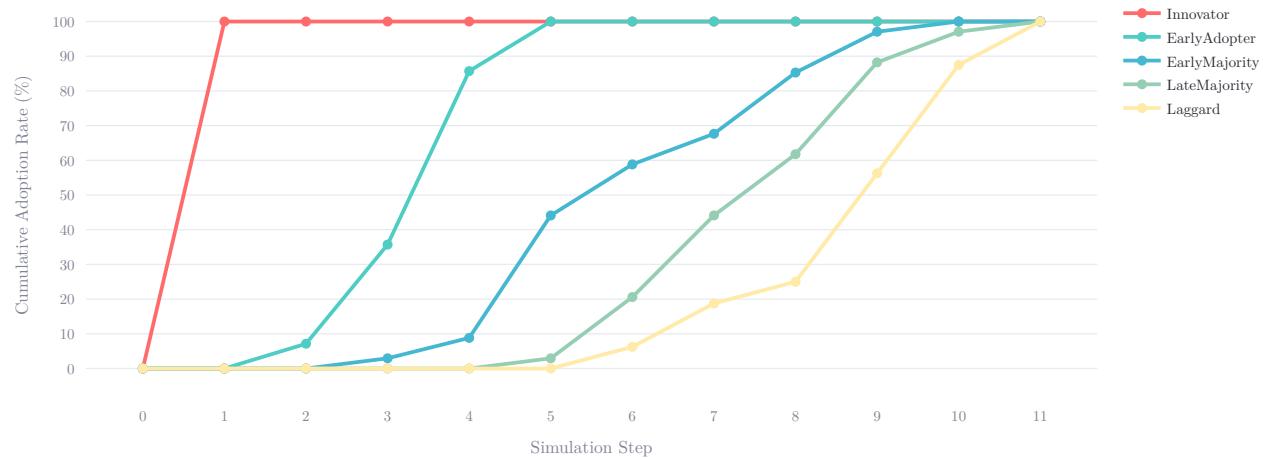


Figure 8: Categories Adoption Rate - Run S1

Adoption Rate Over Steps by Category

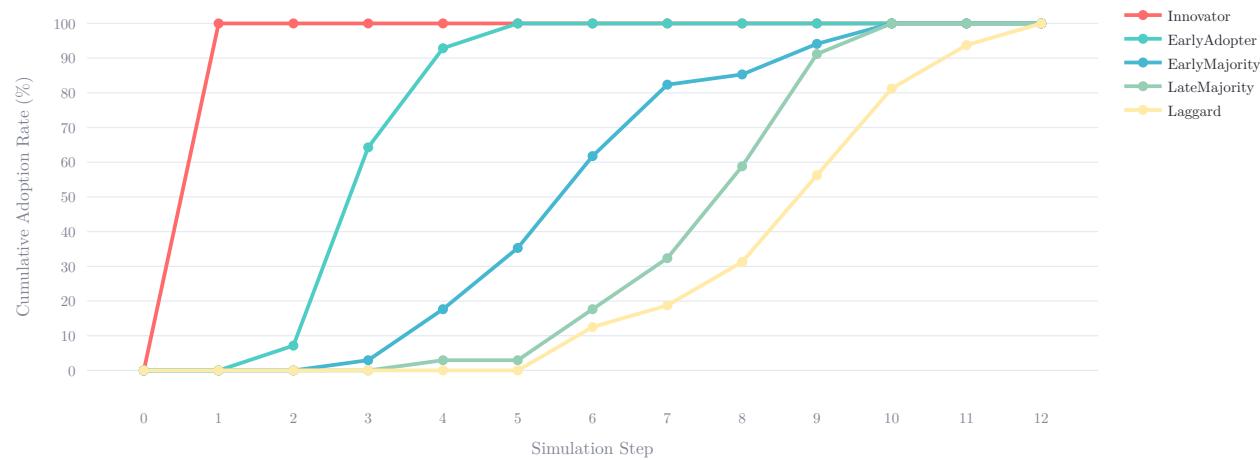


Figure 9: Categories Adoption Rate - Run S2

Adoption Rate Over Steps by Category

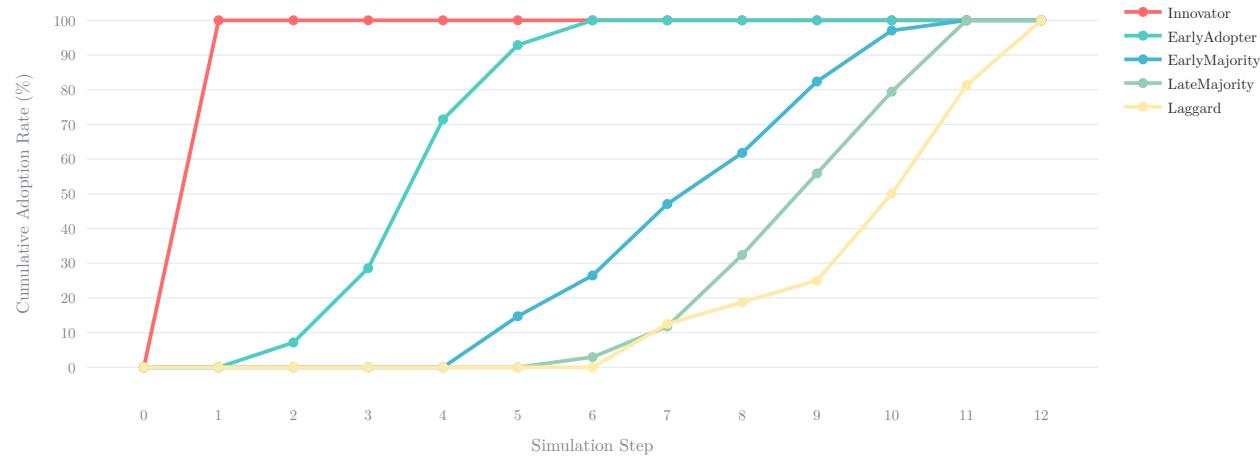


Figure 10: Categories Adoption Rate - Run S3

5.6.5.1. Category-Specific Adoption Patterns - Successful Innovations

The category progression analysis (Figure 8, Figure 9, Figure 10) aligns with Rogers' theoretical sequence while revealing important behavioral nuances:

Innovators: Consistent first adoption within steps 1-2 across all runs, establishing multiple diffusion seed points simultaneously.

Early Adopters: Achieve 100% adoption by steps 5-6 across runs, demonstrating their role as opinion leaders in this simulation context. Their rapid adoption appears to contribute to triggering critical mass.

Early Majority: Initially show resistance even when few others have adopted, but their resistance diminishes due to network position and their profile characteristics.

Late Majority: Similar to Early Majority, they adopt later, but their adoption is more influenced by network pressure and the actions of the Early Majority.

Laggards: Latest to adopt due to their strong aversion to change and reliance on established norms. In this case, network position played a significant role in their adoption timing, while the innovation's attributes acted as a secondary influence.

Adoption Rate Over Steps by Category

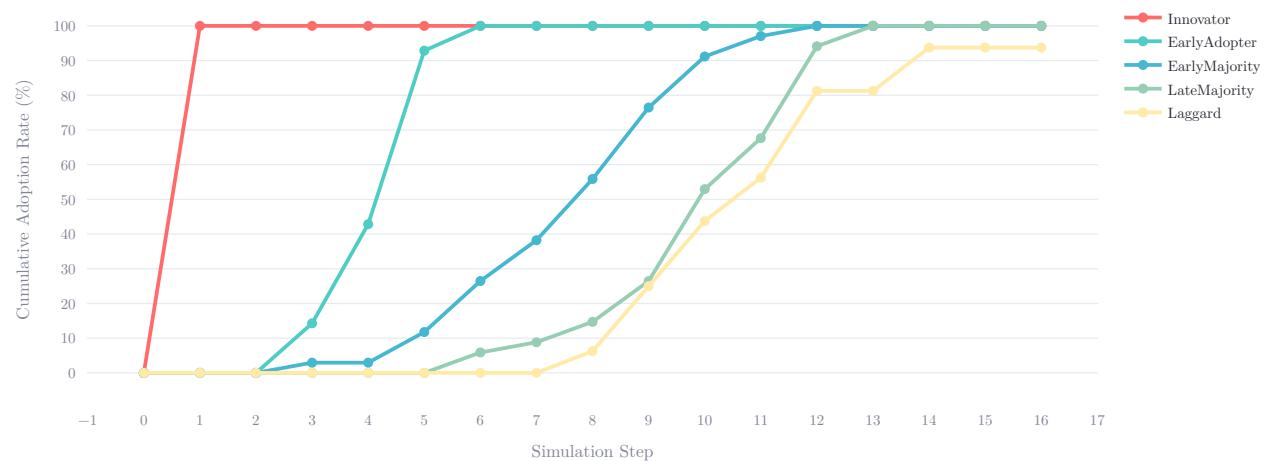


Figure 11: Categories Adoption Rate - Run B1

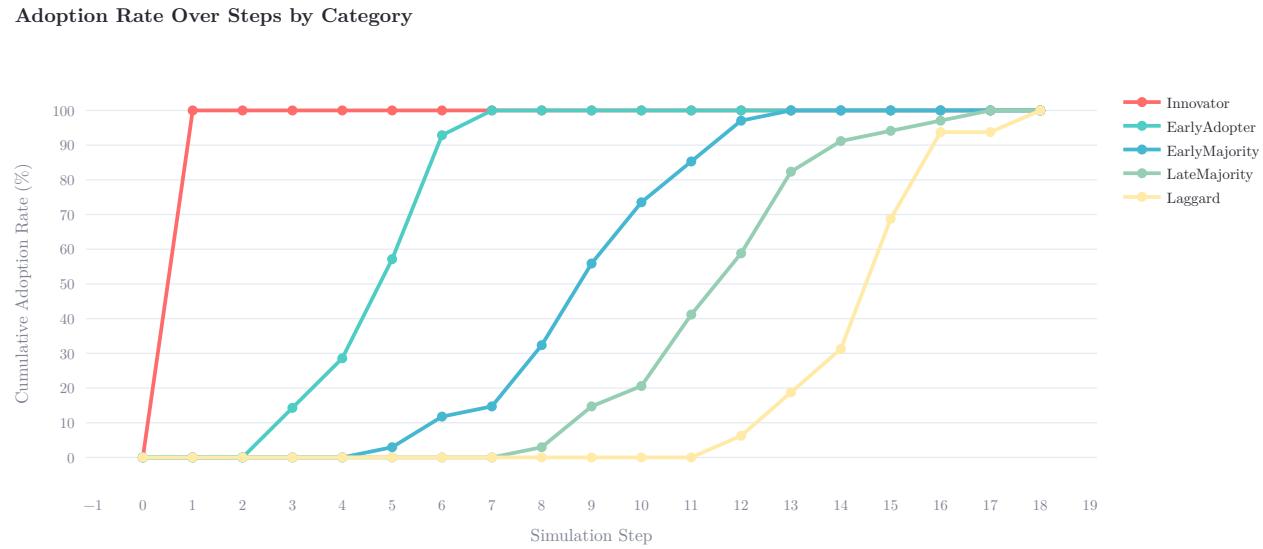


Figure 12: Categories Adoption Rate - Run B2

5.6.5.2. Category-Specific Adoption Patterns - Balanced Innovations

The balanced innovation category analysis (Figure 11, Figure 12) reveals markedly different dynamics:

Extended Evaluation Periods: All categories except innovators require additional evaluation time, with early adopters not completing adoption until steps 6-8 compared to steps 4-5 in successful scenarios.

Heightened Threshold Sensitivity: Early and late majority groups demonstrate much stronger dependence on network adoption signals before committing.

Category Override Phenomena: More frequent instances where network position overrides categorical tendencies, particularly laggards adopting before late majority members.

Persistent Resistance Risk: Evidence of permanent non-adoption even under strong network pressure, as demonstrated in B1 where one laggard never adopts despite 99% global adoption.

5.6.6. Agents Response Analysis

Since it's not possible to show all agent responses, selected examples from key decision points in the simulations are provided, with a general summary of their behavior.

5.6.6.1. Successful Innovation Responses

Agent responses in successful innovation scenarios consistently acknowledge positive innovation attributes even when deciding not to adopt, while using network signals as confirmatory evidence rather than primary drivers. The following examples illustrate this pattern:

Early Adopter Response (Step 4 - S1):

- **Decision:** ADOPT

- **Reasoning:** “With the relatively high relative advantage score (7) and the increasing global adoption rate, I feel more confident in adopting this innovation. The trialability is still high (8), which allows me to test it without major commitment risks.”
- **Thinking:** “The global adoption rate is now at 8%, which suggests that it’s gaining traction and showing promise. Although my network hasn’t adopted it yet, I’m starting to see more validation of its value proposition.”
- **Confidence Level:** 8

Early Majority Response (Step 4 - S1):

- **Decision:** NOT_ADOPT
- **Reasoning:** “While the relative advantage (7) and trialability (8) are still promising, I’m concerned about the complexity (4) and observability (5). I need to see more evidence that it will integrate smoothly into my existing routines before committing.”
- **Thinking:** “The global adoption rate has increased significantly, but it’s still not at the level where I feel comfortable adopting without more information. Although my network hasn’t adopted it yet, the growing number of users suggests there may be some benefits.”
- **Confidence Level:** 6

Late Majority Response (Step 4 - S1):

- **Decision:** NOT_ADOPT
- **Reasoning:** “While the relative advantage (7/10) is more compelling with higher global adoption, I’m still concerned about compatibility issues (6/10) and complexity (4/10). Without any adopters in my network to provide firsthand experience and guidance, I’m not yet ready to take on this innovation.”
- **Thinking:** “The global adoption rate has increased significantly, but it’s still not at a level that makes me feel entirely confident in its success and stability. I’m starting to consider the potential benefits of adopting this innovation.”
- **Confidence Level:** 5

Key Pattern: Even when not adopting, agents consistently frame innovation attributes in positive terms (e.g., “promising,” “compelling,” “attractive features”) and use network adoption patterns as validation of the innovation’s inherent value rather than as the primary decision driver.

5.6.6.2. Balanced Innovation Responses

In balanced innovation scenarios, network influence becomes the crucial factor driving adoption decisions, with agents expressing uncertainty about innovation value and relying heavily on social proof mechanisms:

Early Adopter Response (Step 4 - B1):

- **Decision:** ADOPT
- **Reasoning:** “With a growing number of adopters globally, I believe there’s increasing evidence that this innovation is worth considering. While I still can’t rely on my network for guidance, the trend suggests that others are starting to see value in it, which increases my confidence.”

- **Thinking:** “The global adoption rate has increased from 2% to 5%, which suggests that more people are starting to see value in the innovation. Although no one in my network has adopted it yet, I’m beginning to consider its potential benefits.”
- **Confidence Level:** 7

Early Adopter Response (Step 3 - B1):

- **Decision:** NOT_ADOPT
- **Reasoning:** “The lack of growth in global adoption raises concerns about the innovation’s value and sustainability. Given that no one in my network has adopted, I’m hesitant to take on the risk of being an early adopter.”
- **Thinking:** “The global adoption rate has stayed the same, which might indicate a plateau or some underlying issues. My network’s hesitation is still present, but I’m starting to question whether I should be an early adopter if others in my social circle aren’t”
- **Confidence Level:** 6

Early Majority Response (Step 3 - B1):

- **Decision:** NOT_ADOPT
- **Reasoning:** “The absence of peer experiences and success stories makes it difficult for me to trust this innovation. I need to see more widespread adoption, even if it’s just from people outside my immediate network, before considering its value.”
- **Thinking:** “I’m still not seeing any adoption among my network, and the global adoption rate is extremely low at only 2%. This lack of social proof is a significant concern for me.”
- **Confidence Level:** 6

Key Pattern: Network adoption patterns become the primary consideration, with agents expressing uncertainty about innovation benefits and making decisions primarily based on social validation rather than intrinsic innovation appeal. The absence of network adoption signals creates significant adoption barriers.

5.6.7. Comparison

The simulations reveal substantial differences between scenarios (considering only the decision stage):

- **Diffusion Speed:**
 - **Successful:** Completed in 11-12 steps, with a sharp acceleration phase.
 - **Balanced:** 16-18 steps in successful runs, with more linear progression, a risk of early stalling (e.g., B3) and a less predictable trajectory.
- **Role of the Network:**
 - **Successful:** Intrinsic attributes drive adoption even without strong network pressure; the network acts as an amplifier.
 - **Balanced:** Network influence is the primary driver; adoption is heavily dependent on position and connections.
- **Category Behavior:**
 - **Successful:** Theoretical patterns mostly preserved, with rare overrides due to advantageous positioning.

- **Balanced:** Greater variability, more frequent category overrides (e.g., well-connected Laggards adopting before some Late Majority agents).

In summary, successful innovations demonstrate more predictable diffusion patterns that align closely with Rogers' theoretical framework, while balanced innovations reveal the complex interplay between network structure and individual decision-making.

5.6.8. Network Positioning Effects

The small-world structure created several notable dynamics that influenced adoption patterns:

- **Bridge Agents:** Individuals connecting otherwise separate clusters wield disproportionate influence on global diffusion patterns
- **Hub Effects:** While not scale-free, agents with above-average connectivity become influential regardless of their adopter category as other agents perceived their adoption as successful
- **Peripheral Resistance:** Agents with fewer connections show greater resistance to adoption, requiring stronger local influence

These effects will be highlighted in the detailed case studies below.

5.7. Detailed Single-Run Analysis

The following sections present detailed examination of representative runs to understand the micro-dynamics driving the observed macro-patterns.

5.7.1. Successful Innovation Case Study (Run S1)

Innovation Adoption Progress Over Steps

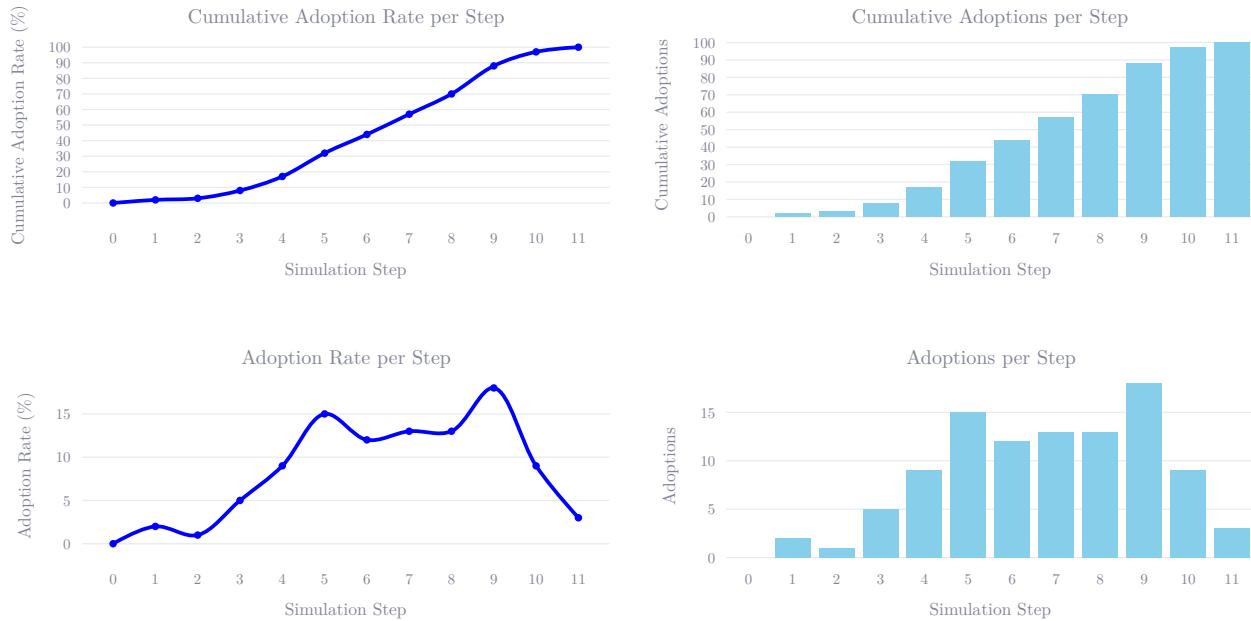


Figure 13: Successful Innovation - Representative Adoption Curve (Run S1)

Social Network: Innovation Diffusion Visualization (Step 2)

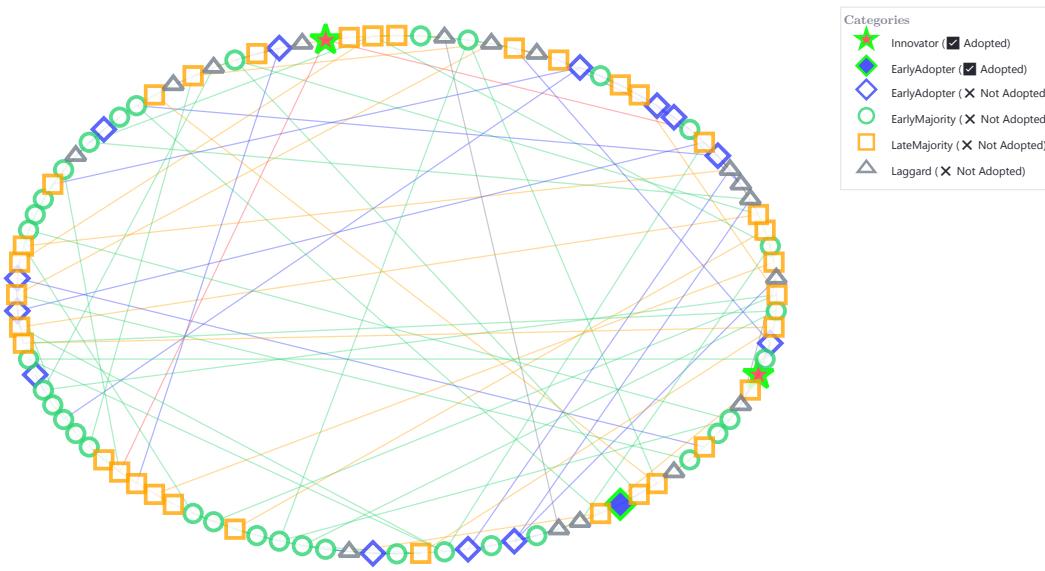


Figure 14: Successful Simulation - Network State (Step 2)

Social Network: Innovation Diffusion Visualization (Step 4)

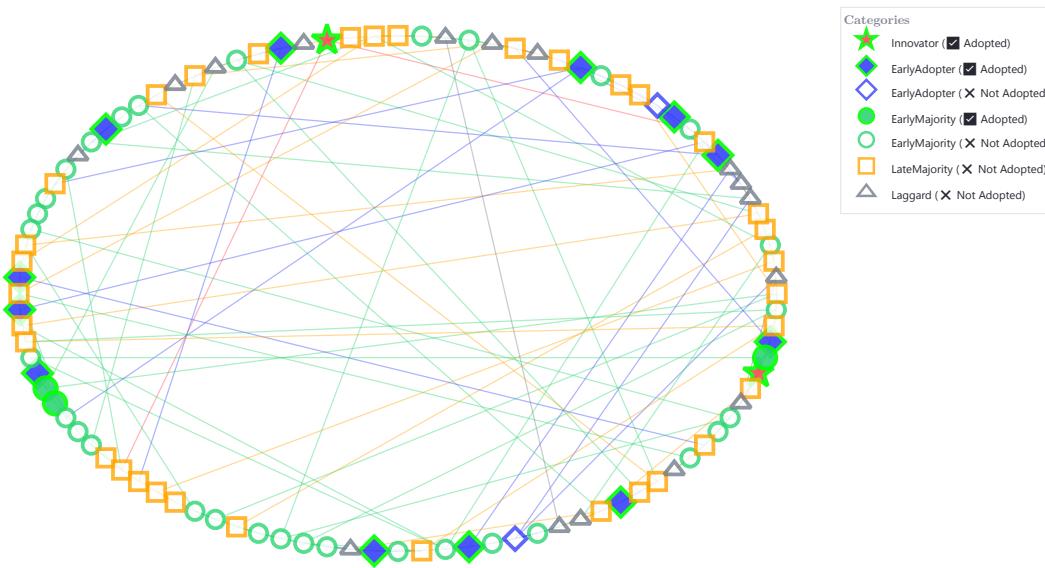


Figure 15: Successful Simulation - Network State (Step 4)

Social Network: Innovation Diffusion Visualization (Step 6)

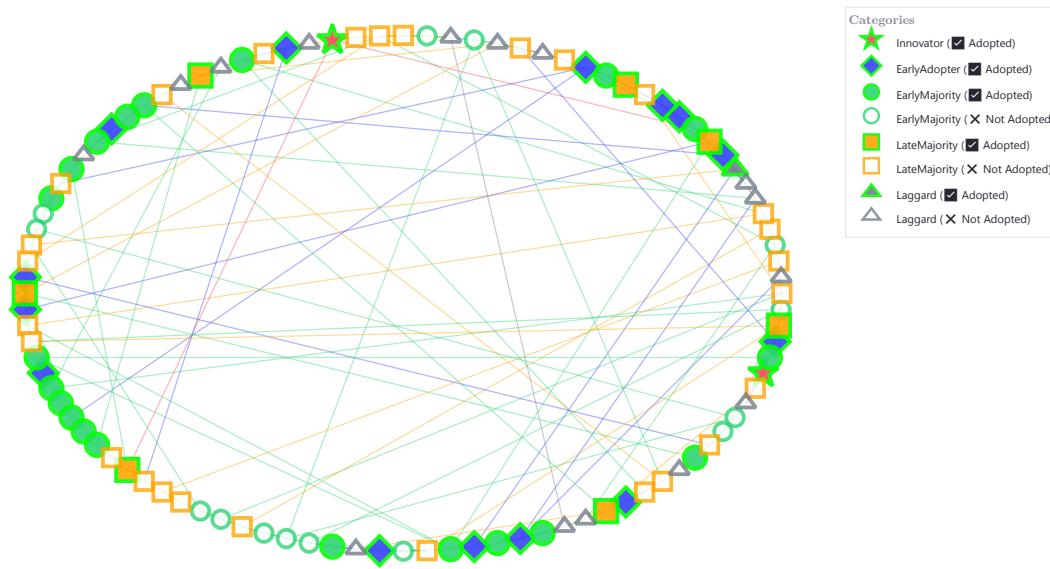


Figure 16: Successful Simulation - Network State (Step 6)

Social Network: Innovation Diffusion Visualization (Step 8)

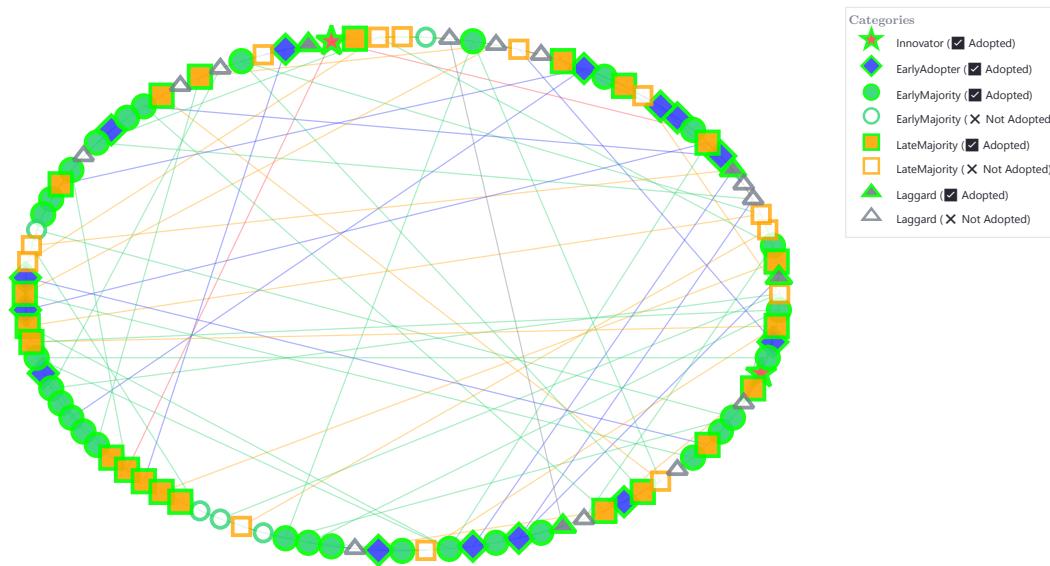


Figure 17: Successful Simulation - Network State (Step 8)

Social Network: Innovation Diffusion Visualization (Step 10)

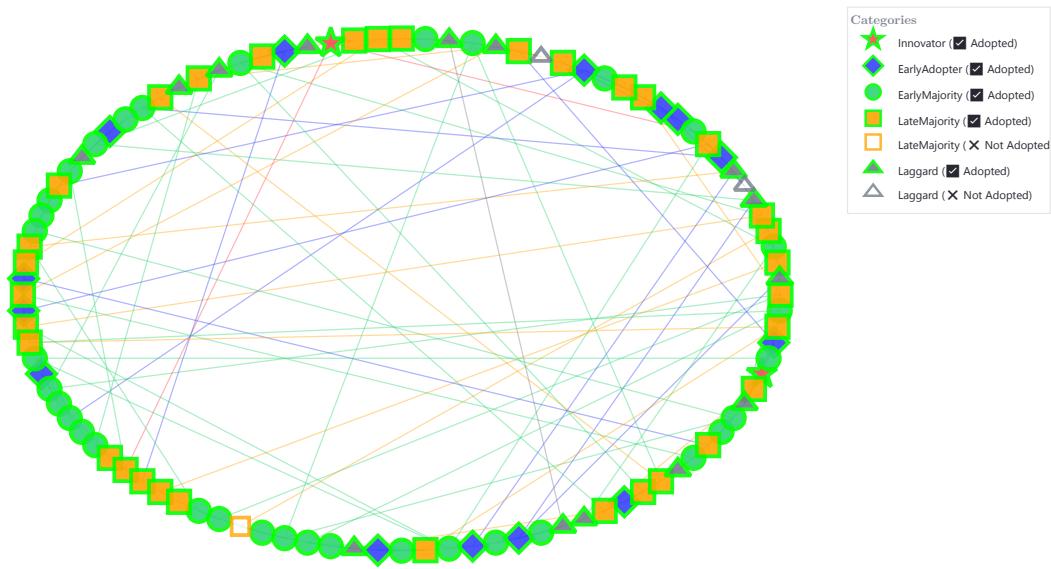


Figure 18: Successful Simulation - Network State (Step 10)

Social Network: Innovation Diffusion Visualization (Final Step)

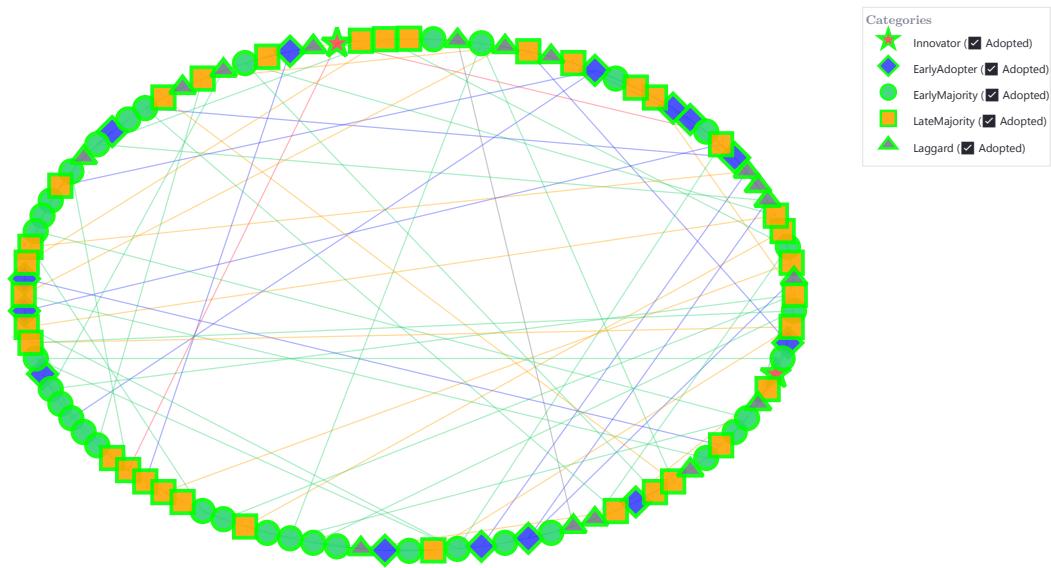


Figure 19: Successful Simulation - Network State (Step 11)

5.7.1.1. Network Evolution Analysis

Steps 1-2 - Distributed Initiation: Both innovators adopt simultaneously in separate network regions (Figure 14), creating distributed innovation sources. This multi-seed pattern accelerates diffusion compared to single-origin models.

Steps 3-4 - Bridge Activation: Early adopters in strategic bridge positions begin connecting adoption islands (Figure 15). Notably, some early majority agents adopt despite minimal local network exposure, indicating strong innovation appeal overcoming typical social proof requirements.

Steps 5-6 - Critical Mass Formation: Rapid expansion creates distinct adoption clusters (Figure 16). Late majority agents begin adopting around step 5-6, with the critical mass phase evident as multiple categories adopt simultaneously.

Steps 7-8 - Cascade Acceleration: Network pressure intensifies with even some laggards adopting before late majority counterparts (Figure 17), illustrating how network position can override categorical tendencies under favorable innovation conditions.

Steps 9-11 - Complete Saturation: Final phases show systematic elimination of remaining resistance (Figure 18, Figure 19), with the small-world topology ensuring no agent remains permanently isolated from adoption influences.

5.7.1.2. General Observations

The S1 simulation (Figure 13) shows favorable diffusion conditions with its compressed 11-step timeline and smooth acceleration pattern. The cumulative adoption curve demonstrates classic S-curve characteristics: gradual initiation (steps 1-2), rapid acceleration (steps 3-7) and final saturation (steps 8-11).

The successful innovation simulation demonstrates several key patterns:

- **Rapid Cascade Dynamics:** The favorable innovation attributes created a “pull effect” where agents were intrinsically motivated to adopt, leading to faster diffusion across all categories. The innovation achieved complete adoption by Step 11, with the network structure serving as an amplifier rather than a primary driver.
- **Category Behavior Validation:** While agents generally followed their prescribed adoption tendencies, network positioning occasionally overrode categorical preferences. Notably, some Laggards adopted before Late Majority agents when positioned in high-influence network locations, demonstrating the power of local social pressure.
- **Network Structure Impact:** The small-world topology facilitated rapid diffusion through both local clustering (enabling rapid local spread) and long-range connections (allowing adoption to “jump” across network regions). Bridge agents connecting different clusters showed disproportionate influence on overall adoption patterns.

5.7.1.3. Specific Behavioral Observations

Observation 1: Early Majority Premature Adoption

Social Network: Innovation Diffusion Visualization (Step 2)

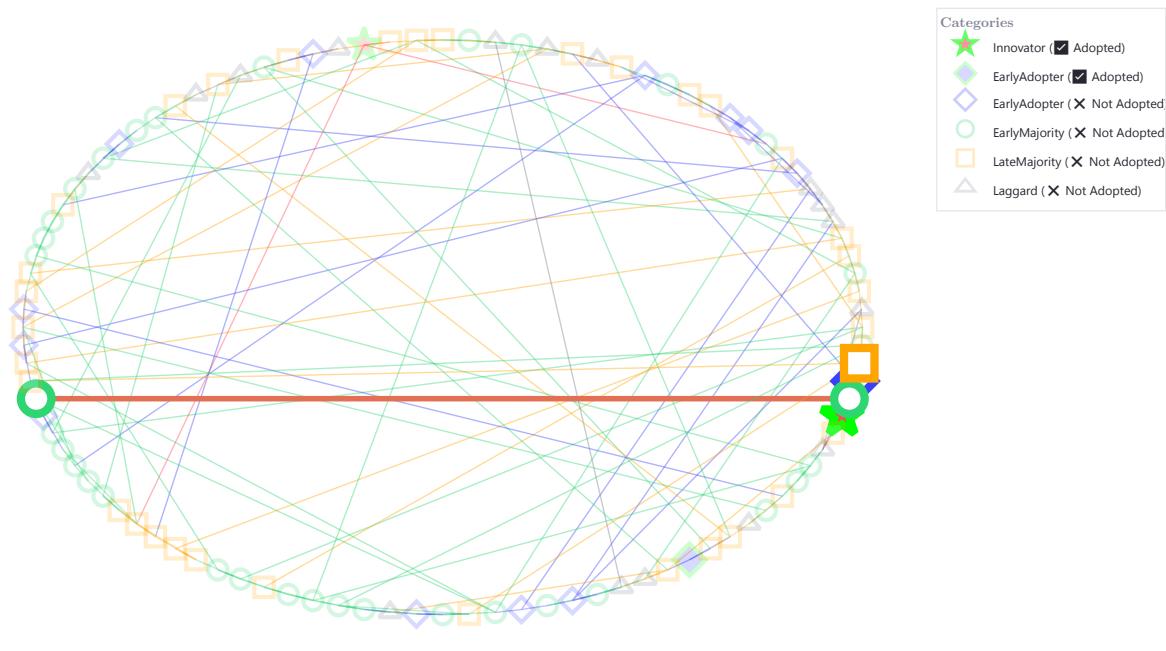


Figure 20: Early Majority Network State at Step 2

Observation: An Early Majority agent adopted in Step 3 despite minimal network exposure (connected to only 1 Innovator).

Agent Response on Step 3:

- Reasoning:** 'I still have concerns about the compatibility (6) and complexity (4) of this innovation. However, the relative advantage (7) and trialability (8) are attractive features that could provide a competitive edge if I were to adopt. I'll wait for a bit more traction from my network or other early adopters before making a decision.'
- Thinking:** 'The global adoption rate is slowly increasing, and it's now over 2%. While my network remains unaffected, I'm starting to see some early signs of potential value in this innovation.'
- Confidence Level:** 7

Analysis: This response highlights an important limitation in LLM-based agents: the disconnect between expressed reasoning and final decision. The agent's reasoning suggests caution and waiting, yet ultimately adopted. This inconsistency suggests possible LLM bias toward adoption or insufficient constraint between reasoning and decision-making processes. While this doesn't necessarily mean the decision to adopt was incorrect, it does indicate a need for improved alignment between cognitive processes and final choices.

Observation 2: Late Majority Social Influence Sensitivity

Social Network: Innovation Diffusion Visualization (Step 4)

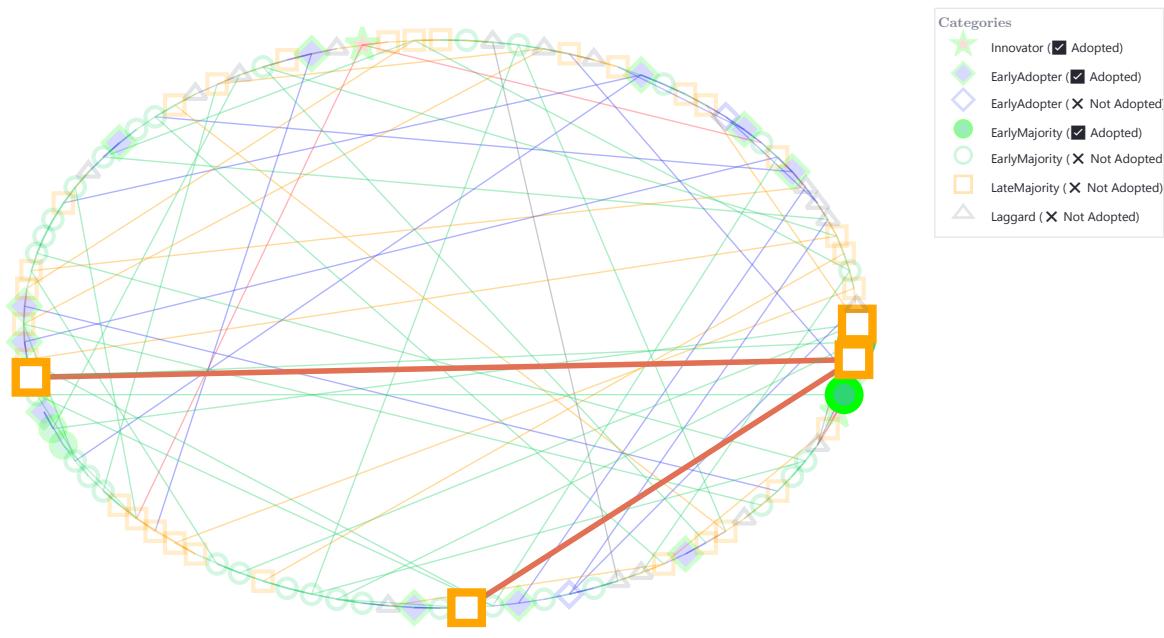


Figure 21: Late Majority Network State at Step 4

Observation: A Late Majority agent adopted in Step 5 after observing multiple connections adopting, despite initial hesitancy.

Agent Response on Step 5:

- Reasoning:** 'While the innovation has some drawbacks (low compatibility and observability), its advantages seem substantial enough to outweigh them. The fact that many others have adopted it without major issues is reassuring, and I can now see some potential benefits for my organization.'
- Thinking:** 'The global adoption rate is now significant, and having one of my connections adopt it successfully gives me more confidence. However, I still need to be cautious about potential complications or problems.'
- Confidence Level:** 7

Analysis: This response demonstrates appropriate Late Majority behavior, balancing innovation attributes with social proof from network connections and global adoption rates.

Observation 3: Laggard Network Pressure Response

Social Network: Innovation Diffusion Visualization (Step 6)

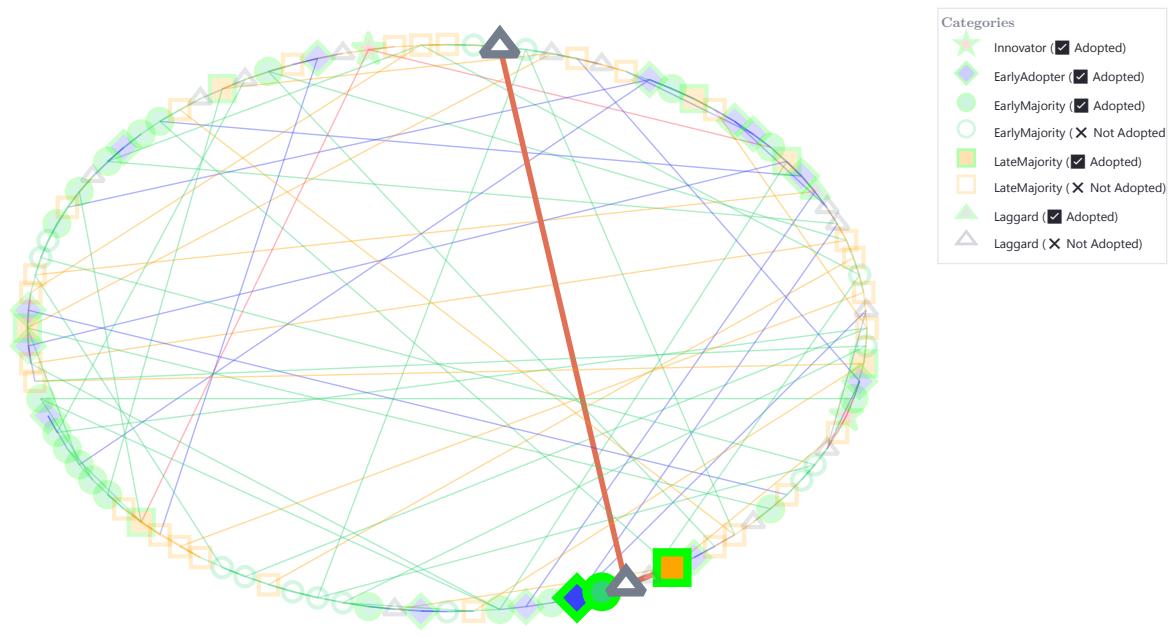


Figure 22: Laggard Network State at Step 6

Observation: A Laggard agent adopted in Step 7 after observing significant network adoption, despite initial resistance.

Agent Response on Step 7:

- **Reasoning:** ‘While I still have some reservations about the innovation’s complexity, its trialability and observability make it easier to try out without too much risk. And with more than a third of the global community already using it, I think it’s time for me to give it a go.’
- **Thinking:** ‘The global adoption rate is really taking off now, and it’s hard not to feel like I’m missing out on something if everyone else is adopting it. Two of my close connections are already using it, so that’s giving me some confidence.’
- **Confidence Level:** 9

Analysis: This demonstrates how network positioning can overcome inherent category resistance. The Laggard’s high confidence level and FOMO (fear of missing out) reasoning reflects realistic social pressure dynamics.

5.7.2. Balanced Innovation Case Study (Run B1)

Innovation Adoption Progress Over Steps

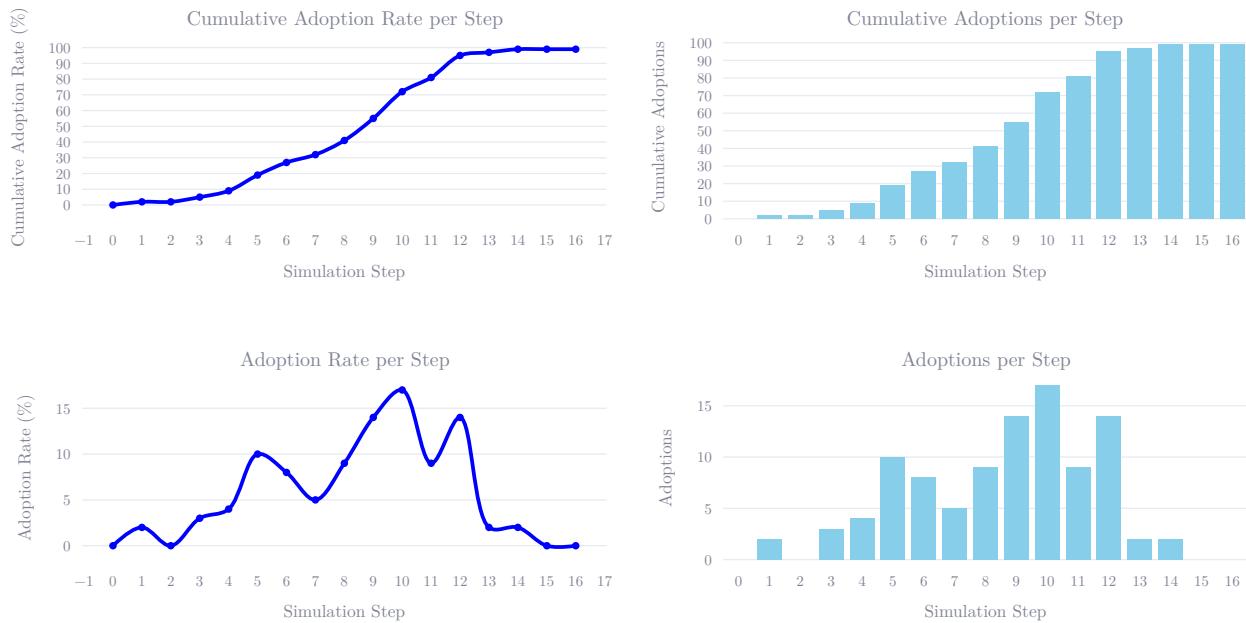


Figure 23: Balanced Innovation - Adoption Progress Analysis

Social Network: Innovation Diffusion Visualization (Step 2)

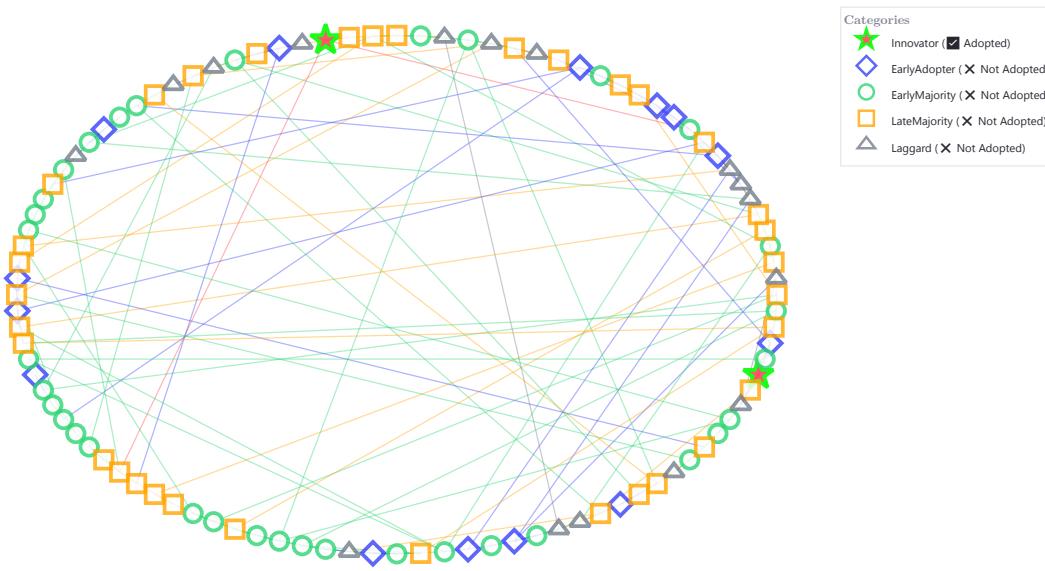


Figure 24: Balanced Simulation - Network State (Step 2)

Social Network: Innovation Diffusion Visualization (Step 3)

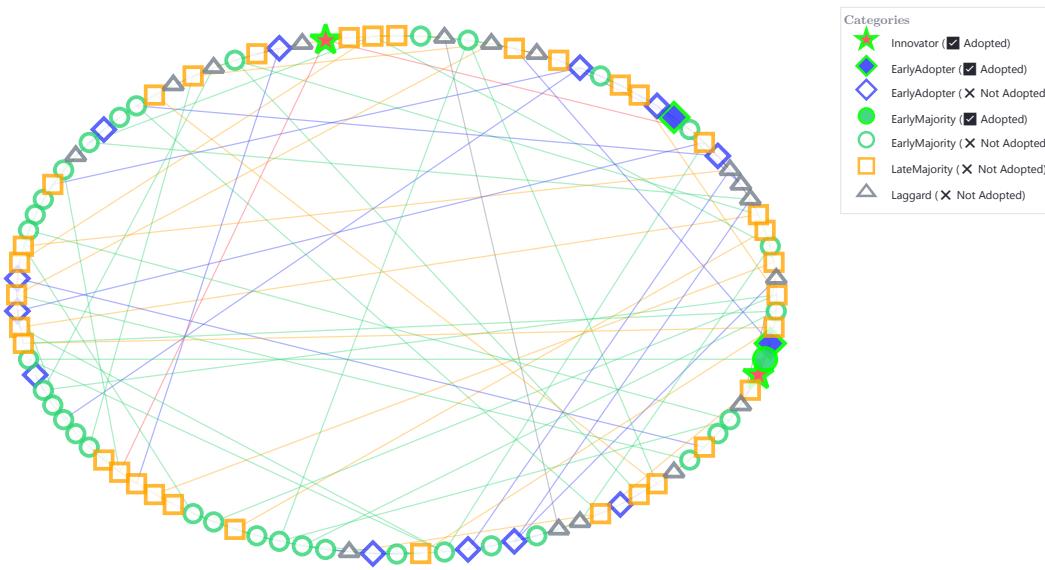


Figure 25: Balanced Simulation - Network State (Step 3)

Social Network: Innovation Diffusion Visualization (Step 6)

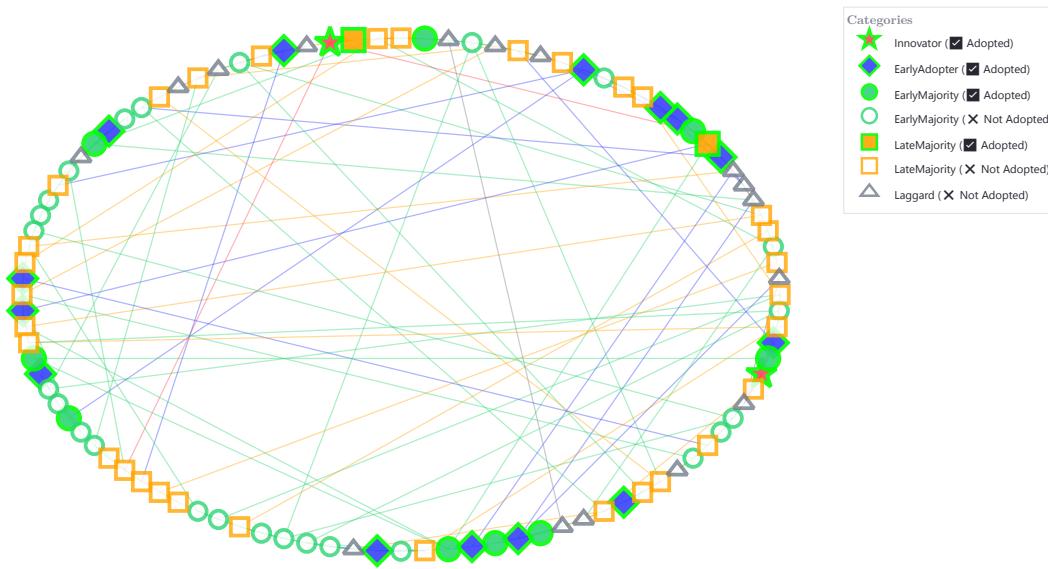


Figure 26: Balanced Simulation - Network State (Step 6)

Social Network: Innovation Diffusion Visualization (Step 8)

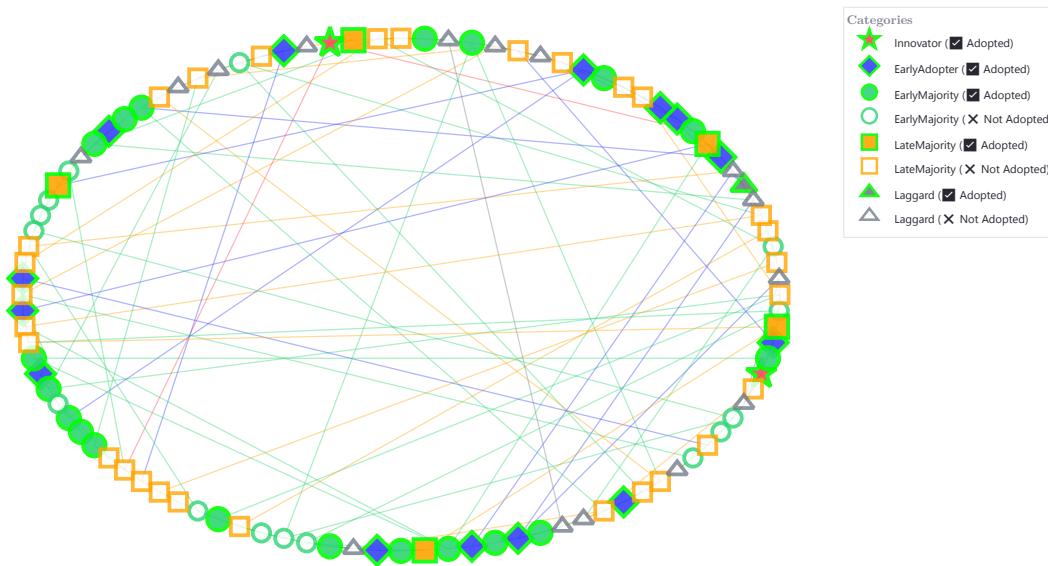


Figure 27: Balanced Simulation - Network State (Step 8)

Social Network: Innovation Diffusion Visualization (Step 9)

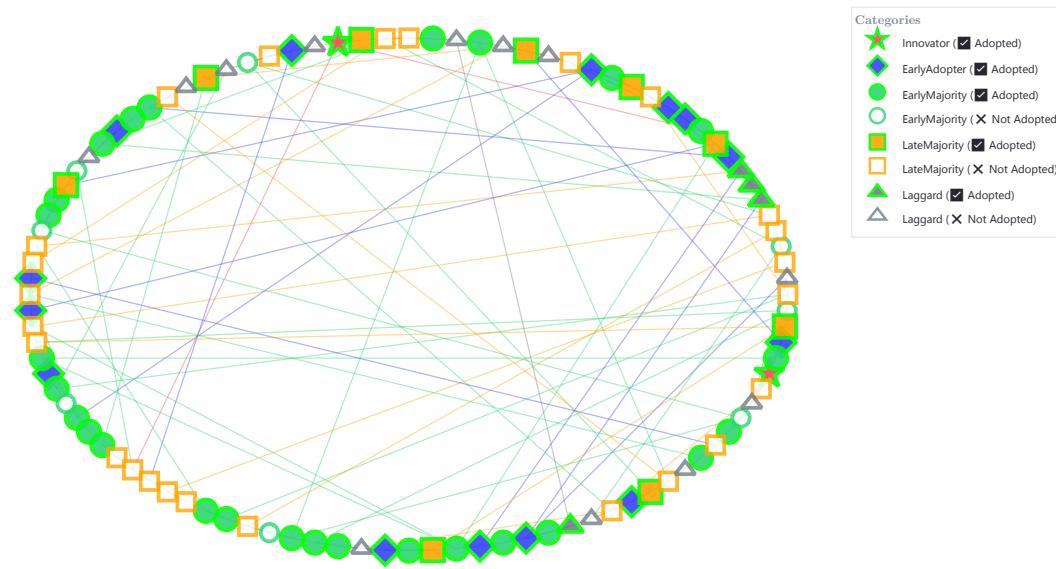


Figure 28: Balanced Simulation - Network State (Step 9)

Social Network: Innovation Diffusion Visualization (Step 10)

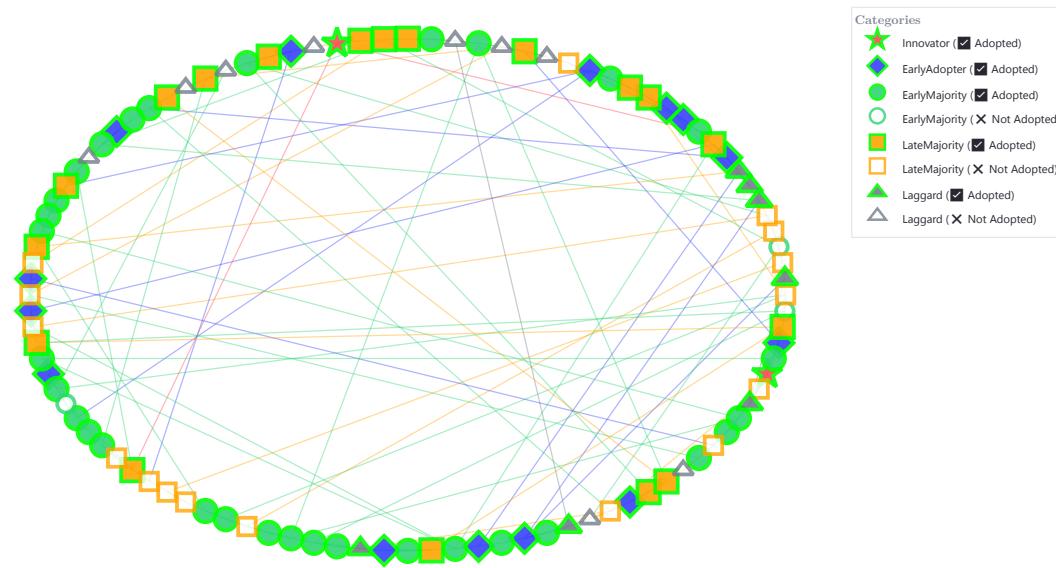


Figure 29: Balanced Simulation - Network State (Step 10)

Social Network: Innovation Diffusion Visualization (Step 12)

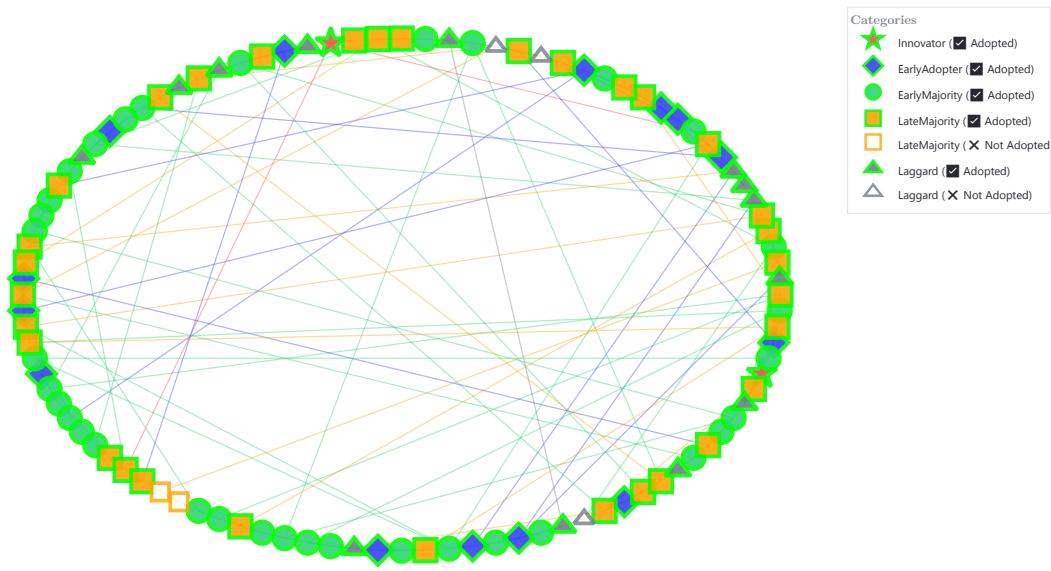


Figure 30: Balanced Simulation - Network State (Step 12)

Social Network: Innovation Diffusion Visualization (Step 14)

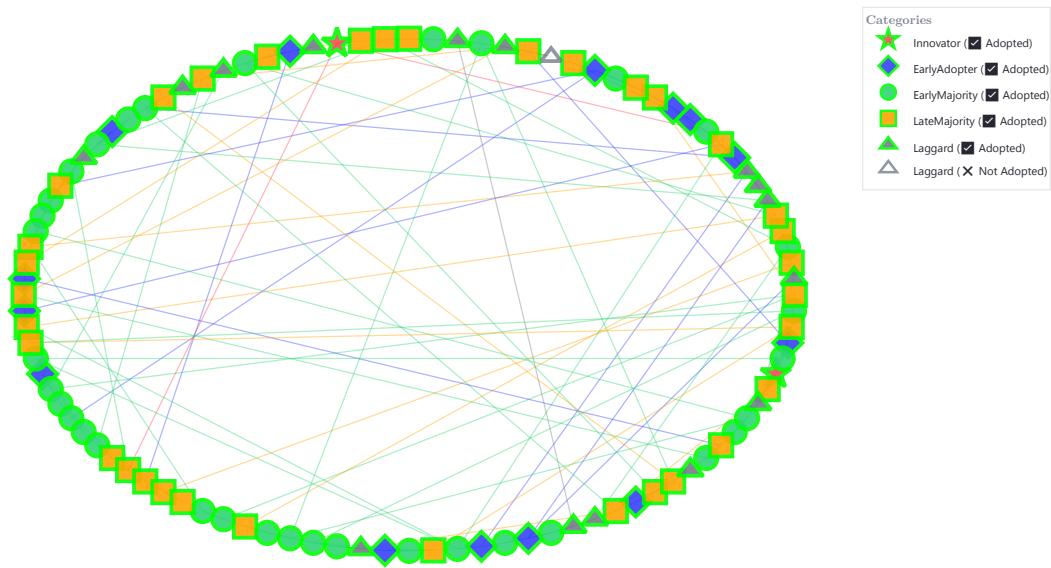


Figure 31: Balanced Simulation - Network State (Step 14)

5.7.2.1. Network Evolution Analysis

Steps 1-2 - Conservative Initiation: Single innovator adoption in step 1 with no additional adoptions in step 2 (Figure 24), contrasting sharply with successful scenario multi-seed patterns.

Step 3 - Strategic Breakthrough: Simultaneous adoption by strategically positioned early adopter and early majority agents (Figure 25) represents critical transition from innovation appeal to network-driven adoption.

Steps 4-6 - Gradual Expansion: Slow cluster growth with clear spatial boundaries between adopted and non-adopted network regions (Figure 26). Adoption concentrates in high-connectivity areas while peripheral regions remain resistant.

Steps 7-8 - Network Pressure Emergence: Late majority activation begins as local network pressure reaches threshold levels (Figure 27). Some laggards adopt due to favorable positioning, demonstrating network override effects.

Steps 9-12 - Network Pressure Peak: Sustained adoption activity with late majority completing adoption and most laggards adopting by step 12 (Figure 30). The network effects become clearly visible as social pressure drives remaining adoptions.

Steps 13-16 - Early Stopping Trigger: By step 14, adoption reaches 99% with one persistent holdout (Figure 31). No additional adoptions occur in steps 15-16, triggering the early stopping mechanism. This illustrates that neutral innovations may face genuine adoption ceilings where even sustained network pressure cannot overcome individual resistance.

5.7.2.2. General Observations

The B1 simulation (Figure 23) represents network-dependent diffusion dynamics, reaching 99% adoption by step 14 before triggering early stopping at step 16 due to no new adoptions. The adoption pattern shows more linear progression with multiple smaller acceleration phases rather than a single dramatic surge.

The balanced innovation simulation revealed fundamentally different dynamics:

- **Network-Dependent Diffusion:** With neutral innovation attributes, network influence became the primary adoption driver. This created a more linear, gradual adoption pattern requiring more steps compared to the successful scenario.
- **Enhanced Category Sensitivity:** Agent categories displayed heightened adherence to their risk profiles. Early Adopters became more selective, requiring stronger network signals before adopting, while Majority groups showed clear threshold behaviors, requiring more local network adoption before committing.
- **Incomplete Diffusion Risk:** Unlike the successful innovation that achieved universal adoption, the balanced innovation reached 99% adoption with one Laggard holdout by Step 14. This suggests that neutral innovations face inherent adoption ceilings in realistic scenarios.

- **Temporal Dependency:** The extended 14-step timeline revealed how network effects accumulate over time. Initial hesitancy (steps 1-2 with minimal adoption) gave way to sudden adoption clusters (step 3), demonstrating non-linear network influence patterns.
- **Spatial Clustering Patterns:** Adoption concentrated in network regions with high connectivity, creating distinct adopter and non-adopter clusters. This spatial heterogeneity persisted throughout the simulation, indicating that network topology significantly shapes diffusion patterns for marginal innovations.

5.7.2.3. Specific Behavioral Observations

Observation 1: Delayed Early Adoption

Observation: In Step 1, only Innovators adopted; in Step 2, there were no adoptions; and in Step 3, only a few Early Adopters decided to adopt.

Analysis: This temporal pattern highlights the non-linear effects of network influence. After an initial period of inertia, the accumulation of previous adoption decisions can suddenly trigger uptake among previously hesitant categories. Such delayed responses underscore how network structure and local thresholds can create tipping points, leading to abrupt shifts in adoption dynamics.

Of course due to the nature of LLM this is not always the case and it might be just a coincidence, but it is an interesting observation nonetheless.

Observation 2: Laggard Social Pressure

Social Network: Innovation Diffusion Visualization (Step 7)

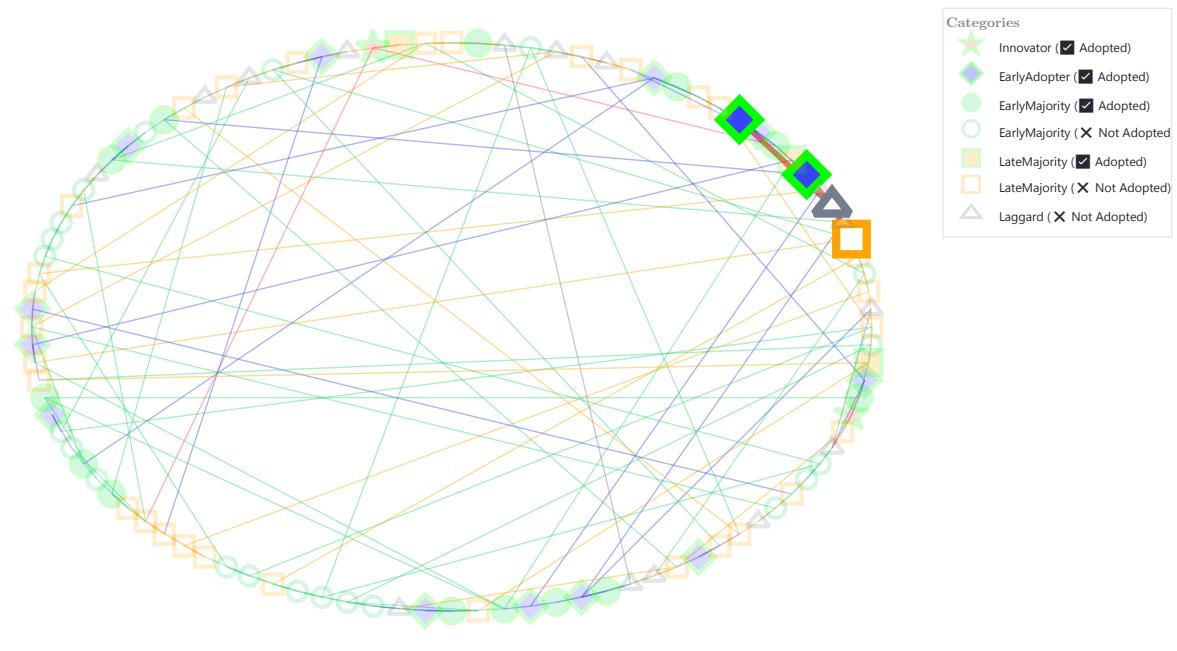


Figure 32: Laggard Network State in Step 7

Observation: Laggards exhibited a delayed response to social pressure, ultimately adopting after observing significant uptake among their peers.

Agent Response on Step 8:

- **Reasoning:** ‘While I still have some reservations about the innovation’s complexity, its trialability and observability make it easier to try out without too much risk. And with more than a third of the global community already using it, I think it’s time for me to give it a go.’
- **Thinking:** ‘The global adoption rate is really taking off now, and it’s hard not to feel like I’m missing out on something if everyone else is adopting it. Two of my close connections are already using it, so that’s giving me some confidence.’
- **Confidence Level:** 9

Analysis: This demonstrates how network connections become primary drivers for neutral innovations, with social influence overcoming individual skepticism.

Observation 3: Persistent Resistance

Social Network: Innovation Diffusion Visualization (Step 14)

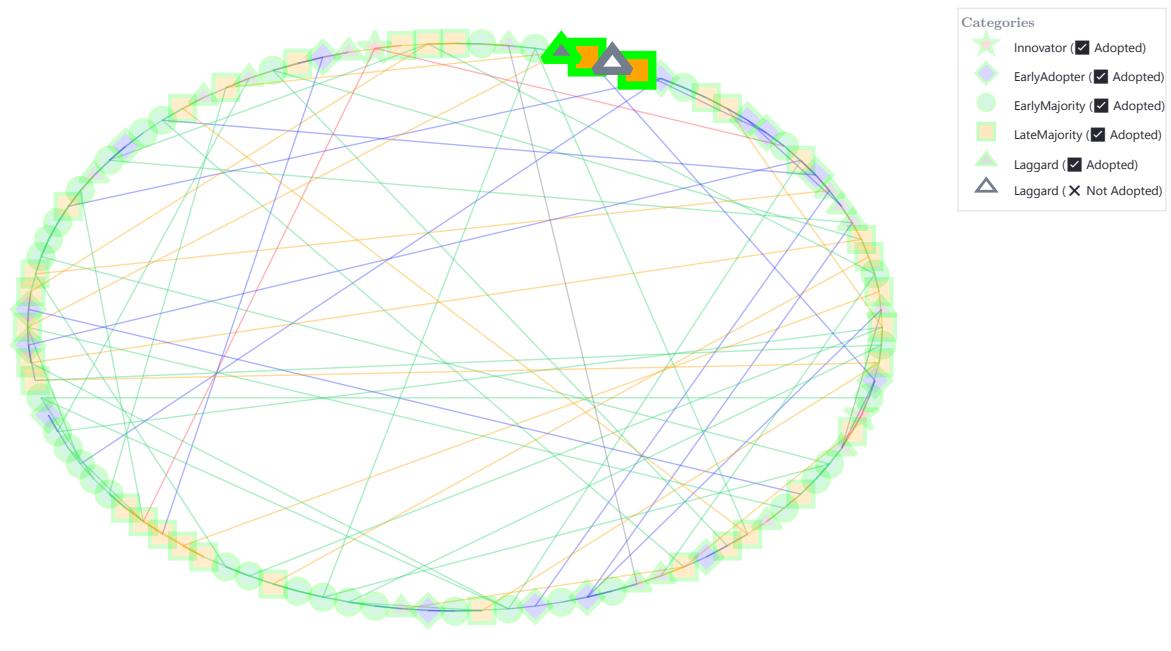


Figure 33: Laggard Network State at Steps 14-16

Observation: One Laggard displayed persistent resistance to adopting the innovation, even as social pressure mounted.

Agent Responses across Steps 14-16:

Step 14:

- **Reasoning:** ‘At this point, the vast majority of people are using this innovation, which makes me wonder if there’s something I’m missing. However, I’ve yet to see any concrete evidence that it offers significant advantages over my current methods.’
- **Thinking:** ‘The global adoption is almost complete, and two of my connections have adopted it, but I’m still hesitant due to the lack of clear benefits.’
- **Confidence Level:** 5

Step 15:

- **Reasoning:** ‘While the majority of people are using this innovation, I’ve yet to see any compelling reasons to switch from what I’m already doing. I’ll continue with my current methods until there’s more concrete evidence or a significant problem arises that this innovation can solve.’
- **Thinking:** ‘The global adoption is almost universal, and all of my connections have adopted it, but I’m still not convinced that it’s worth adopting.’
- **Confidence Level:** 4

Step 16:

- **Reasoning:** 'I'm not convinced that adopting this innovation is necessary at this time. My current methods are still working fine, and I'd rather wait until there's a compelling reason to change.'
- **Thinking:** 'It's clear that this innovation has been widely adopted and seems to be working well for others, but I still need to see a specific problem or issue that it can solve.'
- **Confidence Level:** 5

Analysis: This demonstrates that for some agents (especially laggards), even overwhelming social influence cannot overcome lack of perceived benefit when innovation attributes are neutral. The declining then stabilizing confidence levels reflect realistic skepticism persistence.

Chapter 6

Conclusions and Future Developments

6.1. Evaluation of Results

Although the simulation necessarily simplified many aspects of Rogers' Diffusion of Innovations theory, the results offer valuable insights into the dynamics of innovation adoption using Generative Agent-Based Models (GABMs). The findings highlight the crucial role of individual agent characteristics, social network structures and prompt engineering in shaping decision-making processes within innovation diffusion frameworks.

6.1.1. GABM Methodology Validation and Advantages

The implementation successfully demonstrated core GABM capabilities while revealing significant advantages and disadvantages over traditional Agent-Based Models (ABMs):

6.1.1.1. LLM-Based Decision Generation

Agents produced contextually appropriate responses reflecting their behavioral profiles. Analysis across all simulations revealed that most responses contained reasoning patterns aligned with their assigned adopter category characteristics, validating the prompt engineering approach.

6.1.1.2. Emergent Behavioral Complexity vs. Traditional ABMs

Unlike traditional threshold-based models producing binary outcomes, GABMs captured nuanced decision-making processes including:

- Social proof evaluation using both network and global information
- Risk assessment reflecting individual risk tolerance profiles
- Temporal consideration of adoption timing

Traditional ABM Output: Agent A adopts at step 5

GABM Output: Agent A adopts at step 5 because “While I have concerns about compatibility (score: 6), the high trialability (8) allows me to test it safely, and seeing 3 of my 4 connections succeed gives me confidence the risks are manageable.”

6.1.1.3. Cognitive Process Transparency

This granular insight enabled identification of:

- Specific attribute combinations creating adoption hesitancy
- Individual variation in risk interpretation within categories
- Network threshold effects varying by agent positioning and category

6.1.2. Innovation Attribute Impact and Decision Patterns

Analysis of agent reasoning patterns across innovation types revealed systematic differences:

6.1.2.1. Successful Innovation Reasoning

- Most decisions referenced “relative advantage” as primary motivation
- Risk concerns mentioned in a minority of responses
- Network influence cited as secondary factor in many cases

6.1.2.2. Balanced Innovation Reasoning

- Multi-attribute evaluation in most decisions (vs. fewer in successful scenarios)
- Network influence cited as primary factor in many cases
- Uncertainty expressions considerably more frequent than successful scenarios

6.1.2.3. Network-Category Interaction Effects

The simulation revealed that network positioning can override categorical tendencies:

- Well-connected Laggards adopted considerably earlier than isolated Laggards
- Peripheral Early Adopters delayed adoption compared to central peers
- Network effects overrode individual predispositions in some decisions

6.1.2.4. Dynamic Adoption Thresholds

Adoption thresholds emerged as dynamic constructs rather than fixed percentages:

- Late Majority required high network adoption for balanced innovations
- Lower network adoption needed for successful innovations

6.1.3. Key Empirical Insights

6.1.3.1. Innovation Quality as Diffusion Catalyst

Innovation attributes function as catalysts rather than simple adoption drivers. Successful innovations accelerated diffusion by reducing decision complexity, while balanced innovations forced greater reliance on social proof mechanisms.

6.1.3.2. Network-Aware Reasoning Validation

Agent responses explicitly referenced network conditions, with Late Majority and Laggard agents showing higher sensitivity to network adoption rates compared to Innovators and Early Adopters, confirming theoretical predictions.

6.1.4. Limitations and Boundary Conditions

6.1.4.1. LLM Hallucination and Inconsistencies

LLM-generated responses occasionally contained inaccuracies or inconsistencies, such as misreporting network adoption status.

Another problem occurred when agents decided to adopt while providing reasoning that did not align with their decision.

While these instances do not invalidate all decisions, they highlight the need for further research on mitigating hallucination effects and validating agent reasoning in GABMs.

6.1.4.2. Computational Constraints

The resource-intensive nature of LLM-based agents limited simulation scale and replication frequency, affecting statistical power and exploration of large-scale phenomena.

6.1.4.3. Simplification Artifacts

Focus on the decision stage alone created some unrealistic behaviors, particularly for unsuccessful innovations. Future implementations must balance computational feasibility with theoretical completeness.

6.2. Future Developments Directions

While most of these directions are focused on this project, they might also open up ideas for implementing GABMs in other domains.

6.2.1. Hardware Considerations

As GABMs become more complex and computationally intensive, future developments regarding GABMs could consider using distributed computing to enhance scalability and performance or smaller models as explained in an AWS blog post simulating energy supply chain model [23].

6.2.2. Experimenting with Alternative LLMs

As highlighted in Section 2.6.3 and observed in the better behavior of Llama3.1 8B over Mistral, the choice of LLM is a central factor in GABMs. Future work could investigate alternative LLM architectures or configurations to better understand their impact on GABM performance and outcomes.

Another promising direction is fine-tuning the LLMs used in GABMs to better suit specific domains or tasks. This could involve training on domain-specific datasets or incorporating additional contextual information to improve the relevance and accuracy of agent responses.

6.2.3. Exploring Alternative Network Structures

Beyond experimenting with different LLMs, future research could explore alternative network structures within GABMs. Since the project already supports various network types, such as scale-free and random networks, a logical next step is to examine how these structural variations influence agent interactions and diffusion patterns.

6.2.4. Comprehensive Implementation of Rogers' Five-Stage Model

Future work should prioritize the implementation of all five stages of Rogers' diffusion of innovations theory within the simulation framework. This comprehensive approach would encompass:

- **Pre-adoption stages:** Knowledge acquisition and attitude formation toward innovations
- **Decision stage:** The adoption or rejection decision process (currently implemented)
- **Post-adoption stages:** Implementation behaviors and confirmation processes

By capturing the complete adoption lifecycle, the model would provide deeper insights into temporal dynamics and enable more accurate predictions of innovation spread through social

networks. This holistic approach would also allow for the investigation of stage-specific factors that influence progression through the adoption process.

6.2.5. Refinement of Devil's Advocate Prompting Strategy

The project revealed that Devil's Advocate prompts create a complex trade-off: while increasing adoption bias, they simultaneously promote deeper analytical thinking and consideration of alternative perspectives. Future investigations should focus on:

- Developing balanced prompting strategies that maintain analytical depth while minimizing bias
- Exploring hybrid approaches that selectively apply critical evaluation prompts
- Investigating the optimal timing and frequency of Devil's Advocate interventions

An alternative is AutoGen's Reflection pattern, as described in Section 1.2.4.5.

This line of research could lead to more sophisticated decision-making frameworks that enhance both the quality and objectivity of agent reasoning processes.

6.2.6. Systematic Prompt Engineering Research

Given the significant impact of prompting on agent behavior observed in this study, future research should conduct systematic investigations into prompt variations and their effects. Priority areas include:

- **Micro-variation analysis:** Testing incremental changes in prompt wording and structure
- **Attribution expression enhancement:** Modifying decision prompts to elicit explicit reasoning about innovation attributes without introducing decision biases
- **Prompt optimization frameworks:** Developing methodologies for systematically improving prompt effectiveness

The current study's focus on identifying optimal prompts rather than exploring variations represents a significant opportunity for expanding our understanding of prompt-behavior relationships in GABM contexts.

6.2.7. Integration of Media Influence Mechanisms

A critical limitation of the current simulation is the absence of media influence on innovation adoption. Future developments should incorporate media channels to examine their impact on agent behavior and diffusion patterns. The AutoGen framework provides several implementation pathways:

Tool-based media integration: Utilizing AutoGen's tool capabilities (Section 1.2.3) to create media consumption tools that agents can choose to use. This approach would enable organic information acquisition processes where agents make autonomous decisions about media engagement, rather than receiving constant information streams.

Filtered media exposure: Implementing Message Filter Agents (Section 1.2.2.4) to create realistic media exposure patterns based on individual adopter profiles. This mechanism would prevent universal information saturation while maintaining authentic information asymmetries characteristic of real-world media consumption.

These implementations would significantly enhance the ecological validity of the simulation while providing insights into media's role in innovation diffusion processes.

6.2.8. Advanced Cognitive Modeling Through Society of Mind Agent

Another interesting development involves implementing the Society of Mind Agent (Section 1.2.2.3) to model the complex cognitive processes underlying innovation adoption decisions. This approach would:

- Simulate multiple internal “cognitive agents” representing different reasoning styles and perspectives
- Enable investigation of cognitive conflict resolution in decision-making contexts
- Provide insights into how diverse thought processes influence individual and collective adoption behaviors
- Allow for more nuanced modeling of personality factors and cognitive biases in innovation adoption

This multi-agent cognitive architecture could significantly enhance the psychological realism and explanatory power of the diffusion model.

6.2.9. Addressing LLM Hallucination

A notable challenge encountered was the phenomenon of LLM hallucination. In the context of this project, this occurred when agents made adoption decisions based on incorrect information (e.g., believing a connection had adopted when it was not the case) or provided reasoning that appeared inconsistent with their final choice. While such decisions are not necessarily invalid, these instances highlight the need for research on how to mitigate this phenomenon.

A potential way to address this issue is to utilize another LLM to validate the reasoning provided by agents, similar to the Reflection mechanism. While this could sound contradictory, it is an approach that has been successfully applied in other contexts, such as code generation (e.g., using one LLM to review the code generated by another). Future work could explore the feasibility and effectiveness of this approach in the context of GABMs.

6.3. Conclusions

This report has explored the integration of generative agent-based modeling (GABM) with innovation diffusion theory, demonstrating both the potential and limitations of LLM-driven agents for simulating complex social processes. By leveraging prompting strategies and the autogen framework, the project established that GABMs can provide access to agent reasoning processes and capture nuanced decision-making dynamics that traditional ABMs cannot replicate.

However, the research also revealed significant challenges that must be addressed. The hallucination problem, where agents make decisions based on incorrect information or provide reasoning inconsistent with their choices, represents a fundamental limitation requiring systematic investigation. Additionally, computational constraints and validation difficulties pose practical barriers to this approach.

These findings suggest that GABMs represent a distinct methodological approach with specific advantages and limitations rather than a replacement for traditional ABMs. GABMs offer unique strengths in cognitive process transparency, nuanced behavioral modeling and capturing complex decision-making dynamics, but face significant challenges including computational intensity, LLM hallucination issues and validation difficulties. Traditional ABMs maintain clear advantages for large-scale simulations, resource-constrained environments, and scenarios requiring high statistical power.

Bibliography

- [1] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, “The Impact of AI on Developer Productivity: Evidence from GitHub Copilot.” [Online]. Available: <https://arxiv.org/abs/2302.06590>
- [2] Q. Wu *et al.*, “AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation.” [Online]. Available: <https://arxiv.org/abs/2308.08155>
- [3] “Agents — AutoGen.” Accessed: Aug. 09, 2025. [Online]. Available: <https://microsoft.github.io/autogen/dev/user-guide/agentchat-user-guide/tutorial/agents.html>
- [4] “AssistantAgent — AutoGen.” Accessed: Aug. 09, 2025. [Online]. Available: https://microsoft.github.io/autogen/dev/reference/python/autogen_agentchat.agents.html#autogen_agentchat.agents.AssistantAgent
- [5] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, “Generative Agents: Interactive Simulacra of Human Behavior.” [Online]. Available: <https://arxiv.org/abs/2304.03442>
- [6] A. S. Vezhnevets *et al.*, “Generative agent-based modeling with actions grounded in physical, social, or digital space using Concordia.” [Online]. Available: <https://arxiv.org/abs/2312.03664>
- [7] Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno, “Generative Agent-Based Models for Complex Systems Research: a review.” [Online]. Available: <https://arxiv.org/abs/2408.09175>
- [8] G. D. Marzo, L. Pietronero, and D. Garcia, “Emergence of Scale-Free Networks in Social Interactions among Large Language Models.” [Online]. Available: <https://arxiv.org/abs/2312.06619>
- [9] S. Lai, Y. Potter, J. Kim, R. Zhuang, D. Song, and J. Evans, “Evolving AI Collectives to Enhance Human Diversity and Enable Self-Regulation.” [Online]. Available: <https://arxiv.org/abs/2402.12590>
- [10] G. Aher, R. I. Arriaga, and A. T. Kalai, “Using Large Language Models to Simulate Multiple Humans and Replicate Human Subject Studies.” [Online]. Available: <https://arxiv.org/abs/2208.10264>
- [11] R. Williams, N. HosseiniChimeh, A. Majumdar, and N. Ghaffarzadegan, “Epidemic Modeling with Generative Agents.” [Online]. Available: <https://arxiv.org/abs/2307.04986>
- [12] C. Gao *et al.*, “S³: Social-network Simulation System with Large Language Model-Empowered Agents”. [Online]. Available: <https://arxiv.org/abs/2307.14984>
- [13] T. Liang *et al.*, “Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate.” [Online]. Available: <https://arxiv.org/abs/2305.19118>
- [14] M. Larooij and P. Törnberg, “Do Large Language Models Solve the Problems of Agent-Based Modeling? A Critical Review of Generative Social Simulations.” [Online]. Available: <https://arxiv.org/abs/2504.03274>
- [15] E. Rogers, *Diffusion of Innovations, 5th Edition*. Free Press, 2003. [Online]. Available: <https://books.google.it/books?id=9U1K5LjUOwEC>
- [16] E. Kiesling, M. Günther, C. Stummer, and L. Wakolbinger, “Agent-based simulation of innovation diffusion: A review,” *Central European Journal of Operations Research*, vol. 20, pp. 183–230, 2012, doi: 10.1007/s10100-011-0210-y.
- [17] D. J. Watts and S. H. Strogatz, “Collective dynamics of 'small-world' networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998, doi: 10.1038/30918.
- [18] A.-L. Barabasi, “Scale-Free Networks: A Decade and Beyond,” *Science (New York, N.Y.)*, vol. 325, pp. 412–413, 2009, doi: 10.1126/science.1173299.
- [19] F. M. Bass, “A New Product Growth for Model Consumer Durables,” *Management Science*, vol. 15, no. 5, pp. 215–227, 1969, Accessed: Aug. 11, 2025. [Online]. Available: <http://www.jstor.org/stable/2628128>
- [20] H. Wang, T. Li, Z. Deng, D. Roth, and Y. Li, “Devil's Advocate: Anticipatory Reflection for LLM Agents.” [Online]. Available: <https://arxiv.org/abs/2405.16334>
- [21] A. Hagberg, P. Swart, and D. Chult, “Exploring Network Structure, Dynamics, and Function Using NetworkX,” 2008. doi: 10.25080/TCWV9851.
- [22] A. Grattafiori *et al.*, “The Llama 3 Herd of Models.” [Online]. Available: <https://arxiv.org/abs/2407.21783>
- [23] “Simulating complex systems with LLM-driven agents: leveraging AWS ParallelCluster for scalable AI experiments | AWS HPC Blog.” Accessed: Aug. 11, 2025. [Online]. Available: <https://aws.amazon.com/blogs/hpc/simulating-complex-systems-with-llm-driven-agents-leveraging-aws-parallelcluster-for-scalable-ai-experiments/>