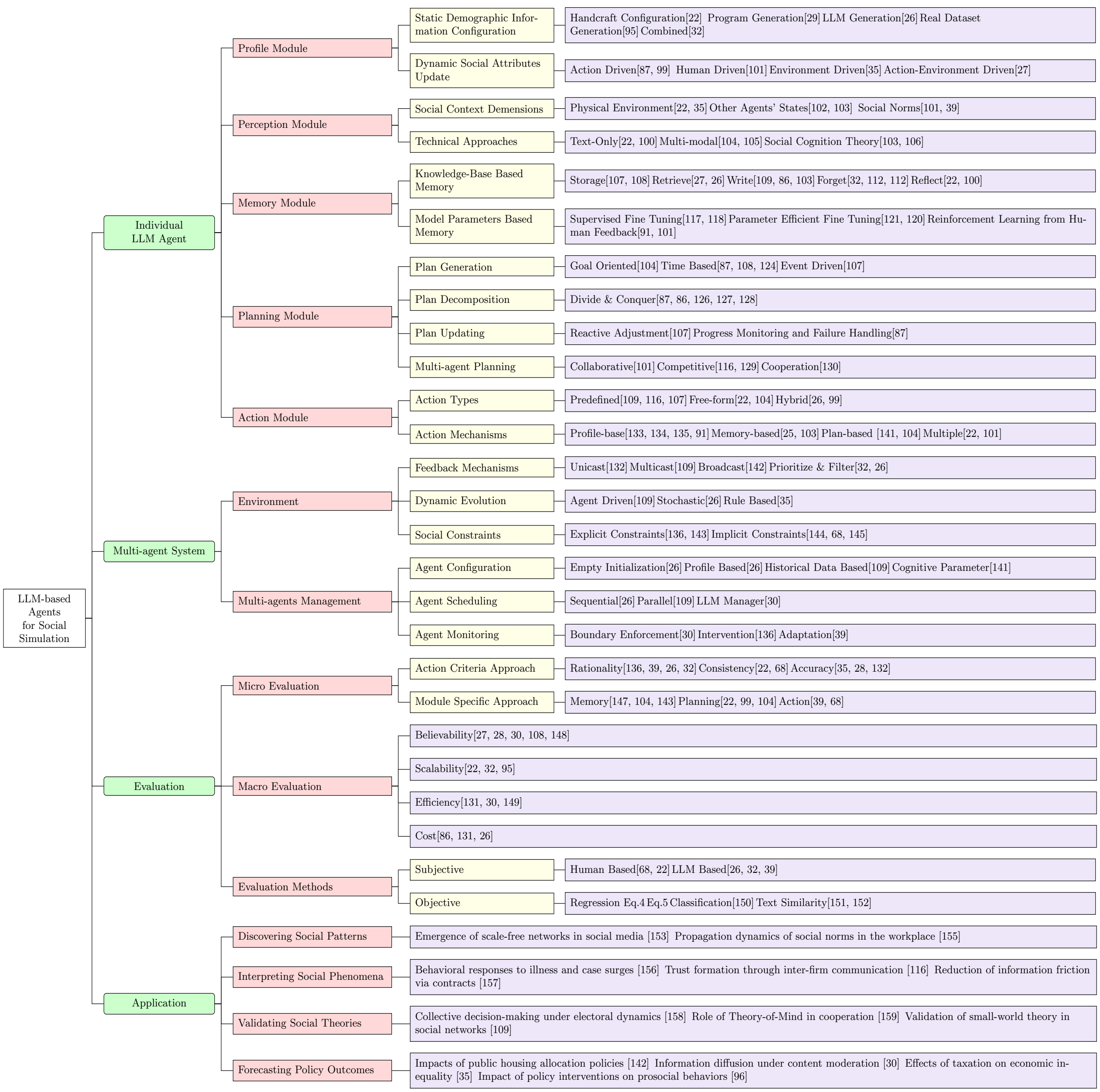
Graphical Abstract

A Survey on LLM-based Agents for Social Simulation: Taxonomy, Evaluation and Applications

Zixu Wang, Bin Xie, Bingbing Xu, Shengmao Zhu, Yige Yuan, Liang Pang, Du Su, Long Yang, Zixuan Li, Huawei Shen, Xueqi Cheng



Taxonomy of LLM-based Agents for Social Simulation.

Highlights

A Survey on LLM-based Agents for Social Simulation: Taxonomy, Evaluation and Applications

Zixu Wang, Bin Xie, Bingbing Xu, Shengmao Zhu, Yige Yuan, Liang Pang, Du Su, Long Yang, Zixuan Li, Huawei Shen, Xueqi Cheng

- We constructed a comprehensive taxonomy of the LLM agent-based social simulation system.
- We summarize the methodologies for constructing individual LLM agent in social simulation.
- We propose a comprehensive multi-agent framework for social simulation based on LLM agents.
- We propose a comprehensive evaluation metric system sorted out from both macro/micro and subjective/objective perspectives.
- We summarize the application of LLM agent-based social simulation systems. We also outline the ongoing challenges and propose several directions for future research in the field of social simulation.

A Survey on LLM-based Agents for Social Simulation: Taxonomy, Evaluation and Applications

Zixu Wang[a,b,1], Bin Xie[a,b,1], Bingbing Xu[a,*], Shengmao Zhu[a,b], Yige Yuan[a,b], Liang Pang[a], Du Su[a], Long Yang[a], Zixuan Li[a], Huawei Shen[a,b], Xueqi Cheng[a,b]

[a] *State Key Laboratory of AI Safety, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China*
[b] *University of Chinese Academy of Sciences, Beijing, 100190, China*

Abstract

Social simulation is a crucial tool in social science research, aiming to understand complex social systems. Recently, large language model (LLM) agents have demonstrated unprecedented human-like intelligence by leveraging the strong language understanding, generation, and reasoning capabilities of large language models. This paper conducts a comprehensive survey of social simulation empowered by LLM agents. We first review the evolution of social simulation paradigms and the development of LLM agents as background knowledge. Building on the foundational requirements for constructing a social simulator, we identify five essential capabilities that an individual LLM agent must possess. Correspondingly, we delineate five core modules that constitute the architecture of an LLM agent: (1) Profile Module for adaptive role-playing; (2) Perception Module for social context awareness; (3) Memory Module for continuous learning; (4) Planning Module for scenario-based reasoning; and (5) Action Module for dynamic decision-making. Additionally, we present a unified framework for LLM agent-based

[*] Corresponding author.

Email addresses: `wangzixu22s@ict.ac.cn` (Zixu Wang), `xiebin23s@ict.ac.cn` (Bin Xie), `xubingbing@ict.ac.cn` (Bingbing Xu), `zhushengmao23s@ict.ac.cn` (Shengmao Zhu), `yuanyige20z@ict.ac.cn` (Yige Yuan), `pangliang@ict.ac.cn` (Liang Pang), `sudu@ict.ac.cn` (Du Su), `yanglong@ict.ac.cn` (Long Yang), `lizixuan@ict.ac.cn` (Zixuan Li), `shenhuawei@ict.ac.cn` (Huawei Shen), `cxq@ict.ac.cn` (Xueqi Cheng)

[1] These authors contributed equally to this work.

social simulation systems, comprising the simulation environment, the agent manager, and interacting LLM agents. We also introduce a comprehensive evaluation metric that integrates macro- and micro-level as well as subjective and objective criteria. The representative applications are categorized into four scenarios: discovering social patterns, interpreting social phenomena, validating social theories, and forecasting policy outcomes. Finally, we identify the challenges and research opportunities in this field. Overall, this survey provides a systematic understanding of LLM agent-based social simulation, offering insights for future research and applications in this field.

Keywords: LLMs, Social Simulation, Agent, Social Science

---

1. Introduction

Social science, employing interpretative methods to explore social dynamics, cultural influences, and human motivations, focuses on understanding complex social systems. By integrating computational methods with insights from the social sciences, social simulation has emerged as a powerful analytical tool to dissect social theory[1]. It elucidates the genesis of social phenomena and probes the collective ramifications of intricate social dynamics[2].

The pioneering social simulations can be traced back to the 1950s, such as the growth models[3, 4]. These models use differential equations or dynamic system models to present the overall evolution of social phenomena, such as population growth and economic fluctuations[5]. These methods focus on the mathematical relationships among macro variables, but which ignores individual heterogeneity and the interactions between them.

In the 1970s, T.C.Schelling proposed the "self-organization idea", which is a theoretical and methodological groundwork for the agent-based modeling (ABM) paradigm. ABM focuses on individual agents and their interactions as the primary unit of analysis, which enables a bottom-up exploration of complex social structures emerging from agent interactions. It has remained the mainstream in social simulation for many years [6, 7, 8, 9]. However, agent behavior has heavily depended on predefined rules or heuristics in ABM, which decays the depth of interactions and often falls short of capturing the complexity of real human decision-making [10, 11, 12, 13]. Recent works (e.g., [14, 15]) apply advance method of machine learning for ABM, such as reinforcement learning, to enhance agent decision-making capabili-

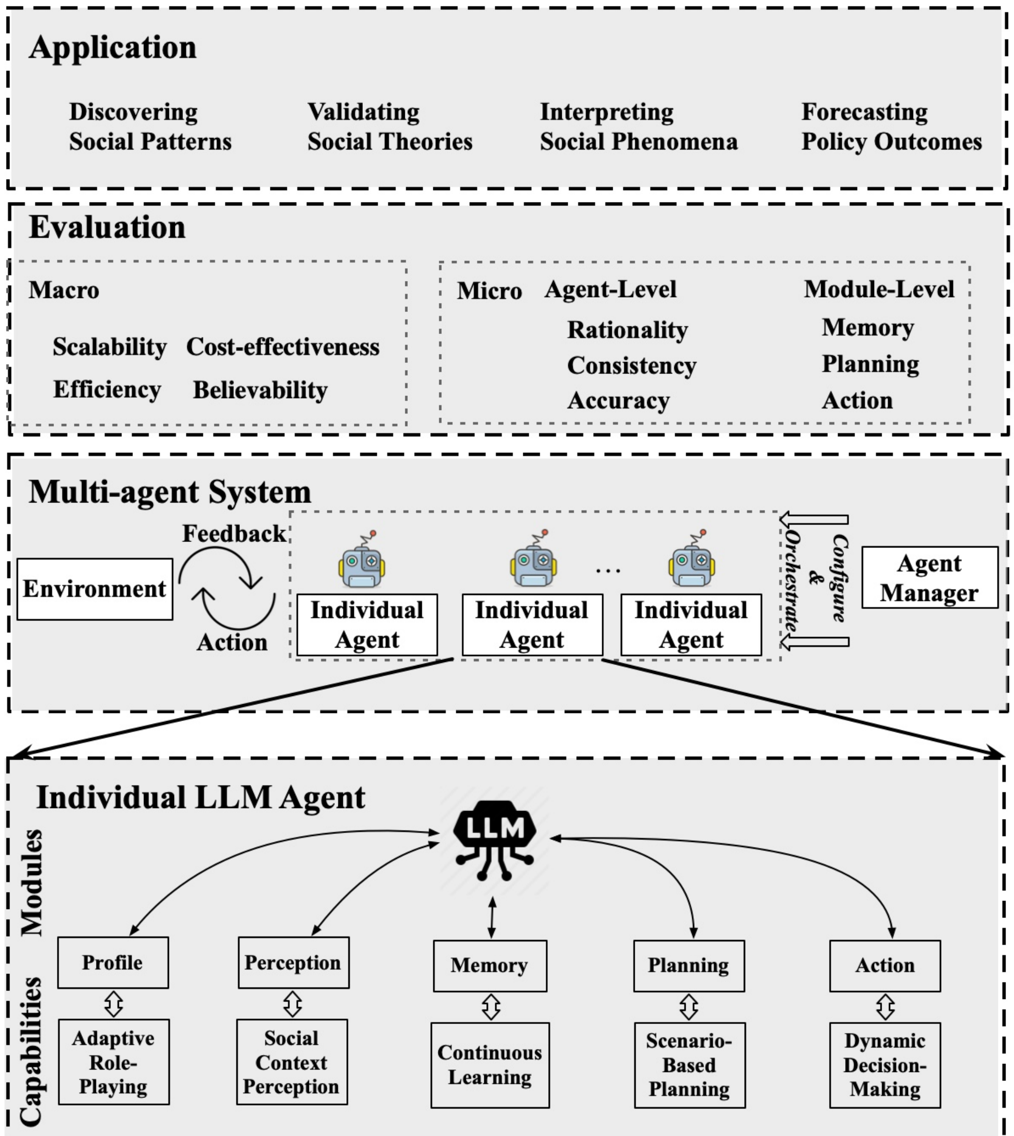


Figure 2: The outline of LLM Agent-Based Social Simulation System, including the construction of individual agent, the framework of multi-agent system, the evaluation metric, and representative applications.

ties. Although machine learning enhances agent decision-making for ABM, its reliance on extensive data and computational resources constrains its scalability, potentially reducing the realism of large-scale simulations [16, 17], and still struggles to generalize complex human behaviors—limitations that LLM-based modeling begins to address more effectively.

Large language models (LLMs) such as ChatGPT[2], Claude3.5[3], Llama 3 [18] and DeepSeek[4] which show great potential for human-like understanding and reasoning capabilities [19, 20, 21]. LLMs engages in sophisticated and adaptive interactions with outside environment and each other, it is possible to propose a promising path for simulating human social behaviors, which serving as the "brain" of simulation gents with perception, memory, planning, and action [13, 20].

Since 2023, researchers have conducted a large-scale exploration of the potential of LLM agents in many social science scenarios, such as social evolution [22, 23, 24, 25, 26], social media[27, 28, 29, 30, 31], recommendation system[32, 33, 34], economy[35, 36, 37, 39] and politics [40, 41] among which the pioneering work is [22], which implemented LLM agents in an interactive sandbox environment. These agents not only simulate daily human behaviors such as waking up, making breakfast, and going to work, but also spontaneously organize and participate in social events, like a Valentine's Day party.

Building agents based on LLM is a revolutionary new paradigm for social simulation. It allows agents to move beyond reliance on predefined rules. Additionally, it enables them to learn and adapt through LLMs, enhancing the simulation system's ability to capture dynamic changes and complex characteristics of social phenomena. However, there are few comprehensive works that have summarized the recent developments focusing on LLM agents based social simulation systems. Several pertinent reviews are centralized in areas about the construction of single agent [13, 42, 43, 44, 45, 46, 47], communication and cooperation mechanisms in multi-agent systems (MAS) [48, 49, 50, 51], and applications across fields like economics [52], chemistry [53], gameplay [54, 55], computational experiments [11], ABM [10, 56, 2], biomedical discovery [57], software engineering [58, 59], and industrial au-

[2] https://openai.com/blog/chatgpt/https://openai.com/blog/chatgpt/

[3] https://www.anthropic.com/claude/sonnethttps://www.anthropic.com/claude/sonnet

[4] https://www.deepseek.com/https://www.deepseek.com/

tomation [60, 61]. Although these reviews focus on LLM agents, they lack consideration for critical issues such as how to construct agents that meet the requirements of social simulation, how to organize single agents to build multi-agent social simulation systems, how to validate and evaluate the effectiveness of system simulations, and the applications of LLM-based social simulation systems in sociology and real-life scenarios.

Compared with recent works[62, 10, 11, 2], our study provides a more systematic and in-depth overview of the evolution of social simulation, dividing it into three key stages: mathematical modeling, agent-based modeling, and LLM-agent modeling. After reviewing the development of social simulation, we identified five essential agent capabilities and proposed a way to build individual LLM agent suited for social simulation. Based on this, we introduced the multi-agent system, evaluation, and application components,which together define the outline of this paper.

We therefore divide the LLM agent-based social simulation system into four interrelated components as shown in Fig.2: the individual LLM agent component constructs foundational agents to simulate single users; the multi-agent system provides the manager, interaction platform, and environment to enable large-scale social simulation; the evaluation component assesses the system at both micro and macro levels; and the application component identifies four representative use cases of the simulation system. Based on this foundation, we conducted an comprehensive study of each component, constructing the taxonomy of the LLM agent-based social simulation system as illustrated in Fig.3. In summary, the contributions of this survey can be summarized as follows:

- We summarize the methodologies for constructing individual LLM agent in social simulation. We propose that agents are endowed with five necessary capabilities to implement social simulation: adaptive role-playing, social context perception, continuous learning, scenario-based planning, and dynamic decision-making. These capabilities are supported by five key modules: profile, perception, memory, planning, and action.
- We propose a comprehensive multi-agent system for social simulation based on LLM agents, including environment, participating LLM agents, and agent manager.
- We propose a comprehensive evaluation metric system sorted out from

both macro/micro and subjective/objective perspectives.

- We summarize the application of LLM agent-based social simulation systems, which are classified into four typical scenarios: discovering social patterns, interpreting social phenomena, validating social theories, and forecasting policy outcomes. We also outline the ongoing challenges in the field of social simulation and propose several directions for future research.

The structure of this survey is as follows. Section 2 provides background information, including: the evolution of social simulation paradigms, the development of LLMs from pure language models to social intelligent agents, the comparison between traditional social simulation and LLM agent based methods, related surveys and the scope of our survey. Section 3 discusses the construction of individual LLM agent for social simulation, focusing on five key capabilities which corresponds to five modules. Section 4-6 summarizes LLM agent based social simulation systems' frameworks, evaluations and applications. Section 7 summarizes challenges of LLM agent based social simulation systems and suggests potential research directions. Section 8 offers a brief discussion and conclusion.

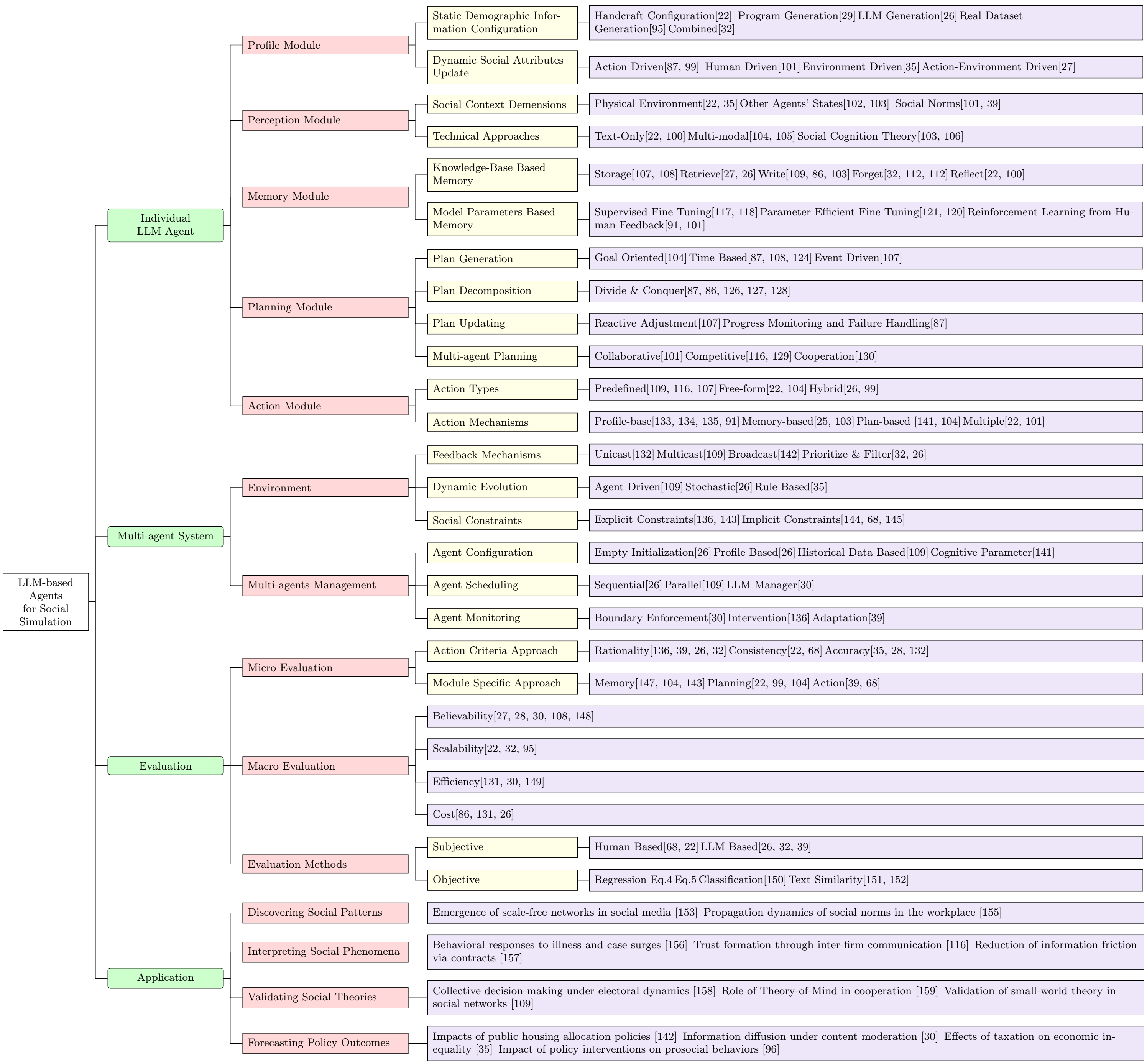


Figure 3: Taxonomy of LLM-based Agents for Social Simulation.

2. Background

This section aims to provide sufficient background knowledge for the entire survey. First, we review the evolution of social simulation paradigms across three phases: traditional mathematical modeling phase, agent-based modeling phase and LLM-based modeling phase. Second, we review the development of large language models (LLMs) from pure language models to social intelligence agents. Then, we compare three phases of social simulation approaches across scalability, complexity, assumptions, and replicability. In addition, we define the scope of our survey, specifying which papers are included and which are excluded. We also summarize related surveys and highlight the differences between our work and existing ones.

2.1. Evolution of Social Simulation Paradigms

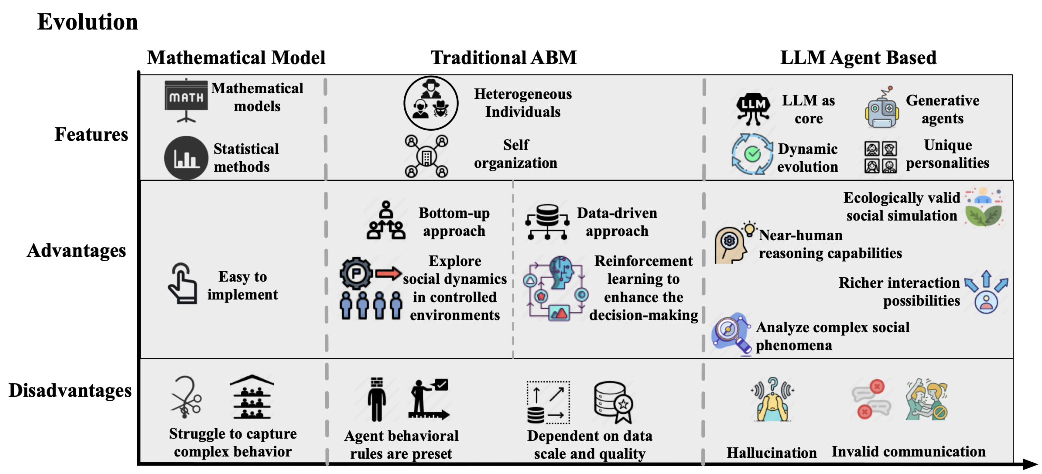


Figure 4: Evolution of Social Simulation Paradigms, including three major phases: mathematical modeling phase, traditional agent-based modeling (ABM) phase, and LLM agent-based phase.

Social simulation aims to understand, predict, and explain complex social dynamics through advanced modeling and simulation technologies. It is a methodological pillar of computational social science, an interdisciplinary field using computational methods to investigate social phenomena. Up to now, social simulation paradigms has traversed three pivotal developmental phases as shown in Fig.4 Initially, there was the static mathematical modeling phase. This was succeeded by the agent-based modeling (ABM) phase, during which dynamic interactions were introduced. Currently, it is in the stage centered around large language model agents. We will expound on the developmental trajectory of this discipline by examining each of these three phases in turn.

2.1.1. Mathematical Modeling Phase

This phase adopts methods that integrate the concept of system-level mathematical models to explain observed social phenomena and predict future trends under simplified assumptions [63]. They focus more on the overall structure and dynamic interactions within the system from a macro-perspective and struggle to capture the complexity and uncertainty of human behavior.

2.1.2. Agent-Based Modeling Phase

In the 1970s, Schelling's concept of "self-organization" enabled researchers to recognize that society is a complex system emerging from the interactions of heterogeneous individuals. This marked the birth of agent-based modeling (ABM). With increased computational power, "Swarm" and "NetLogo" were introduced in the 1990s as open-source tools that support visual and parameterized simulations, reducing the technical barriers of ABM. Based on these achievements, social science researchers started incorporating ABM [64, 65] to design agents with decision rules that simulate macro-level social system behavior through agent interactions. This "bottom-up" approach allowed social scientists to explore social dynamics in controlled environments, exemplified by Schelling's segregation model [66] and Axelrod's cultural dissemination model [67]. However, traditional ABMs face significant limitations: agent behavioral rules are typically preset and simplified, inadequately reflecting human decision-making complexity, particularly regarding adaptive behaviors under uncertainty and environmental changes [64]. Inspired by advances in machine learning, researchers have increasingly integrated data-driven approaches into ABM, using techniques like reinforcement learning to improve agent decision-making capabilities. This paradigm shift is highlighted in studies by Kavak et al. [14, 16], Venkatramanan et al. [15], and Keller & Hu [17]. However, data-driven ABMs depend on data scale and quality, which may limit model generalization and introduce bias.

2.1.3. LLM-Agent Modeling Phase

Recent breakthroughs in artificial intelligence, especially generative AI, have initiated a new paradigm shift in social simulation. Introducing LLMs as foundations for socially intelligent agents has enabled researchers to construct simulation environments with more complex and realistic human behavioral characteristics [22, 39]. These "generative agents" transcend simple rule sets, demonstrating near-human reasoning capabilities, emotional responses, and social interaction patterns [22]. LLM agents push the transformation of social simulation from static replication to dynamic evolution, where agents not only remember historical interactions but also develop unique "personalities" and make contextually appropriate decisions in social settings [68, 69]. The core advantages of LLM agent-based social simulation include:

- Improved ecological validity of social simulations that more closely approximate real human societies;
- Richer interaction possibilities unrestricted by predetermined behavioral rules;
- Enhanced capacity to observe and analyze complex social phenomena, including cultural evolution, opinion dynamics, group polarization, and norm formation [70, 71].

In summary, LLM agent-based social simulations provide social scientists with an unprecedented "digital laboratory" to safely and controllably test social science theories and explore the potential impacts of various social phenomena [68].

2.2. Development of LLMs

We focus on the key developmental stages where LLMs progress from early text generation models to sophisticated agent systems that support complex social interactions. This development reflects AI's expansion from pure language processing toward broader social cognition and interaction capabilities.

Breakthrough LLM advancements began with the Transformer architecture [72], followed by significant performance improvements through scaled models like GPT [73], PaLM [74], and LLaMA [75]. These models acquired robust language understanding and generation capabilities through large-scale pretraining, establishing foundations for subsequent developments. ChatGPT's release marked the transition of LLMs to practical application, demonstrating natural conversational abilities [19]. Subsequently, instruction tuning and reinforcement learning from human feedback (RLHF) enhanced model responsiveness to human intentions [76].

The transformation from LLMs to LLM agents represents a crucial AI research advancement, fundamentally shifting from passive language processing tools to proactive intelligent entities [13]. Researchers developed LLM-based agent by incorporating planning capabilities [77, 78], tool utilization [38, 79], memory management [80], and self-reflection [81], creating systems capable of autonomously completing complex tasks. Projects like AutoGPT [82] and BabyAGI [69] demonstrated LLM agents' ability to decompose tasks, formulate plans, and execute multi-step operations while prompting techniques like ReAct [77] and Chain-of-Thought [78] enhanced reasoning capabilities.

The progression from LLM agents to social intelligence agents represents the frontier of AI research. These agents not only execute tasks but engage meaningfully with humans and other agents in social environments [22, 68]. Park et al. [22] proposed a generative agent framework demonstrating how LLM-based agents can develop persistent "personalities" through memory, planning, and reflection, exhibiting human-like social behaviors in simulated environments. Zhou et al. [68] further explored multi-agent social interaction possibilities with their Sotopia framework, enabling researchers to study complex social dynamics in controlled environments. These social intelligence agents serve as minimal units in social simulations, generating decisions and behaviors that help social researchers analyze complex human mechanisms across dimensions of needs, emotions, cognition, and behaviors, ultimately facilitating the study of emergent phenomena in social groups.

2.3. Comparison Between Traditional Social Simulation And LLM Agent-Based Methods

We compare mathematical modeling, agent-based modeling, and LLM-based modeling across four key dimensions: scalability, complexity, assumption, and replicability. A detailed analysis is provided below.

- **Scalability.** Mathematical modeling uses macroscopic equations with low computational cost, making it easy to scale. Agent-based modeling simulates individual behaviors, but faces limitations in large-scale scenarios due to computational and design complexity. LLM-based modeling involves massive models and high costs in training, inference, and interaction, making scalability even more challenging.

- **Complexity.** Agent modeling evolves from non-existent (in mathematical models) to rule-based individuals (in ABMs), and further to intelligent, language-driven agents (in LLM-based models). Agent behavior shifts from homogeneous to personalized, and interactions from rigid to flexible. As agent intelligence and interaction richness increase, system complexity grows accordingly.

- **Assumption.** Mathematical models rely on strong macro-level assumptions and overlook individual heterogeneity. ABMs assume rule-based self-organization with limited learning, though data-driven variants improve flexibility. LLM-based models make minimal assumptions, using language to approximate human behavior, but may suffer from hallucinations that reduce simulation reliability.

- **Replicability.** Mathematical models are highly reproducible due to their simplicity and transparency. ABMs have moderate reproducibility, with results sensitive to rule design and parameter tuning. LLM-based models are the least reproducible due to stochastic outputs, opaque architectures, and prompt sensitivity.

2.4. Related Surveys

To clarify the unique contributions of this survey, we conducted a comprehensive analysis of existing literature in related fields, categorizing them into three types: LLM surveys, LLM agent surveys, and surveys addressing both LLM agents and social simulation.

2.4.1. LLM Surveys

Multiple comprehensive surveys on LLMs currently exist, focusing primarily on model architecture, training methods, capability assessment, and application scenarios. Zhao et al. [83] provided a comprehensive LLM survey, systematically analyzing LLMs across five dimensions: historical development, architectural design, training methods, evaluation benchmarks, and application domains. This survey presents a clear structure, first reviewing the evolution from early neural language models to modern Transformer architectures, followed by detailed discussions of pre-training and fine-tuning strategies, and exploration of LLM applications across various domains. Min et al. [84] focused on LLMs' language understanding and generation capabilities, with particular attention to emerging learning paradigms such as few-shot, zero-shot learning, and in-context learning. Their paper is structured around LLM capabilities, including reasoning ability, instruction following, dialogue capability, and knowledge application. Qiao et al. [85] concentrated more specifically on LLMs' reasoning abilities, systematically analyzing how different reasoning strategies (e.g., chain-of-thought, self-consistency) affect LLM performance. These LLM surveys provide a comprehensive overview of the technical foundations but primarily focus on the capabilities and technical characteristics of the models themselves, rather than applications of LLMs as social intelligence agents in simulated environments.

2.4.2. LLM Agent Surveys

As LLMs evolve, researchers are increasingly studying how to turn them into autonomous agents. Xi et al.[13] provided a comprehensive overview, defining LLM agents as systems that grant LLMs autonomy and initiative. Their survey covers three aspects: architecture, core functions (e.g., planning, tool use, memory, self-reflection), and applications. Wang et al.[42] focused on decision-making in LLM agents, outlining a process involving perception, planning, execution, and reflection. While these works offer valuable insights into LLM agent design and capabilities, they pay less attention to their use in social simulation.

2.4.3. LLM Agent Based Social Simulation Surveys

Recently, we noticed that works such as [10, 62, 11, 2] also cover the pics of LLM agents and social simulation. Compared with these works, our survey provides a more in-depth analysis of social simulation's development and a clearer explanation of the logic of introducing LLM. Specifically, we traced back the development history of social simulation and divided it into three stages: the mathematical modeling phase, the agent-based-modeling phase, and the LLM-agent modeling phase. We conducted a detailed analysis of the characteristics, advantages, and limitations of the three stages. Based on this analysis, we identified five key capabilities essential for effective social simulation and proposed five corresponding core modules for the LLM agent. Guided by modular thinking, we reviewed and analyzed recent related work, proposed a framework for LLM-agent-based social simulation, and outlined evaluation mechanisms from both macroscopic and microscopic, as well as objective and subjective perspectives. We also summarized its applications in sociological research and real-world contexts.

2.5. Survey Scope

Our survey focuses on social simulation systems that involve LLM-based agents. More specifically, we adopt the following inclusion and exclusion criteria for paper collection.

2.5.1. Inclusion Criteria

A paper is included if it meets at least one of the following criteria: (i) it constructs a social simulation system involving LLM-based agents; (ii) it emphasizes the general design of LLM-based agents with applications in social simulation, including the explanation or prediction of social phenomena.

2.5.2. Exclusion Criteria

Papers are excluded if: (i) an LLM is used solely as a passive question-answering tool, without incorporating agent-specific functionalities such as tool usage or active interaction; (ii) the work focuses only on the design of LLM-based agents, with no discussion of social simulation.

3. LLM Agent Construction for Social Simulation System

LLM agent plays a pivotal role in contemporary social simulation research and it aims to replicate believable human behaviors. In order for agents to exhibit the behaviors and characteristics of humans in the real society, each agent is required to have complex and multifaceted personality settings, observe the evolving social environment, draw on past experiences, formulate plans and decisions, and transform these decisions into tangible actions. To achieve this, we have summarized five key capabilities that agents need, namely, adaptive role-playing, social context perception, continuous learning, scenario-based planning and dynamic decision-making.

Based on these key capabilities, we introduce a unified framework for constructing LLM agents in social simulation systems which leverages five structured modules: profile module, perception module, memory module, planning module and action module as shown in Fig.2. These modules enable LLMs to have the aforementioned five capabilities respectively and collectively enable the transformation of LLMs from mere language models into social simulation agents endowed with complex social behaviors. In the following sections, we will detail the methodologies for constructing the five modules of LLM-based social simulation agents in sequence.

3.1. Profile Module for Adaptive Role-playing

Profile module is designed to endow LLM agents with adaptive role-playing capability, which enables LLM agents to flexibly assume and adapt roles based on dynamic social scenarios. LLM agents in social simulation must be able to play diverse social roles, because individuals often assume multiple roles in human society and sociological experiments require representativeness and diversity within the sample population. For example, a man might be an active content creator on social media, work as a pediatrician, and also fulfill the role of a father within a family. This versatility requires LLM agents to adapt their behavior patterns and language styles based on contextual cues, aligning with the specific traits and requirements of each role. Besides, in studies on language evolution to avoid social media regulation, LLM agents must represent both supervisors and participants, each bringing unique perspectives and constraints to the interaction.

Table 1: Methods and Representative Works of Profile Module.

	Method	**Representative Works**
Static Configuration	Handcraft Configuration	[22], [86], [87], [88], [89], [90]
	Program Generation	[29], [91], [92]
	LLM Generation	[26], [25], [22], [27], [93], [94]
	Real Dataset Generation	[95], [40], [96], [97]
	Combined	[32], [98], [30]
Dynamic Update	Action Driven	
	Self-Action Driven	[87]
	Interaction Driven	[99], [100], [26]
	Human Action Driven	[101]
	Environment Driven	[35]
	Action-Environment Driven	[27]

Existing profiling methods generally divides into static demographic information and dynamic social attributes and we summarize the methods and representative works of both profiling methods as shown in table 1. In the following, we introduce the two categories and their respective representative approaches in detail.

3.1.1. Static Demographic Information Configuration

Static demographic information refers to general and stable user attributes, such as ID, name, gender, age, personality traits, career, and interests. These attributes remain consistent over time and form the foundational characteristics of an agent's profile. They can be configured using one or a combination of the following methods.

- **Handcraft Configuration:** This method involves manually designing agent profiles, typically suitable for scenarios with a small number of agents. Researchers meticulously craft detailed background stories for each agent to emphasize their unique characteristics. For instance, in [86], background stories were created to initialize agents in a murder mystery scenario. This approach allows for high flexibility and alignment with specific research contexts. However, it becomes time-intensive as the number of agents increases.

- **Program Generation:** To enhance efficiency, programmatic methods leverage coding tools to automate agent configuration. For example, in [29], researchers used Python's faker library to randomly assign political leanings and other characteristics (e.g., age, education level, and Big Five personality traits) from predefined ranges. In [91], researchers used the names-dataset Python library to assign agent names, while ages are randomly selected between 18 and 64 years. The personality traits were based on the Big Five model, with each agent having a 50% probability of exhibiting positive or negative versions of a trait. Although these methods speed up the configuration process by utilizing existing packages, they lack the flexibility to define complex or scenario-specific attributes.

- **LLM Generation:** LLMs offer a more advanced approach by generating detailed agent profiles based on predefined input prompts or rules. For example, in [26, 25], researchers designed keywords-based prompts to instruct LLMs to generate brief biographies for agents, including personality, skills, and employment status (e.g. working as a software engineer or looking for new opportunities). Furthermore, LLMs can infer demographic characteristics from input data, such as personal descriptions or historical blogs, as demonstrated in [27], where attributes such as gender, age, and occupation are derived. Although LLMs can save significant time compared to manual design, issues such as hallucination and inherent biases still lead to problems like overly similar or inaccurate profiles.

- **Real Dataset Generation:** This method utilizes real-world datasets to configure agent profiles, ensuring diversity and authenticity. For example, [95] used the MovieLens-1M dataset to define agent profiles based on user ratings. Similarly, [40] used public information about U.S. senators to create profiles for agents in a political simulation. Using real-world data, this approach provides LLM agents with authentic characteristics that reflect human diversity and behavior patterns.

Combining the above methods can maximize their respective strengths. For example, in [32], researchers sampled 20 users and their interactions from the MovieLens-1M dataset. They used the final interaction of each user as ground truth, initialized agent profiles with other interactions, and then employed ChatGPT to summarize five prominent behavioral traits for inclusion in the profiles. These diverse methodologies provide researchers with tools to tailor agent profiles effectively for varied social simulation needs, balancing realism, efficiency, and adaptability.

3.1.2. Dynamic Social Attributes Update

Dynamic social attributes are fluid characteristics that evolve in response to external stimuli, social interactions, and an agent's behaviors. These include goals, needs, attitudes, emotions, relationships, cognition, values and so on. These attributes typically follow deterministic update rules, unified through state transition functions that formalize their dynamic changes as follows:

$$s_{t+1} = f(s_t, a_t)$$

where s_t represents the agent's state at time t, a_t denotes the action taken at time t, and f represents the specific transition logic. We have summarized three implementation methods of the transition logic f as follows :

- **Action Driven Transition Logic.** Agent attribute changes are often driven by agents' actions. According to who performs the attribute-altering action, they can be categorized into three types: self-action driven transition logic, interaction-driven transition logic, and human-driven transition logic. **1) Self-driven transition logic** refers to that the attribute changes of an agent are completely caused by its behaviors and not affected by other agents. For example, in the Humanoid Agents system [87], if an agent eats, its "fullness" need will increase; if it engages in rest or relaxation activities, its "energy" need will be replenished; **2) Interaction-driven transition logic** means that the change of an agent's attributes is caused by the interaction behaviors among agents. For example, Qian et al. [99] demonstrate how agents update their beliefs and intentions through communicative acts, thereby modifying their subsequent behaviors;In AgentVerse [100], the evaluator assesses the group's hydrogen storage station proposals and suggesting improvements, which drives agents to refine attributes like knowledge and decision-making for better solutions in subsequent collaborations;In MetaAgents[26], when recruiting agents interact with job-seeking agents, exceeding capability expectations may lead recruiting agents to revise their perceptions and adjust team formation strategies. Agents also reevaluate and update their beliefs when encountering information conflicting with prior convictions. **3) Human-driven transition logic** means that LLM agents' attribute changes due to the involve of human user. AutoGen[101] supports human participation in agent conversations, and different settings of the human input mode will lead to changes in the attributes of agents. When the UserProxyAgent is set to human input mode = 'ALWAYS', the agent needs to wait for human input and adjust its own behavior accordingly. Its decision-making process and the way of handling tasks will change, and the corresponding attributes (such as the degree of dependence on human feedback, the state of waiting for input, etc.) will also change.

- **Environment Driven Transition Logic.** The environment-based transition logic means that the change of an agent's attributes has nothing to do with the agent's own behavior and is completely determined by the environment. For example, in [87], different basic attributes have different decay rates. "Fullness" and "health" decrease by 1 approximately every 5 hours, "social" and "fun" decrease by 4 every 5 hours, and "energy" decreases by 5 every 5 hours. This natural decay mechanism simulates the changes in human needs in daily life, making the behavior of the agents more realistic. In [35], if the demand for goods in the consumer market continuously exceeds the supply, the wages in the labor market will keep rising to encourage more agents to participate in work and increase the production of goods. In this process, the market, as the environment, promotes the change of the agents' income attributes under the preset economic principles.

- **Action-Environment Driven Transition Logic.** This transition logic means that both actions and the environment jointly drive the changes in the attributes of an agent. For example, in [27], after integrating the user's historical actions, the event information sent by the environment to the agent, and the agent's current emotional information, the LLM predicts the emotion at the next moment.

To sum up, the profile module enables LLM agents to exhibit adaptive role-playing through a combination of static configuration and dynamic attribute updates. Together, these approaches support diverse, context-sensitive agent behaviors. Future work should further integrate these two aspects, leveraging real-time feedback and advanced models to enhance profile adaptability, coherence, and representational richness in complex social scenarios.

3.2. Perception Module for Social Context Awareness

Table 2: Methods and Representative Works of Perception Module

	Method	**Representative Works**
Social Context Dimensions	Physical Environment	[26], [35], [22]
	Other Agents' States	[26], [102], [103]
	Social Norms	[101], [39]
Technical Approaches	Text-Only	[22], [100]
	Multi-modal	[104], [105]
	Social Cognition Theory	[103], [106]

In real social scenarios, every individual will receive a large amount of information from the environment, other agents or even human users. These pieces of information are collectively referred to as social context. Perception module is designed to endow LLM agents with social context perception capability and enables the LLM agent to observe the evolving social environment. As shown in table 2, we divide social context into three dimensions: 1) Physical Environment; 2) Social Norms; 3) Other Agents' States. We present the current technical approaches for perceiving social context which are introduced in detail in the following.

3.2.1. Dimensions of Social Context Perception

Social context can be categorized into following three dimensions, reflecting the complexity of social environments that agents must navigate:

- **Physical Environment.** This refers to observations about the material environment as a whole or specific parts of it. For example, in [26], the perception module enables job-seeking agents at a job fair to gather valuable information from various sources, such as company posters and promotional materials. Similarly, in [35], the perception module allowed agents to capture the dynamics of the economic landscape, fostering the emergence of macroeconomic phenomena. Park et al. [22] implement perception modules that translate spatial coordinates and object properties into natural language descriptions, creating a coherent representation of physical surroundings.

- **Other Agents' States.** This involves perceiving the observable characteristics and behaviors of nearby agents. The scope of observable information often depends on proximity or the relationship between agents. In [26], the perception module allows generative agents to assess their peers' current level of engagement, enabling them to determine whether another agent is available for interaction or already occupied in a conversation. In [102], agents with closer relationships are granted access to more detailed information about one another's states, reflecting a nuanced model of inter-agent perception. Recent work by Sumers et al. [103] demonstrates how agents can infer others' mental states through observable behaviors, implementing a computational theory of mind.

- **Social Norms.** Beyond physical environments and agents states, social simulations increasingly incorporate the perception of social norms. Social norms are shared standards of acceptable behavior in a society or culture, guiding how individuals and groups interact and adapt to contextual expectations. Wu et al. [101] implement norm-aware agents that can perceive and reason about acceptable social behaviors in different contexts. Similarly, Horton et al. [39] demonstrates how LLM agents can be prompted to perceive cultural differences and adjust their behavior accordingly. Social norm is crucial for realistic social interactions, as human behavior is significantly shaped by cultural context.

3.2.2. Technical Approaches of Social Context Perception

The implementation of social perception modules in LLM-based agents generally follows several technical approaches:

- **Text-only Approach.** The most common approach involves converting environmental information into textual descriptions. This transformation can be rule-based or template-driven, as seen in [22], where spatial relationships are converted to text using predefined patterns. More sophisticated implementations use nested prompting techniques, where one LLM generates environmental descriptions for another to process [100].

- **Multi-modal Approach.** Recent advancements incorporate multimodal perception capabilities. Wang et al. [104] implemented agents that can process visual inputs alongside textual information, translating visual scenes into language representations. Similarly, Driess et al. [105] developed embodied agents that integrate visual perception with language understanding, bridging the gap between perception and action through multimodal processing.

- **Social Cognition Theory Appoarch.** More recent implementations integrate social cognition theory, particularly Theory of Mind (ToM) concepts. Sumers et al. [103] and Sap et al. [106] have implemented computational ToM models that enable agents to perceive and reason about others' mental states, beliefs, and intentions—a critical component of human social cognition.

To summarize, the perception module equips LLM agents with the ability to understand dynamic social contexts by capturing information from the physical environment, prevailing social norms and other agents' states. Technically, perception is achieved through text-based transformation, multimodal integration, and cognitive modeling approaches such as Theory of Mind. Together, these methods enable agents to interpret complex social cues, enhancing their contextual awareness and interaction quality in social simulations.

3.3. Memory Module for Continuous Learning

Table 3: Main Operations of Knowledge-Base Based Memory Approach

	Method	**Representative Works**
Storage	Text Based	[107], [40], [22]
	Vector Based	[108], [27]
Retrieve	Traditional Key Factors	Eq.1, Eq.2, Eq.3
	Cognitive Factors	[27], [26], [25]
Write	Append Based	[22], [109]
	Summary Based	[86], [25]
	Human Cognition Based	[110], [111], [32], [86], [91] [104],[103]
Forget	Time Based	[32]
	Similarity Based	[86], [112], [113]
	Capacity Based	[86], [112], [113]
Reflect	Extract Insights	[22], [77], [81], [104], [100]

The memory module is designed to endow LLM agents with the ability of continuous learning, enabling them to continuously accumulate, summarize, and reflect on historical experiences, so as to simulate the knowledge update and behavioral adaptation capabilities of human individuals in the real society. According to the storage methods of memory information, there are two complementary approaches to design memory module for LLM agents: external knowledge-base based memory approach and the internal model parameters based memory approach. The following sections explore these approaches in detail.

3.3.1. Knowledge-Base Based Memory Approach

This approach utilizes a meticulously designed external knowledge-base to store key memories, which can be viewed as an exclusive memory database for the LLM agent.

As summarized table 3, the main operations in the memory module are: memory storage, memory retrieval, memory writing, memory forgetting, and memory reflection.

Memory Storage. Agent memory is typically stored in text or vector formats:

- **Text-based Storage**: This comprises natural language descriptions of historical data. For instance, in [107], agents maintain records of economic activities including income, expenditures, customer flow, and daily reviews. Text memory can also adopt structured formats—in [40], agent memories are stored as JSON objects capturing dialogue interpretations at each simulation step. Similarly, [22] implements memory as chronologically ordered text entries describing agent experiences and interactions.

- **Vector-based Storage**: This more sophisticated approach encodes memories as high-dimensional vectors, often integrated with Retrieval-Augmented Generation (RAG) techniques. In [108], agents store memories as vector embeddings, facilitating efficient similarity-based retrieval. This method not only supports researcher analysis of agent behavior but also enables computationally efficient memory operations. Recent work by [27] further enhances this approach by integrating hierarchical vector indices to manage complex memory structures.

Memory Retrieve. The operation corresponding to memory storage is to retrieve relevant memories from the knowledge-base of the memory module for decision-making. The retrieval process typically evaluates three key factors:

- **Relevance:** This measures the semantic alignment between memory and the current context, typically computed using embedding similarity:

  $$rel_i = f_{rel}\left(e_t, m_i\right) \tag{1}$$

  where e_t represents the current context embedding and m_i represents the memory embedding.

- **Importance:** This reflects a memory's informational significance, often assigned during storage based on contextual factors:

  $$imp_i = f_{imp}\left(m_i\right) \tag{2}$$

- **Recency:** This prioritizes recent memories through temporally-weighted scoring:
$$rec_i = f_{rec}\left(t - t_i\right) \tag{3}$$
where t is the current time and t_i is when memory i was created.

Recent research has introduced additional retrieval factors. [27] incorporates authenticity by assigning credibility scores to memories based on their sources, improving reliability assessment. Meanwhile, [26] implements gist extraction where agents prioritize thematic elements and salient terms rather than complete textual entries, aligning with human memory processes. [25] further refines retrieval by implementing multi-stage filtering where initial coarse retrieval precedes more precise memory selection.

Memory Write. Agents must continually update their memories with new observations. Two primary methods facilitate this process:

- **Append-based Writing:** Agents directly add raw observations or minimal summaries to their memory repository. In [22], agents continuously record executed actions and observed behaviors in a chronological memory stream. Similarly, [109] implements sequential logging of authored tweets and reading experiences. While straightforward, this approach risks memory bloat, compromising retrieval efficiency and introducing noise over extended simulations.

- **Summary-based Writing:** Agents consolidate memories through LLM-generated summaries. In [86], similar memories are clustered and transformed into high-level summaries, with only these compact representations retained. [25] creates memory entries by synthesizing working memory with cooperative history, producing concise representations for future reference. While reducing redundancy, this approach introduces additional computational overhead through frequent LLM invocation.

- **Human Cognition based Writing.** To balance efficiency and comprehensiveness, many studies implement multi-tier memory architectures inspired by human cognition [110]. Systems described in [111, 32, 86, 91] employ sensory, short-term, and long-term memory components. Raw observations first enter sensory memory, then undergo LLM-based summarization into short-term memory. Periodically, long-term memory updates incorporate short-term content through consolidation processes. This architecture, exemplified by [104] and [103], enables efficient handling of both immediate context and historical patterns.

Memory Forgetting. To manage memory growth and reduce redundancy, forgetting mechanisms simulate human memory decay patterns. Drawing from cognitive psychology, [114, 115] prioritize the retention of recent and significant memories while gradually discarding older, less relevant content. We summarize forgetting strategies as follows:

- **Memory-Forgetting Over Time.** This type of method for memory forgetting heuristically designs a rule for memory decay over time, thereby imitating the situation where humans forget certain memories as time passes. In [32], forgetting probability follows a power function, where memory retention decreases over time according to established decay patterns. This approach ensures agents gradually forget memories while maintaining critical information.

- **Similar Memories Elimination.** Memories containing similar knowledge will lead to redundancy in the knowledge-base of the memory module. Therefore, in order to optimize memory storage, several systems employ similarity-based redundancy elimination [86, 112, 113], where embedding comparisons identify and remove memories closely resembling newer entries. Fixed window approaches offer another forgetting strategy.

- **Memory Capacity Limitation.** This type of memory forgetting strategy focuses on limiting the capacity of knowledge-base in memory module. In [35], agents maintain memory pools of 2L+1 entries, including economic conditions and decisions spanning L previous months. Similarly, [116] limits agents to the most recent 20 rounds of historical pricing information, demand data, profit records, and competitor pricing—effectively reducing noise and ensuring consistent pricing behaviors.

In fact, the above-mentioned forgetting strategies and their variants can be used in combination to achieve better memory management and reduce redundancy. Advanced implementations combine multiple forgetting strategies. [22] employs both time-based decay and importance-weighted retention, while [101] implements context-sensitive forgetting where memory retention varies with relevance to current tasks.

Memory Reflection. Beyond storage and retrieval, advanced memory systems incorporate reflection processes where agents analyze past experiences to extract insights and patterns. This enables agents to intelligently capture advantageous knowledge from their memories, simulating the process by which humans gain inspiration through reflection. In [22], agents periodically engage in reflective processing to identify behavioral patterns and generate higher-level insights. These reflections produce abstract knowledge representations that guide future decision-making without requiring detailed memory retrieval. Similarly, [77] implements a "revise" mechanism where agents critically evaluate past reasoning steps to improve future performance. More sophisticated reflection approaches include [81], where agents implement structured self-criticism and improvement cycles, and [104], where periodic reflection generates skill libraries and improvement strategies. [100] demonstrates how multi-agent reflective processes can identify emergent social dynamics and behavioral patterns across extended interactions.

3.3.2. Model Parameters Based Memory Approach

Table 4: Mainstream Post-Training Methods for Model Parameters Based Memory Module.

Methods	Representative Works
SFT	[117], [100], [118], [119], [120]
PEFT	[121], [122], [123], [120]
RLHF	[91], [22], [101], [104], [25]

Different from the approach that relies on external knowledge-base to explicitly store key memories, key memories can also be regarded as domain knowledge and injected into the internal model parameters of the LLM through post-training. In this approach, key memories are embedded as knowledge into the parameters of the LLM agent, thus directly utilizing these memories to profoundly modify the behavioral strategies of the acting agents. Currently, there are three mainstream post-training methods as shown in table 4 : supervised fine-tuning, parameter-efficient fine-tuning and reinforcement learning from human feedback which will be introduced in detail as follows.

Supervised Fine-Tuning(SFT). SFT adapts models using labeled datasets representative of target domains or behaviors. By training on carefully curated examples, models develop specialized capabilities while preserving general knowledge. Recent implementations demonstrate SFT's effectiveness in social simulation contexts. [117] fine-tuned models on corpus data reflecting specific demographic perspectives, producing agents that authentically represent diverse viewpoints. [100] employed domain-specific fine-tuning to create specialized agents for political discourse simulation. Similarly, [118] fine-tuned models on historical diplomatic communications to simulate international relations scenarios more convincingly. While powerful, SFT faces data collection challenges, particularly for specialized domains or underrepresented populations. Careful dataset curation remains essential to avoid reinforcing biases or producing stereotypical behaviors—concerns highlighted by [119] and [120].

Parameter-Efficient Fine-Tuning (PEFT). PEFT techniques modify only a small subset of model parameters, preserving core capabilities while enabling specialization with minimal computational overhead. Major PEFT approaches include Low-Rank Adaptation (LoRA) [121], where weight updates occur through low-rank decomposition matrices, and prompt tuning [122], which optimizes continuous prompt vectors while freezing model parameters. Adapter-based methods [123] insert trainable modules between transformer layers, creating specialized pathways while preserving base model knowledge. Social simulation research increasingly leverages these efficient approaches. [120] implemented adapter-based fine-tuning to efficiently model domain-specific expertise across different professions. These methods prove particularly valuable for resource-constrained research environments requiring multiple specialized agents.

Reinforcement Learning from Human Feedback (RLHF). RLHF incorporates human evaluator preferences to optimize model outputs, aligning agent behavior with human expectations and values. The RLHF process typically involves three stages: supervised fine-tuning on demonstration data, reward model training using comparative human preferences, and policy optimization through reinforcement learning. This approach has proven particularly effective for developing agents that exhibit socially appropriate behaviors and reasoning patterns. In social simulation contexts, [91] employed RLHF to create agents with varying degrees of skepticism regarding misinformation. [22] utilized preference-based optimization to align agent behaviors with expected social norms. [101] demonstrated how RLHF can produce agents that negotiate social interactions with appropriate communication styles and cooperation patterns. Recent work by [104] and [25] combines RLHF with multi-agent interaction data, enabling models to learn from both human feedback and emergent social dynamics. This hybrid approach shows particular promise for developing agents capable of realistic social behaviors in complex interactive scenarios.

Future research opportunities include developing more sophisticated integration mechanisms, addressing efficiency-effectiveness tradeoffs, and exploring how different learning approaches might best capture distinct aspects of human social cognition and behavior.

3.4. Planning Module for Scenario-based Planning

Table 5: Main functions and operations of Planning Module.

	Method	**Representative Works**
Generation	Goal Oriented	[104]
	Time Based	[87], [22], [108], [124]
	Event Driven	[107], [125]
Decomposition	Divide & Conquer	[87], [86], [126], [127], [128]
Updating	Reactive Adjustment	[107]
	Progress Monitoring	[87],
	Failure Handling	[87]
Multi-Agent	Collaborative	[101]
	Competitive	[116], [129]
	Cooperation	[130]

Planning module is designed to endow LLM agents with the capability of scenario based planning, enabling agents to formulate behavioral strategies across diverse situations thereby simulating real human behaviors such as daily scheduling , long-term planning and group coordination. As summarized in table 5, the planning module includes four functions: plan generation, plan decomposition, plan updating and multi-agent planning, which will be introduced in detail as follows.

3.4.1. Plan Generation

Plan generation serves as a roadmap that guides an agent's behavior during the simulation process. It outlines the sequence of actions the agent will take, helping to maintain consistency in the agent's behavior over time. For instance, in [87, 22, 131], the agent creates a rough plan for the entire day, which includes specific activities like waking up, eating breakfast, and going to work. AgentSociety [132] introduces a large-scale simulation of LLM-driven agents, highlighting the role of plan generation in understanding human behaviors and society. Plan generation typically takes several forms as shown below:

- **Goal-oriented planning:** This kind of planning generation focuses on achieving specific objectives, often organized hierarchically. In [104], agents translate high-level intentions into concrete action sequences.
- **Time-based planning:** There are some systems that generate plans along the timeline. For example, [87] and [22] implement daily schedules, while [108] extends planning to weekly horizons. [124] introduces progressive time frames where planning detail increases as execution time frames approach.
- **Event-driven planning:** There are also some agents that make contingency plans for possible future events. In [107], agents develop response plans for market conditions and customer feedback. [125] advances this through proactive planning where agents anticipate potential events and prepare contingencies before they occur.

3.4.2. Plan Decomposition

The plans generated by agents do not always directly translate into executable actions. Plans may sometimes be overly generic and require further refinement to translate into actionable steps. Complex plans require systematic decomposition into manageable sub-tasks, enabling more precise execution and evaluation. Most planning module use "divide and conquer" strategy to break down complex plans into simple and achievable sub-plans. Each sub-plan can be executed and evaluated individually, ensuring that the overall task remains manageable.

For example, in [87], the daily schedule generated by the agent is recursively decomposed into plans with one-hour intervals, followed by further breakdowns into 15-minute intervals. Besides, Lyfe-Agents[86] use a hierarchical option-action framework where a cognitive controller selects options and subgoals, and actions are chosen within options until termination. InterCode[126], decomposes plans by guiding agents to understand problems and devise step-by-step plans, like in Plan & Solve[127], and adjust plans based on execution feedback. In [128], a Multimodal Conversation Foundation Model (MCFM) is utilized to generate a solution outline based on the user's instructions and the conversation context, thus completing the task decomposition. In the scenario of multimodal long-content generation, MCFM will leverage its understanding of world knowledge to expand the user's brief instruction into a more comprehensive text description, clarifying the steps required to complete the task.

3.4.3. Plan Updating

Once a plan is formulated, it is not set in stone. As agents execute actions based on the plan, they need to dynamically update it in response to new information and changing circumstances. After completing an action, the agent must evaluate the progress of the plan. If the current plan is either completed or no longer feasible, the agent may need to adjust its planning. Here are some adjustment strategies:

- **Reactive adjustment.** This kind of adjustment modifies plans in response to immediate environmental changes. In [107], the agent analyzes all available information and either designs or modifies its strategy for the following day, which may include actions such as hiring a new chef or updating the menu.

- **Progress monitoring.** Another adjustment strategy is progress monitoring which evaluates plan completion status and effectiveness. [87] implements checks after each action to assess whether goals are being achieved as expected.

- **Failure handling.** Failure handling is the adjustment strategy that develops contingencies for plan failures. [87] demonstrates how agents generate alternative approaches when initial plans become unviable.

Recent implementations incorporate more sophisticated updating mechanisms. For instance, [99] implements belief revision protocols where agents update not only their plans but also their underlying assumptions based on new information. This iterative process ensures that the agent's behavior remains aligned with its goals.

3.4.4. Multi-agent Planning

Social simulation inherently involves multiple agents, requiring coordination mechanisms to produce coherent collective behaviors:

- **Collaborative Planning.** Collaborative planning is when multiple agents work together to create, coordinate, and execute plans for shared goals. For example, AutoGen[101] refines task planning via multi-round dialogues with diverse agents. In dynamic group chats, GroupChatManager selects speakers to foster discussions, enabling real-time planning adjustments for task changes.

- **Competitive Planning.** Competitive planning involves strategic decision-making in rivalry, analyzing opponents' moves to craft plans for advantage and goal achievement. For example, [116] demonstrates how market simulation agents develop strategies by analyzing competitor behaviors and potential responses. [129] shows how multi-agent debate can improve factuality and reasoning through structured competitive interactions.

- **Coopetition Planning.** Coopetition planning is strategic planning where entities collaborate to pursue shared goals while acknowledging and balancing individual competitive interests. For example, [130] realizes coopetition planning in multi-party negotiations via a complex negotiation game, interaction protocols, and a thought-chain framework, enabling agents to balance self-interest and collaboration for agreements.

3.5. Action Module for Dynamic Decision-making

Action module is engineered to grant LLM agents the capacities of decision-making and action execution. It empowers agents to make context-sensitive real-time decisions, ultimately enabling LLM agents to leverage all module-endowed capabilities to fulfill individual simulations in social scenarios. Specifically, we propose the decision workflow including all the modules. We also summarize the agent action types and the action mechanisms.

3.5.1. Decision Workflow

The decision-making process in LLM-based agents typically follows a cyclical workflow as shown in Fig.5 that consists of several interconnected phases.

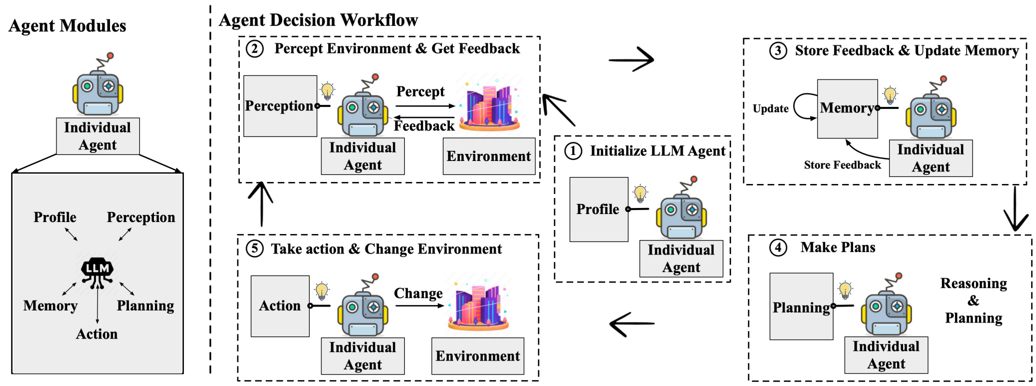


Figure 5: Modular architecture and decision-making workflow of an LLM-based individual agent. This figure illustrates the internal structure and cyclic workflow of an individual LLM agent in social simulation. The agent integrates multiple functional modules—profile, perception, memory, planning, and action(left part). The agent first performs initialization, and then iteratively cycles through four phases: perception, memory update, planning, and action execution(right part). The architecture enables dynamic, context-aware decision-making and continuous adaptation to evolving social environments.

- **Initialize LLM Agent.** The agent completes its own initialization through the profile module. Then it will cyclically execute the behaviors of the following four phases.

- **Percept Environment and Get Feedback.** The agent perceives its environment via perception module and get feedback including other agents' actions, environmental states, and new information.

- **Store Feedback and Update Memory.** The agent store the feedback into memory while updating these in-memory structures.

- **Make Plans.** Relevant experiences and knowledge are retrieved via memory module and the agent processes retrieved information to evaluate options and formulate action plans via planning module.

- **Take Action and Change Environment.** Based on the planning process, the agent selects and executes appropriate actions which will change the environment. And the agents will percept these new changes in the next round.

Recent research has enhanced this basic workflow with additional components. For instance, [22] incorporates reflection phases where agents periodically evaluate their decision patterns, while [103] implements recursive belief updating to refine subsequent decisions.

Table 6: Action Types and Mechanisms.

	Method	Representative Works
Action Types	Predefined	[32], [109], [116], [107], [35]
	Free-form	[22], [104]
	Hybrid	[26], [25], [99]
Action Mechanisms	Profile-base	[88], [133], [134], [135], [136], [137] [138], [139], [140],[39],[91]
	Memory-based	[22], [32], [25], [103]
	Plan-based	[87], [141], [26], [104]
	Multiple	[22], [101]

3.5.2. Action Types

Agent actions can be categorized based on their implementation approach and degree of constraint:

- **Predefined Actions.** This approach restricts agents to selecting from a predetermined set of actions designed for specific research contexts or simulation goals. For example, in social media simulations [32, 109], agents might choose from options like liking posts, sharing content, or commenting. In market simulations [116, 107], actions might include pricing decisions, product development, or marketing strategies. Predefined action spaces offer several advantages: they ensure actions remain within reasonable bounds, reduce computational complexity, and facilitate comparative analysis across simulation runs. However, they limit agent adaptability and potentially oversimplify complex social behaviors. Recent work by [35] demonstrates how carefully designed action spaces can balance constraint with sufficient expressivity to capture economic decision nuances.

- **Free-form Actions.** Free-form implementation allows agents to generate actions dynamically based on context without restrictive predefined options. This approach leverages the generative capabilities of LLMs to produce flexible, creative responses to diverse scenarios.

In [22], agents interact through natural language conversations without predetermined response options, enabling emergent social dynamics. [104] implements free-form action generation where agents develop novel approaches to environmental challenges. This flexibility enhances realism in open-ended social simulations but introduces risks of LLM "hallucinations" or contextually inappropriate responses.

- **Hybrid Actions.** The selection between predefined and free-form approaches depends on research objectives, with many implementations combining elements of both to optimize the tradeoff between control and adaptability. Thus researchers have developed hybrid approaches. [26] implements action verification mechanisms where generated free-form actions undergo consistency checking before execution. [25] employs action templates that provide structure while allowing contextual variation. [99] demonstrates how constrained free-form actions can balance flexibility with simulation stability.

3.5.3. Action Mechanisms

The action mechanism defines how agents determine their next actions based on internal states and external inputs. These mechanisms vary in complexity and can be classified into three primary categories.

- **Profile-based Action Mechanisms.** In profile-based mechanisms, agent actions derive primarily from predefined attributes stored in their profiles. These attributes typically include personality traits, preferences, goals, or demographic characteristics that guide behavior. Agents with profile-based mechanisms often function like role-players, using their assigned identities to inform decisions without extensive memory or planning capabilities. For example, agents participate in social games like Avalon [88, 133], Mafia [134], or Werewolf [135, 136] by acting according to their assigned character roles and game rules. Similarly, profile-based agents serve as subjects in psychological experiments [137], interviewees in social surveys [138], or participants in simulated political [139] and economic scenarios [140, 39]. Recent work by [91] demonstrates how profile-based decision-making can simulate varied degrees of susceptibility to misinformation based on personality attributes.

- **Memory-based Action Mechanisms.** Memory-based mechanisms leverage stored experiences to inform current actions, creating more contextually aware decision-making. These agents retrieve and analyze relevant past experiences to identify patterns, anticipate outcomes, and select appropriate actions for their current situation. For instance, in [22], agents accumulate memories of interactions with other agents, using this history to develop nuanced social relationships that influence subsequent behavior. [32] implements recommendation agents that track user preferences through interaction histories to personalize content suggestions. [25] demonstrates how memory-based decision-making enables agents to adapt communication strategies based on the success of previous exchanges. Recent work by [103] shows how episodic memory retrieval enables more socially coherent interactions by maintaining consistency across multiple encounters.

- **Plan-based Action Mechanisms.** Plan-based mechanisms structure decision-making around explicit goal-directed plans that define action sequences. These agents formulate strategies to achieve objectives and execute actions according to these plans, adjusting as needed when circumstances change. In [87], agents generate daily schedules that guide their activities, breaking them down into progressively finer time intervals for implementation. [141] enhances planning through Chain of Thought (CoT) and Chain of Action (CoA) techniques, allowing agents to extract declarative and procedural memory during reflection phases. [26] implements hierarchical planning where high-level goals decompose into concrete action sequences. Recent research by [104] demonstrates how agents can use recursive planning to navigate complex environments, continuously refining their approaches based on feedback.

Some sophisticated agents employ multiple behavioral driving mechanisms simultaneously to fully leverage the agent's profile, memory, and planning capabilities in guiding their actions. For example, [22] integrates profile attributes with memory retrieval and planning processes to create a comprehensive decision framework. Similarly, [101] implements a hybrid architecture where profile-based preferences influence planning processes, which in turn guide action selection based on the memory of past outcomes.

3.6. Summary of This Chapter

In this chapter, centering around five key modules: the profile module, the perception module, the memory module, the planning module, and the action module, we summarize the methods for constructing an LLM agent for social simulation. We also introduce the workflow of how a single agent takes actions when confronted with an external social context. All these five modules and the workflow enable the LLM to evolve from a mere language model into a social simulation agent endowed with complex social behaviors, providing a foundation for the the construction of social simulation system.

4. The Construction of Social Simulation System

Social simulation system is an essential tool for exploring social evolution and dynamics by simulating individual behaviors within a specific environment. Following agent design, the next step is to build the simulation system, which consists of an environment and multiple agents. Achieving this goal involves addressing three critical aspects: (1) What functions must the environment fulfill? (2) How does the simulation system schedule and manage multiple agents? (3) How does the social simulation system operate in general? The following section will discuss each of these aspects in detail.

4.1. Environment

Table 7: Construction of Environment in LLM Agent based Social Simulation Systems

	Method	Representative Works
Feedback Mechanisms	Unicast	[132]
	Multicast	[109]
	Broadcast	[142]
	Prioritize & Filter	[32], [26]
Dynamic Evolution	Agent Driven	[109], [22]
	Stochastic	[26]
	Rule Based	[35], [116]
Social Constraints	Explicit	[22], [136], [143]
	Implicit	[144], [68], [35], [145]

The environment is a crucial component of the social simulation system. It mimics the factors in the real society other than specific individuals, such as scenarios, news, personnel structures, and message transmission structures. Most importantly, the environment promotes agents' behaviors like learning and communication among agents by providing feedback information. Besides, the environment itself also undergoes continuous dynamic evolution as the system simulation progresses, providing a platform to test theories, predict outcomes, and analyze the impacts of various factors on agent behaviors. In addition, the environment serves as an entry point for researchers to add prior knowledge and constraints to the system. As shown in table 7, we summarize the feedback, evolution and social constraints mechanisms of environment in LLM agent based social simulation systems in this section.

4.1.1. Feedback Mechanisms

LLM agents rely on textual feedback from the environment to make decisions, take actions, and learn, enabling emergent behaviors in the process. This feedback primarily involves information on the environment's state (either partially or entirely). To ensure timely feedback for each agent, the environment must manage and distribute feedback messages effectively. We summarize feedback messages distribution mechanisms as follows:

- **Unicasting.** This approach directs feedback messages to individual agent, enabling private communications. For instance, in [132], agents can receive quality resources shared privately by friends, facilitating the formation of information asymmetry within social networks.

- **Multicasting.** Multicasting feedback messages distribution mechanism targets a specific group of agents, modeling information dissemination within communities or networks. In [109], social media simulations use multicasting to distribute content to followers or interest groups.

- **Broadcasting.** Broadcasting means feedback messages are sent to all agents simultaneously, creating shared knowledge. For example, in [142], participants receive information broadly distributed through social media platforms, simulating public information flows.

The selection of feedback messages distribution mechanism depends on simulation requirements and the nature of modeled interactions. Advanced environments integrate algorithms to prioritize and filter feedback messages based on relevance and urgency. For example, in [32], agents receive posts and tweets from recommendation systems that filter content based on personalized relevance scores. Similarly, [26] implements attention mechanisms where agents process environmental information according to relevance hierarchies.

4.1.2. Dynamic Evolution

The environment in a social simulation system is not static but continually evolving, reflecting the dynamic nature of social systems. We summarize the social simulation system's evolution pathways as shown bellow:

- **Agent-driven.** Environment states change in response to agent actions. In social network simulations like [109], when a user posts a new tweet, the environment updates by adding the content to information streams and recommendation systems. Besides, agent interactions modify the social landscape in[22], creating new opportunities and constraints for future interactions.

- **Stochastic processes.** Environment states also change due to random events such as breaking news, natural disasters, or unexpected opportunities which introduce unpredictability as in real human society. In [26], adding new staff or new positions is a randomly occurring situation for the recruitment simulation system. This makes the information and personnel composition in the recruitment scenario more complex, thereby changing the original environment and simulating the situational changes that agents must respond to as a result of random events.

- **Rule-based changes.** Besides, predefined rules can dictate environmental transitions. In [35], economic environments evolve according to supply-demand equations, inflation rates, and market dynamics. [116] employs a Bertrand Duopoly Game with Differentiable Goods to model competition between two companies offering substitutable products. Each company operates as a Bertrand firm engaged in price competition with profit-maximizing objectives under complete information, with demand functions $q_i(p_i, p_j)$ determining market responses.

These evolution mechanisms ensure simulations remain adaptive and realistic, providing closer representations of actual social dynamics.

4.1.3. Social Constraints

Environments impose social constraints, such as behavioral norms or interaction rule, on agents, shaping their actions to reflect real world social dynamics. This adds authenticity and complexity to social simulations, as agents must navigate both environmental challenges and societal expectations, enhancing the realism of their interactions. We divide the social constraints in the social simulation environment into two categories: explicit and implicit.

- **Explicit constraints.** Direct limitations on agent actions. In [22], daily schedules limit when agents can perform certain activities. In [136], agents are restricted to operate within designated areas, with specific locations limiting simultaneous occupancy. [143] implements resource limitations that agents must consider when planning actions.

- **Implicit constraints.** Socially-derived limitations. In [144], norms emerge from agent interactions, subsequently constraining future behaviors. [68] incorporates cultural contexts that influence acceptable behaviors and communication styles. [35] implements market regulations that constrain economic agent behaviors. [145] demonstrates how reputation systems create emergent constraints on agent behavior as they attempt to maintain positive social standing.

These constraints create realistic, nuanced simulations that reveal how environmental factors shape social dynamics and emergent behaviors. It worth noted that the design of effective simulation environments ultimately requires balancing fidelity with social constraints, ensuring environments are rich enough to produce meaningful results while remaining computationally feasible and analytically tractable.

4.2. Multi-Agents Management

In a social simulation system, the involvement of numerous agents, often numbering in the hundreds or thousands, necessitates efficient management and scheduling strategies to maintain system coherence. The Agent Manager serves as a centralized intermediary, orchestrating agent behaviors and ensuring synchronization throughout the simulation. This component, sometimes referred to as a meta-agent when implemented as a specialized type of agent, fulfills several critical roles as summarized in table 8: initializing multi-agents' configuration, coordinating interactions among multi-agents and with their environment, and managing multi-agents behaviors to prevent unrealistic actions. The following sections elaborate on these critical roles agent manager plays for multi-agents management.

Table 8: Summary of agent configuration, scheduling and monitoring of multi- agents management

	Method	Representative Works
Agent Configuration	Empty Initialization	[26]
	Profile Based	[26]
	Historical Data Based	[109]
	Cognitive Parameter	[141]
Agent Scheduling	Sequential	[26]
	Parallel	[109]
	LLM Manager	[32], [30]
Agent Monitoring	Boundary	[30]
	Intervention	[136]
	Adaptation	[39]

4.2.1. Agent Configuration

The first responsibility of the Agent Manager is to configure multi-agents, a process that includes a critical initialization step. This step involves setting up the agents' profile information—such as demographics, personality traits, goals, and social relationships. For instance, Sotopia [68] initializes agents with diverse personality profiles based on the Big Five personality traits, along with social backgrounds and personal preferences that influence their decision-making processes. In addition, Jinxin et al. [141] propose a tree-structured role model for role assignment, detection, and maintenance, which facilitates agents in organizing and recalling distinct roles.

Beyond profile initialization, memory setup is also essential and can be implemented through the following approaches.

- **Empty initialization.** In simulations focused on emergent behavior, such as in Park et al. [22], agents begin with minimal memory structures that develop organically through interactions.

- **Profile-based initialization.** Systems like MetaAgents [26] initialize agent memory with their pre-configured profile information, providing a foundation for consistent behavior aligned with their designed characteristics.

- **Historical data initialization.** Chuang et al. [109] initialize agent memories with social network structures derived from real-world data, establishing initial relationship patterns that evolve during simulation.

The Agent Manager may also configure internal states such as emotional status, belief systems, and other cognitive parameters. For example, Jin et al. [141] propose a tree-structured character model that helps agents establish and maintain different roles by organizing memory hierarchically, facilitating more coherent role-based interactions.

4.2.2. Agent Scheduling

In multi-agent scenarios, designing effective scheduling strategies that realistically simulate human interaction patterns presents significant challenges. The Agent Manager must implement scheduling mechanisms that balance computational efficiency with behavioral fidelity. Common approaches include:

- **Sequential Scheduling.** This method applies to interactions with direct causal dependencies between agent actions. Agents are scheduled in a specific sequence to ensure accurate simulation of action dependencies. For instance, in Li et al. [26], recruitment scenarios follow a strict sequence: job-seeking agents first select companies, followed by recruiting agents conducting interviews, and finally hiring decisions are communicated. This sequential approach ensures the logical progression of interdependent actions.

- **Parallel Scheduling.** This approach handles interactions where agents primarily affect the environment rather than directly impacting other agents. Since actions do not directly depend on other agents' behaviors, they can execute simultaneously. For example, Chuang et al. [109] implement parallel scheduling for social media simulations where multiple users simultaneously engage with content through likes, shares, and comments in each simulation round. The environment then aggregates these actions and synchronizes state updates.

The Agent Manager often implements models determining which agents interact with each other. Wang et al. [32] employ Pareto distributions to control agent activity levels and interaction frequencies, creating realistic patterns of engagement. Advanced scheduling systems also incorporate temporal models to manage simulation time progression, as seen in OASIS [30], where different timescales (hourly, daily, weekly) govern different types of LLM agent interactions.

4.2.3. Agent Monitoring

Due to issues such as hallucination and over-alignment inherent in LLMs, agents may exhibit behaviors that exceed expected boundaries. The Agent Manager is tasked with monitoring agent behaviors and states, modifying interactions, or adjusting scheduling based on real-time feedback.

- **Boundary Enforcement.** Agent Manager establishes constraints on agent behaviors to maintain simulation realism. In OASIS [30], a time engine manages each agent's activity level per hour by referencing historical interaction frequency or applying custom settings, preventing agents from becoming overly active or passive.

- **Intervention Systems.** When agent behaviors diverge from expected patterns, intervention mechanisms can redirect them. Xu et al. [136] implement a referee system that evaluates agent actions against predefined rules, rejecting inappropriate behaviors and requesting alternatives when necessary.

- **Adaptation Mechanisms.** Advanced Agent Managers implement dynamic adaptation based on observed behaviors. Horton et al. [39] describe systems that adjust agent parameters during simulation runtime in response to performance metrics, gradually optimizing agent behavior patterns.

These monitoring approaches enhances simulation adaptability to unexpected developments, ensuring that agent behaviors remain aligned with the overarching simulation objectives while still allowing for emergent phenomena.

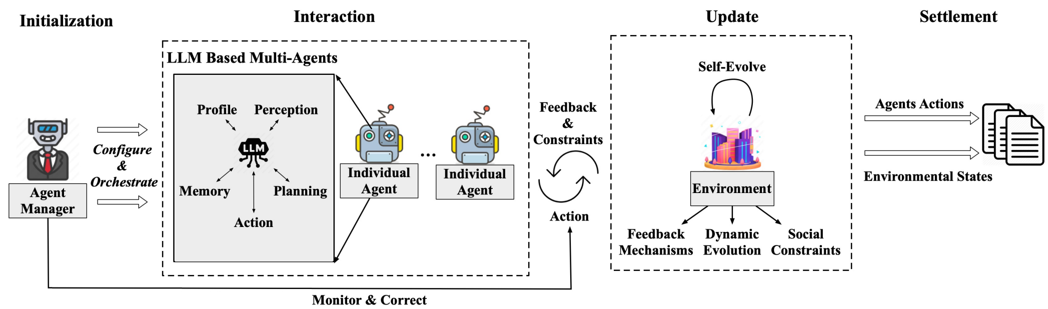


Figure 6: The general framework of an LLM agent-based social simulation system. During the Initialization phase, the system configures the environment and the agent manager sets up individual LLM agents. In the Interaction phase, agents perceive the environment and act based on internal modules, while the agent manager monitors and corrects behaviors. The Update phase is driven by the environment, which evolves in response to agent actions and internal dynamics. Finally, in the Settlement phase, the system logs agent actions and environmental states for analysis.

4.3. General Operation Workflow

In this section, we describe the general operational workflow of a social simulation system, which is divided into four phases: Initialization, Interaction, Update, and Settlement. These phases encompass the fundamental steps of most existing simulation systems as shonw in Fig.6. We will now demonstrate the specific tasks involved in each phase, drawing from notable implementations in the literature.

4.3.1. Initialization

The initialization phase serves as the starting point for social simulation, encompassing two primary components.

The first part is initialization of the environment, which involves configuring the initial time, establishing a virtual or physical space for agent interactions, and initializing the states of interactable objects and scenario constraints. For example, Sotopia [68] provides some interfaces to establish social scenarios with specific locations, available objects, and interaction rules that constrain subsequent agent behaviors.

The second part is initializing the LLM agents participating in the simulation. This includes determining the number of agents, configuring each agent's profile and memory, and setting up the predefined characteristics or behaviors, which is managed primarily by the agent manager. For instance, in Generative Agents [22], the agent manager establishes each agent's starting attributes including memory structures, reasoning systems, and behavioral tendencies based on predefined personas. Similarly, MetaAgents [26] implements initialization protocols that establish agent-specific goals, knowledge bases, and interaction capabilities before the simulation begins.

Additional initialization includes setting up parameters for evaluation metrics [136], establishing observation frameworks for data collection [109], and configuring environmental constraints such as communication protocols [143] or resource limitations [30]. In RecAgent [32], system parameters governing recommendation algorithms are also configured during this phase to ensure the simulation accurately reflects real-world recommendation dynamics.

4.3.2. Interaction

After initialization, the system moves into the interaction phase, which is the core of the simulation process, jointly managed by the environment and agents. In this phase, agents act either sequentially or in parallel, as coordinated by the agent manager.

The interaction begins with the perception module, where agents receive feedback and external stimuli from the environment. Park et al. [22] implement this using a retrieval-augmented generation approach where agents process environmental observations through memory modules before determining responses. Agents then process inputs through internal modules such as memory, planning, and action, which together determine each agent's specific response. CAMEL [143] demonstrates how LLM-based agents employ sophisticated planning mechanisms to translate perceptions into actionable decisions, while Voyager [104] shows how agents can develop increasingly complex interaction patterns through iterative learning from environmental feedback.

Notably, an agent may take multiple actions within a single interaction phase, allowing for more dynamic engagement. For example, in Generative Agents [22], agents can chain multiple micro-actions into coherent behavioral sequences when responding to complex scenarios. The agent manager actively monitors for and corrects abnormal behaviors, ensuring alignment with simulation objectives. As demonstrated in OASIS [30], this supervision framework can redirect agent behaviors when they become inconsistent with simulation goals or realistic human behavior patterns. Inter-agent communication represents a crucial element of the interaction phase. Zhou et al. [68] implement structured communication protocols that allow agents to exchange information, negotiate, and collaborate. Similarly, SwiftSage [146] shows how dynamically adapting communication patterns can emerge from repeated agent interactions under appropriate environmental conditions.

This intricate interaction mechanism fosters the emergence of complex, unpredictable behaviors, essential for studying social dynamics. For instance, Chen et al. [144] demonstrate how social norms can emerge organically from repeated agent interactions without explicit programming, providing insights into norm formation processes in human societies.

4.3.3. Update

After the interaction phase, the simulation enters the update phase, which focuses primarily on the evolution of the environment. As discussed in Section 4.1 and Section 3, agents impact the environment through their actions, and the environment itself may also undergo independent changes. During this phase, the environment update its states, establishing a new starting point for the next simulation cycle.

For example, in a financial market simulation presented by Li et al. [35], stock prices fluctuate based on agents' buying and selling activities according to market dynamics models. Wang et al. [32] implement a recommendation system simulation where content ranking algorithms update based on user interactions, creating a dynamic information environment that evolves. Additionally, if each agent performs only one action per interaction phase, this is also when agents update their internal states and memories. Park et al. [22] demonstrate how agents incorporate new experiences into structured memory representations, allowing past interactions to influence future decisions and enhancing simulation realism. Environmental updates may also include stochastic elements to simulate unpredictable real-world events. Xu et al. [136] incorporate randomized external events that agents must respond to, while Horton et al. [39] implement probabilistic state transitions in environmental conditions that represent factors beyond agent control.

It is important to note that the interaction and update phases repeat across multiple cycles throughout the simulation. The final state of each cycle provides the initial conditions for the next, facilitating a continuous, dynamic process that reflects the evolving social system.

4.3.4. Settlement

Once the simulation has executed a predefined number of rounds or reached a termination condition, it enters the settlement phase. During settlement, the system compiles and outputs simulation results, including memory streams and action logs of each agent. These records provide valuable insights into the dynamics of the simulated social system. Li et al. [143] demonstrate how agent conversation logs can reveal emergent communication patterns, while Chuang et al. [109] shows how tracking opinion evolution across agents can illuminate polarization dynamics in social networks.

4.4. Summary of This Chapter

This chapter provides a detailed introduction to the construction of a social simulation system, including the management of the environment and multi-agent systems, as well as the overall operation process of the system. The environmental section emphasizes its importance in social simulation, covering three aspects: feedback mechanisms, dynamic evolution, and social constraints. The multi-agent management section discusses strategies for agent configuration, scheduling, and monitoring. Finally, the operation process of the social simulation system is elaborated through four stages: initialization, interaction, update, and settlement. Together, these contents form the core framework of the social simulation system, providing a powerful tool for studying social dynamics.

5. The Evaluation of Social Simulation system

By evaluating the results of a social simulation system, researchers can identify its limitations and areas for improvement, enabling iterative refinement to enhance the reliability and effectiveness of the system. To achieve this, it is essential to design the appropriate evaluation metrics and methodologies. This section will systematically review existing evaluation metrics and methods from both the micro-level and macro-level perspectives as summarized in table.9.

5.1. Micro Evaluation

Micro evaluation focuses on assessing the performance of a single LLM agent within the social simulation system, which can be divided into two approaches: Approach 1 involves establishing behavioral criteria for agents, while Approach 2 focuses on modular evaluation. Below, we detail the specific methods for these two micro evaluation approaches.

Table 9: Evaluation of the LLM agent based social simulation system

	Category	**Subcategory**	**Representative Works**
Micro Evaluation	***Action Criteria Approach***	Rationality	[68], [22], [136], [39] [26], [32]
		Consistency	[22], [68]
		Accuracy	[35], [28], [132]
	Module Specific Approach	Memory	[147], [22], [104], [143]
		Planning	[22], [99], [104]
		Action	[39], [68]
Macro Evaluation	***Believability***	-	[27, 22, 28, 30] [68, 35, 32, 39] [148, 116, 68, 108]
	Scalability	-	[22, 32, 28, 30, 95] [149, 132, 131, 148]
	Efficiency	-	[131, 30, 149]
	Cost	-	[86, 131, 28, 26]
Evaluation Methods	***Subjective***	Human Based	[68, 22]
		LLM Based	[26, 32, 39]
	Objective	Regression	Eq.4, Eq.5
		Classification	[150]
		Text Similarity	[151], [152]

5.1.1. Action Criteria Micro Evaluation Approach

One of the most import goal of a social simulation system is to simulate believable human behavior using LLM agents. The micro evaluation primarily addresses whether agent actions are rational, consistent, and accurate. We will discusses the following criteria and the assessment methods.

- **Rationality** refers to whether an agent acts in a natural and realistic manner that would be expected of humans in similar situations. This criterion evaluates if agent behaviors appear plausible within their social context.

  In Sotopia [68], researchers employ a Believability (BEL) score ranging from 0 to 10 to evaluate how natural and realistic agent behaviors appear. Park et al. [22] assess rationality through behavioral coherence metrics that measure whether agent actions follow logical progressions based on contextual cues. Similarly, Xu et al. [136] evaluate agent rationality through "naturalness" scores that reflect how well agent behaviors align with human expectations.

  The evaluation of rationality typically involves both qualitative and quantitative approaches. Qualitatively, human annotators or LLM-based evaluators rate behaviors based on established rubrics. For example, Horton et al. [39] employ human evaluators to score agent interactions on a naturalness scale, while Li et al. [26] utilize GPT-4 as an evaluator to assess behavioral plausibility. Quantitatively, researchers like Wang et al. [32] measure rationality through decision consistency with predefined rational choice models.

- **Consistency** evaluates whether an agent's behavior aligns with its defined character traits (personality, preferences, values) and historical actions. This criterion ensures that agents maintain coherent identities throughout simulations.

  Consistency assessment employs both subjective and objective approaches. For subjective evaluation, Park et al. [22] use human annotators to rate how well agent behaviors align with their established personas, while Zhou et al. [68] employ LLM evaluators to assess profile-behavior alignment. For objective evaluation, researchers configure agent profiles with specific traits and transform evaluation into a classification problem. Several studies have developed specialized consistency metrics.

- **Accuracy** is more challenging than consistency, as it requires the agent to accurately replicate or predict human behavior as observed in the real world. However, this requires prior access to real-world data for comparison. The evaluation of accuracy is typically objective, primarily through comparison of the agent's behavior to the ground truth. Common evaluation metrics for accuracy include accuracy and F1 score. For example, in [35], agent behavior was compared to actual real-world data in a financial market simulation, where the F1 score was used to measure how closely agent decisions aligned with real-world results. Additionally, in cases where the agent generates text as part of its actions, the semantic similarity of the generated text may also be evaluated. This can be done using text-based metrics such as cosine similarity of embeddings, BLEU score, or BERT score, to measure how closely the generated text matches a reference or ground truth [28]. [132] also require that the agent's generated text aligns with real-world opinion. When the opinion is discrete, represented as support, opposition, and neutral, the evaluation metrics can be accuracy or the F1 score. If the opinion is continuous, metrics such as MSE or MAE are used.

5.1.2. Module-Specific Micro Evaluation Approach

Apart from action criteria, researchers evaluate specific agent modules to understand their contribution to overall system's performance.

- **Memory Module Evaluation** focuses on retrieval relevance and reflection quality. Gu et al. [147] evaluate memory retrieval using Shannon entropy to measure information content in retrieved memories, while Park et al. [22] assess memory relevance through cosine similarity between current contexts and retrieved memory embeddings. For reflection quality, Wang et al. [104] use human evaluators to rate the informativeness of agent reflections, while Li et al. [143] employ automated metrics like information entropy and unique n-gram ratios to measure the richness of reflected memories.

- **Planning Module Evaluation** assesses plan appropriateness, coherence, and effectiveness. Park et al. [22] evaluate planning quality through goal-achievement rates that measure how effectively plans lead to intended outcomes. Qian et al. [99] employ plan execution success rates, measuring the proportion of plans that are successfully completed without requiring revision. Wang et al. [104] evaluate planning efficiency by measuring the number of steps required to achieve goals.

- **Action Module Evaluation** examines how agents process information and make choices. Horton et al. [39] evaluate decision quality by comparing agent choices to optimal decisions in controlled scenarios. Zhou et al. [68] assess decision-making through counterfactual analysis, examining how agents might have acted given different information or contexts. It is worth noting that the action criteria evaluation focuses on behavioral outcomes, such as rationality, consistency, and accuracy. In contrast, the action-module evaluation aims at the decision-making processes and the functionality of the action module itself.

5.2. Macro Evaluation

Macro evaluation focuses on assessing the overall performance of a social simulation system composed primarily of the environment and multiple agents. Traditional sociological research faces challenges of limited efficiency and high costs, necessitating believable, efficient, scalable, and cost-effective social simulation systems. Therefore we summarize the target of macro evaluation into four key aspects: believability, efficiency, scalability, and cost-effectiveness.

5.2.1. Believability

A simulation system's believability depends on its ability to replicate real-world social phenomena. This criterion serves as a form of validation that the system can produce results that align with observed social dynamics. Current research has demonstrated believability through replicating several key phenomena.

Several systems have successfully modeled information spread patterns. Gao et al. [27] demonstrated cascading information spread through social networks. Park et al. [22] demonstrated emergent relationship formation where agents developed friendships based on shared interests, proximity, and interaction frequency. Zhou et al. [68] simulated the evolution of trust and relationship strength through repeated interactions. Horton et al. [39] modeled competitive behaviors in resource allocation scenarios, demonstrating how agents strategically cooperate or compete based on perceived advantage. Han et al. [116] simulated economic competition in duopoly markets, producing pricing strategies similar to those observed in real markets. Wang et al.

[32] demonstrated how recommendation systems can create self-reinforcing information bubbles through feedback loops between user preferences and content algorithms. Mou et al. [28] quantified filter bubble formation in simulated social media environments, showing patterns consistent with empirical studies. Yang et al. [30] replicated how individual agents adopt majority behaviors despite contradicting private information. Tang et al. [148] modeled labor market fluctuations through agent-based simulation, producing unemployment patterns that matched empirical data.

Researchers validate believability through various approaches, including comparison with real-world data [35], expert evaluation [68], and controlled experiments [108].

5.2.2. Efficiency

For simulations to provide practical value, they must operate significantly faster than real-world processes. Efficiency evaluation examines computational performance and simulation speed.

The efficiency of social simulation systems is primarily influenced by agent interactions and LLM inference time. As simulations progress, agents accumulate memories, causing inputs to language models to grow increasingly lengthy, which impacts reasoning speed—particularly for transformer-based models with quadratic complexity in sequence length. Researchers have proposed various optimization techniques to enhance efficiency. Yu et al. [131] implemented memory compression methods that reduce token count while preserving essential information. Yang et al. [30] employed hierarchical memory structures with selective retrieval to minimize inference overhead. Pan et al. [149] demonstrated batch processing techniques that significantly improve throughput for large-scale simulations.

5.2.3. Scalability

Scalability refers primarily to a system's capacity to support large numbers of agents while maintaining performance. Real-world social phenomena often involve thousands or millions of participants, necessitating corresponding scale in simulation systems.

The evolution of scalability in LLM-based social simulations has been rapid. Park et al. [22] demonstrated simulations with up to 25 agents, establishing baseline approaches for LLM-based agent architecture. Wang et al. [32] and Mou et al. [28] expanded capabilities to support approximately 1,000 agents through optimized architectures and simplified agent models. Yang et al. [30] and Pan et al. [149] have achieved simulations with up to one million agents, though with simplified scenarios and reduced per-agent complexity. Besides, Yang et al. [30] demonstrated that simulation accuracy for collective behaviors improves with increasing agent counts, highlighting the importance of scalability for valid social simulations. Current technical approaches to achieve scalability include distributed computing architectures [149, 132], agent simplification for specific scenarios [95], memory-efficient representations [131], and hierarchical simulation designs where full-complexity agents interact with simplified agents [148].

5.2.4. Cost

The economic feasibility of social simulations depends largely on the computational costs of LLM inference. Evaluating cost-effectiveness requires standardized metrics that enable meaningful comparisons across implementations.

Kaiya et al. [86] evaluate the monetary cost incurred by each agent per unit of simulation time, providing a measure of economic efficiency at the agent level. Yu et al. [131] measure the number of tokens processed by the same model performing equivalent tasks, offering a standardized comparison that remains stable despite fluctuations in commercial API pricing. Mou et al. [28] report the total expense of running complete simulation cycles, providing holistic cost assessments for planning large-scale studies. Li et al. [26] evaluate the relationship between computational investment and simulation fidelity, identifying optimal operating points for different research objectives.

5.3. Evaluation Methods

Evaluating social simulation systems presents significant challenges, requiring assessment at both agent and system levels. These evaluation methods can be categorized into subjective and objective approaches, each with distinct advantages and limitations.

5.3.1. Subjective Evaluation

Subjective evaluation proves particularly valuable when standardized benchmarks are unavailable or when designing appropriate quantitative metrics is challenging. This approach typically involves:

- **Human-based Assessment** Traditional subjective evaluation relies on human annotators who assess system performance based on predefined criteria. For example, Zhou et al. [68] recruited crowdworkers to evaluate agent behaviors on dimensions including believability, consistency, and goal achievement. Similarly, Park et al. [22] employed human evaluators to rate the naturalness and coherence of agent interactions using Likert scales. While these assessments provide intuitive and interpretable results with detailed justifications, they face limitations including high resource requirements, potential cultural biases, and challenges with reproducibility.

- **LLM-based Assessment** Recent advancements in LLMs have enabled their use as evaluators, partially addressing the limitations of human assessment. Li et al. [26] utilized GPT-4 to evaluate agent behaviors by providing evaluation criteria and simulation results as prompts. Similarly, Wang et al. [32] compared human and LLM-based evaluations of recommendation agents, finding strong correlation between the two. Horton et al. [39] demonstrated that LLM evaluators can provide consistent ratings across multiple trials when assessing collective intelligence tasks. While LLM-based evaluation reduces costs and improves reproducibility, potential hallucinations necessitate robustness checks.

5.3.2. Objective Evaluation

Objective evaluation utilizes quantitative metrics to measure simulation performance in a standardized manner. While comprehensive benchmarks for social simulation systems remain under development, researchers commonly adapt established metrics from machine learning:

Regression Metrics. For continuous variables (e.g., opinion strength, economic indicators), common metrics include:

- Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

  where y_i is the ground truth and $\hat{y}_i$ is the prediction.

- Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

Classification Metrics. For discrete variables (e.g., binary decisions, categorical choices), researchers utilize Accuracy and F1 Score[150].

Text Similarity Metrics. For evaluating textual outputs, researchers employ BERT score, BLEU, cosine similarity or Information Entropy.

- BERT Score: [151]uses contextual embeddings to evaluate semantic similarity between texts.
- BLEU: [152] evaluates text similarity based on n-gram overlap.
- Cosine Similarity: Measures vector similarity, calculated as $\text{CosineSimilarity} = \frac{A \cdot B}{\|A\|\|B\|}$ for vectors A and B.
- Information Entropy: Quantifies uncertainty in a system, defined as $H(X) = -\sum_i p(x_i)\log p(x_i)$. Wang et al. [32] employed entropy measures to evaluate information richness in agent memory systems.

6. The Application of LLM Agent Based Social Simulation System

The incorporation of LLM agent into social simulations offers a new direction for sociological research. We will explore existing studies on the applications of LLM agent based social simulations in sociology .

At the current stage, research on social simulations involving LLM-based autonomous agents can be categorized into four main focuses based on their objectives and outcomes: discovering social patterns, interpreting social phenomena, validating social theories, and forecasting policy outcomes. Each application aims to enhance our understanding of complex social dynamics through controlled simulation environments.

6.1. Discovering Social Patterns

Social patterns refer to recurring behaviors, interactions, or trends that emerge within social groups, such as the formation of communities, the spread of ideas, and the development of cultural norms. LLM-based agents provide a novel approach to study pattern emergence in controlled environments. De

Freitas et al. [153] demonstrated how multiple GPT-3.5-based agents could exhibit collective phenomena inherent to human societies, particularly the emergence of scale-free networks in social media-like environments. Their research revealed that token prior distributions in language models might influence network formation, and adjusting agent naming conventions could produce more realistic social network structures. Similarly, Thudium et al. [154] explored strategic capabilities of generative agents in competitive multi-agent games modeled after the television show "Survivor". Their framework, built upon GPT-4, evaluated how agents with different cognitive capabilities performed in competitive, situational environments. The research revealed that strategic agents display more diverse action choices and exploration behaviors, and agents with socially appropriate roles were more likely to succeed in these challenges, illustrating how social dynamics emerge from individual agent characteristics. Ghaffarzadegan et al. [155] studied norm propagation through a workplace attire simulation involving 20 employee agents choosing between green or blue shirts. Each agent received prompts containing background information, personal traits, previous day's choices, and group statistics before making decisions. This simple model demonstrated how social norms develop and propagate through observational learning and conformity pressure.

6.2. Interpreting Social Phenomena

Social phenomena encompass observable behaviors, events, or trends within societies, such as social movements, economic shifts, or collective decision-making. Interpreting these phenomena requires analyzing the motives, contexts, and external influences that drive collective behaviors.

Williams et al. [156] employed LLM-based agents to simulate epidemic responses, providing evidence that generative agents effectively mimicked real-world behaviors like self-isolation during illness and increased precautionary measures during case surges. Their agents exhibited multi-wave patterns similar to those observed during recent pandemics, demonstrating how individual decision-making processes aggregate into population-level outcomes. This research illustrates how agent-based simulations can interpret complex social responses to public health crises. Han et al. [116] investigated business competition and collusion dynamics using GPT-4-powered intelligent agents representing competing firms. Their Simulation with Autonomous Behavior Model (SABM) framework revealed that without communication, intelligent agents consistently achieved tacit collusion with prices converging above Bertrand equilibrium but below monopoly levels. When communication was permitted, agents achieved higher collusion levels approaching cartel pricing. The research demonstrated how communication enhances trust between firms, reduces price war likelihood, and provides insights into real-world market behaviors. Chen et al. [157] addressed information asymmetry challenges in cooperative AI systems by developing a programmable contract framework. Their approach enabled modular task composition and extended sequences of interdependent tasks in distributed agent workflows. The framework demonstrated how contractual mechanisms could mitigate information friction and coordinate agent behaviors, offering insights into institutional structures that facilitate cooperation in complex social systems.

6.3. Validating Social Theories

Social theories offer frameworks for explaining social behaviors and structures, such as theories of social stratification or cultural evolution. Social simulations with LLM-based agents provide scalable and controlled environments to test social theories across diverse conditions.

Zhao et al. [158] examined collective decision-making (CDM) in LLM-based multi-agent systems, analyzing 52 recent systems through the lens of social choice theory. Their investigation revealed limitations in current approaches, which predominantly rely on dictatorial and majority voting methods. The study demonstrated how social theories about collective decision-making could be tested and refined through agent-based simulations, identifying gaps between theoretical predictions and observed outcomes. Shao et al. [159] investigated how cognitive capabilities like Theory of Mind (ToM) influence cooperative behavior in multi-agent systems. Contrary to intuitive expectations, their research found that agents with higher ToM capabilities did not necessarily exhibit better cooperation than those with lower capabilities. They proposed a novel matching alliance mechanism that explicitly considers belief consistency and specialized capabilities when forming coalitions. This work validates and refines theories about the relationship between cognitive abilities and social cooperation. Chuang et al. [109] validated the "small-world" network theory using LLM-based agents in simulated social environments. By recreating networks with varying structures (from highly clustered to random) and observing communication patterns, the researchers tested Milgram's famous "six degrees of separation" hypothesis. The simulation confirmed that information could indeed traverse the network in a small number of steps, particularly in networks with the small-world property of high clustering and short average path lengths. The study extended the theory by identifying how different types of information spread differently based on content characteristics and urgency, demonstrating how simulations can both validate and extend existing social theories. Yulan-OneSim[160] replicates and quantitatively verifies Axelrod's cultural dissemination theory by simulating the formation of cultural regions through local agent interactions. Additionally, it demonstrates real-world alignment by modeling the Brazilian real estate market and comparing simulation outputs with empirical rental price data, showcasing the simulator's capacity to validate both theoretical and empirical social patterns. Recently, [161] proposes the PROSIM multi-agent simulation system to validate key social theories. It demonstrates that structural holes in network topology can amplify inequality during policy interventions. It also finds that perceived fairness fully mediates the effect of policy, challenging the assumption of direct causality. Additionally, it reveals a threshold effect in behavioral contagion—when network density exceeds 0.38, the effect disappears. Together, these findings support a four-layer framework linking policy, network structure, psychological mechanisms, and collective behavior.

6.4. Forecasting Policy Outcomes

LLM-based simulations enable the projection of future scenarios by simulating agent behaviors over time under varying hypothetical conditions. By generating data-driven predictions, researchers and policymakers can make more informed decisions and test the effects of various interventions before real-world implementation.

Ji et al. [142] developed the SRAP-Agent framework (Simulating and Optimizing Resource Allocation Policies using LLM-based Agents) to bridge the gap between theoretical economic models and real-world dynamics. Using public housing allocation as a case study, they conducted extensive policy simulation experiments with specific optimization objectives. Their approach demonstrated how LLM-based agents could accurately model complex human decisions in resource allocation scenarios, enabling policymakers to forecast the outcomes of different allocation strategies. Yang et al. [30] utilized large-scale simulations to predict information diffusion patterns and evaluate content moderation policies on social platforms. Their system incorporated thousands of agents with diverse demographic backgrounds and behavioral tendencies to forecast how different policies would affect discourse quality, user engagement, and platform health metrics. The research demonstrated how scaled simulations could provide actionable insights for platform governance decisions. Li et al. [35] implemented a policy-driven simulation modeling the impact of different taxation schemes on economic inequality. Their EconAgent framework simulated a virtual economy where LLM-powered agents represented consumers, producers, and workers with distinct preferences and decision-making processes. The simulation tested progressive, flat, and regressive taxation policies, revealing that while progressive taxation reduced inequality measures most effectively, it also produced complex secondary effects on production incentives and market dynamics. Recently, [96] studies the capabilities of LLM agents in simulating human prosocial cooperative behaviors. It systematically replicates the results of several classic experiments in the public goods game, and further explores the mechanisms for generating "unrestricted" behaviors such as cheating and collaboration in open environments like classrooms and parking lots. It demonstrates the potential of such systems in depicting complex human social interactions. Ultimately, it provides a feasible technical path and mechanism insights for the application of LLM-based social behavior simulation in policy design and evaluation. To sum up, these results provided policymakers with nuanced understanding of potential long-term outcomes beyond simple redistribution metrics, demonstrating how agent-based simulations can capture the complex adaptive responses that often undermine policy intentions in real-world implementations.

7. Challenges and Future Directions

Integrating LLM-based agents into social simulation marks a transformative shift from traditional approaches. By harnessing LLMs' advanced language comprehension, reasoning, and generative capacities, these agents can capture nuanced human behaviors and model complex social dynamics with unprecedented realism and flexibility. This paradigm allows simulations to reflect the richness and diversity of real-world societies, offering powerful tools for theoretical exploration, policy evaluation, and behavioral analysis. Despite these advancements, LLM-based agent modeling in social simulation remains in its nascent stages. Below, we synthesize key challenges and future directions from existing literature.

7.1. Dynamic Adaptation of Individual Agent

The dynamic adaptation of roles means agents are expected to assume context-specific roles that evolve in response to real-world events, yet current methods often rely on static, oversimplified, or biased role definitions with underdeveloped dynamic adjustment mechanisms. In the physical world, roles inherently evolve dynamically as events unfold, but simulation environments struggle to mirror this process, as initial role setups fail to capture the nuanced impacts of real-world dynamics. To address this, the research opportunity lies in integrating structured real-world data with LLM-driven profile generation to construct rich agent personas, preceded by deep research to form an in-depth understanding of role dynamics, and their contextual dependencies. Additionally, incorporating adaptive role management through continuous feedback loops, reinforcement learning from human feedback (RLHF), and agent-level theory-of-mind modeling enables simulated roles to evolve dynamically, ensuring alignment with the role transformations observed in the physical world.

7.2. Behavioral Coherence of Individual Agent

The inability of current LLM agents to maintain adequate long-term memory and behavioral coherence is also a significant challenge, as human behavior in the physical world is characterized by enduring memories and consistent, contextually linked actions over time. Existing agents often suffer from fragmented memory and inconsistent behaviors. To bridge this gap, the research opportunity lies in emulating human memory mechanisms through hierarchical memory systems, which could include short-term, long-term, and episodic memory layers to store and retrieve information dynamically. Considering that past experiences consistently shape present and future actions of humans, we can incorporate techniques like context-sensitive memory retrieval, selective forgetting, and self-reflective summarization to better maintain the continuity of actions and decisions.

7.3. Scalability of Multi-agent System

Scalability is another critical challenge. Large-scale agent populations are necessary for generating emergent social behaviors and capturing the complexity of real-world societies, while deploying LLM-based agents for all individuals is impractical. To address this, hybrid architectures that balance sophistication and efficiency may be a feasible way: pairing powerful LLMs with lighter models or rule-based systems for simpler interactions or low-activity agents, and leveraging aggregation techniques to group similar agents into coherent clusters, reducing redundant computations without sacrificing behavioral fidelity.

7.4. Inadequacy of Evaluation Metrics and Benchmark

The absence of a universally accepted evaluation framework and benchmarks is also an important challenge, given the inherent complexity of social systems. Considering that the real-world observational data represents just one sampling among countless plausible scenarios, treating data reconstruction as the observation is overly restrictive, while focusing solely on macro-level phenomena like herd behavior fails to capture the nuanced complexities of social dynamics. To address this, fostering interdisciplinary collaboration with social sciences to design multi-level evaluation metrics may be reasonable. Incorporating causal inference tools and interpretability methods can further help validate simulation outputs by uncovering underlying behavioral mechanisms, striking a balance between empirical rigor and the qualitative richness of social systems.

8. Conclusion

This paper presents a comprehensive survey of social simulation powered by LLM-based agents. First, it introduces a modular framework for constructing LLM agents, comprising five core components: profile, perception, memory, planning, and action. These modules enable agents to perform adaptive role-playing, perceive social contexts, learn continuously, plan by scenario, and make dynamic decisions. Each module is discussed with various implementation methods. Second, the paper reviews current LLM-based social simulation systems, covering their frameworks, evaluation strategies, and applications. Lastly, the paper discusses the representative applications, challenges, and future directions. LLM-based social simulation is promising for studying complex and interconnected social dynamics, driven by advances in the powerful reasoning and planning of LLM. Meanwhile, realizing this vision requires parallel efforts in ethics, safety, and alignment. A multidisciplinary approach is essential to ensure that LLM agents are fair, ethical, and accountable, ultimately enabling more insightful and trustworthy social simulations.

9. Declaration of generative AI and AI-assisted technologies in the writing process

In order to improve the language and readability of this work, the authors used ChatGPT and Deepseek during its preparation. The authors reviewed and edited the content after using these tools and take full responsibility for the final publication.

References

[1] F. Squazzoni, W. Jager, and B. Edmonds, "Social simulation in the social sciences: A brief overview," *Social Science Computer Review*, vol. 32, no. 3, pp. 279–294, 2014.

[2] Ö. Gürcan, "Llm-augmented agent-based modelling for social simulations: Challenges and opportunities," *HHAI 2024: Hybrid Human AI Systems for the Social Good*, pp. 134–144, 2024.

[3] T. W. Swan, "Economic growth and capital accumulation," *Economic record*, vol. 32, no. 2, pp. 334–361, 1956.

[4] R. M. Solow, "A contribution to the theory of economic growth," *The quarterly journal of economics*, vol. 70, no. 1, pp. 65–94, 1956.

[5] W. A. Lewis *et al.*, "Economic development with unlimited supplies of labour," 1954.

[6] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the national academy of sciences*, vol. 99, no. suppl_3, pp. 7280–7287, 2002.

[7] C. M. Macal and M. J. North, "Tutorial on agent-based modeling and simulation," in *Proceedings of the Winter Simulation Conference, 2005.* IEEE, 2005, pp. 14–pp.

[8] M. W. Macy and R. Willer, "From factors to actors: Computational sociology and agent-based modeling," *Annual review of sociology*, vol. 28, no. 1, pp. 143–166, 2002.

[9] S. F. Railsback and V. Grimm, *Agent-based and individual-based modeling: a practical introduction.* Princeton university press, 2019.

[10] C. Gao, X. Lan, N. Li, Y. Yuan, J. Ding, Z. Zhou, F. Xu, and Y. Li, "Large language models empowered agent-based modeling and simulation: A survey and perspectives," *Humanities and Social Sciences Communications*, vol. 11, no. 1, pp. 1–24, 2024.

[11] Q. Ma, X. Xue, D. Zhou, X. Yu, D. Liu, X. Zhang, Z. Zhao, Y. Shen, P. Ji, J. Li *et al.*, "Computational experiments meet large language model based agents: A survey and perspective," *arXiv preprint arXiv:2402.00262*, 2024.

[12] H. Jin, L. Huang, H. Cai, J. Yan, B. Li, and H. Chen, "From llms to llm-based agents for software engineering: A survey of current, challenges and future," *arXiv preprint arXiv:2408.02479*, 2024.

[13] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey," *arXiv preprint arXiv:2309.07864*, 2023.

[14] H. Kavak, J. J. Padilla, C. J. Lynch, and S. Y. Diallo, "Big data, agents, and machine learning: towards a data-driven agent-based modeling approach," in *Proceedings of the Annual Simulation Symposium*, 2018, pp. 1–12.

[15] S. Venkatramanan, B. Lewis, J. Chen, D. Higdon, A. Vullikanti, and M. Marathe, "Using data-driven agent-based models for forecasting emerging infectious diseases," *Epidemics*, vol. 22, pp. 43–49, 2018.

[16] H. Kavak, "A data-driven approach for modeling agents," 2019.

[17] N. Keller and X. Hu, "Towards data-driven simulation modeling for mobile agent-based systems," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 29, no. 1, pp. 1–26, 2019.

[18] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[19] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[20] C. Xie, C. Chen, F. Jia, Z. Ye, K. Shu, A. Bibi, Z. Hu, P. Torr, B. Ghanem, and G. Li, "Can large language model agents simulate human trust behaviors?" *arXiv preprint arXiv:2402.04559*, 2024.

[21] C. Chen, B. Yao, Y. Ye, D. Wang, and T. J.-J. Li, "Evaluating the llm agents for simulating humanoid behavior." The First Workshop on Human-Centered Evaluation and Auditing of Language . . . , 2024.

[22] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th annual acm symposium on user interface software and technology*, 2023, pp. 1–22.

[23] G. Dai, W. Zhang, J. Li, S. Yang, S. Rao, A. Caetano, M. Sra *et al.*, "Artificial leviathan: Exploring social evolution of llm agents through the lens of hobbesian social contract theory," *arXiv preprint arXiv:2406.14373*, 2024.

[24] G. Piatti, Z. Jin, M. Kleiman-Weiner, B. Schölkopf, M. Sachan, and R. Mihalcea, "Cooperate or collapse: Emergence of sustainability behaviors in a society of llm agents," *arXiv preprint arXiv:2404.16698*, 2024.

[25] C. Zhu, Y. Cheng, J. Zhang, Y. Qiu, S. Xia, and H. Zhu, "Generative organizational behavior simulation using large language model based autonomous agents: A holacracy perspective," *arXiv preprint arXiv:2408.11826*, 2024.

[26] Y. Li, Y. Zhang, and L. Sun, "Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents," *arXiv preprint arXiv:2310.06500*, 2023.

[27] C. Gao, X. Lan, Z. Lu, J. Mao, J. Piao, H. Wang, D. Jin, and Y. Li, "S^3: Social-network simulation system with large language model-empowered agents," *arXiv preprint arXiv:2307.14984*, 2023.

[28] X. Mou, Z. Wei, and X. Huang, "Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation," *arXiv preprint arXiv:2402.16333*, 2024.

[29] G. Rossetti, M. Stella, R. Cazabet, K. Abramski, E. Cau, S. Citraro, A. Failla, R. Improta, V. Morini, and V. Pansanella, "Y social: an llm-powered social media digital twin," *arXiv preprint arXiv:2408.00818*, 2024.

[30] Z. Yang, Z. Zhang, Z. Zheng, Y. Jiang, Z. Gan, Z. Wang, Z. Ling, M. Ma, B. Dong, P. Gupta *et al.*, "Oasis: Open agents social interaction simulations on one million agents," in *NeurIPS 2024 Workshop on Open-World Agents*.

[31] A. Surve, A. Rathod, M. Surana, G. Malpani, A. Shamraj, S. R. Sankepally, R. Jain, and S. S. Mehta, "Multiagent simulators for social networks," *arXiv preprint arXiv:2311.14712*, 2023.

[32] L. Wang, J. Zhang, H. Yang, Z. Chen, J. Tang, Z. Zhang, X. Chen, Y. Lin, R. Song, W. X. Zhao *et al.*, "User behavior simulation with large language model based agents," *arXiv preprint arXiv:2306.02552*, 2023.

[33] S. Cai, J. Zhang, K. Bao, C. Gao, and F. Feng, "Flow: A feedback loop framework for simultaneously enhancing recommendation and user agents," *arXiv preprint arXiv:2410.20027*, 2024.

[34] D. Ebrat and L. Rueda, "Lusifer: Llm-based user simulated feedback environment for online recommender systems," *arXiv preprint arXiv:2405.13362*, 2024.

[35] N. Li, C. Gao, M. Li, Y. Li, and Q. Liao, "Econagent: large language model-empowered agents for simulating macroeconomic activities," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 15 523–15 536.

[36] C. Zhang, X. Liu, M. Jin, Z. Zhang, L. Li, Z. Wang, W. Hua, D. Shu, S. Zhu, X. Jin *et al.*, "When ai meets finance (stockagent): Large language model-based stock trading in simulated real-world environments," *arXiv preprint arXiv:2407.18957*, 2024.

[37] S. Gao, Y. Wen, M. Zhu, J. Wei, Y. Cheng, Q. Zhang, and S. Shang, "Simulating financial market via large language model based agents," *arXiv preprint arXiv:2406.19966*, 2024.

[38] G. Chen, Z. Zhang, X. Cong, F. Guo, Y. Wu, Y. Lin, W. Feng, and Y. Wang, "Learning evolving tools for large language models," *arXiv preprint arXiv:2410.06617*, 2024.

[39] J. J. Horton, "Large language models as simulated economic agents: What can we learn from homo silicus?" National Bureau of Economic Research, Tech. Rep., 2023.

[40] Z. R. Baker and Z. L. Azher, "Simulating the us senate: An llm-driven agent approach to modeling legislative behavior and bipartisanship," *arXiv preprint arXiv:2406.18702*, 2024.

[41] F. Moghimifar, Y.-F. Li, R. Thomson, and G. Haffari, "Modelling political coalition negotiations using llm-based agents," *arXiv preprint arXiv:2402.11712*, 2024.

[42] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[43] Z. Zhang, X. Bo, C. Ma, R. Li, X. Chen, Q. Dai, J. Zhu, Z. Dong, and J.-R. Wen, "A survey on the memory mechanism of large language model based agents," *arXiv preprint arXiv:2404.13501*, 2024.

[44] X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen, "Understanding the planning of llm agents: A survey," *arXiv preprint arXiv:2402.02716*, 2024.

[45] F. He, T. Zhu, D. Ye, B. Liu, W. Zhou, and P. S. Yu, "The emerged security and privacy of llm agent: A survey with case studies," *arXiv preprint arXiv:2407.19354*, 2024.

[46] Y. Li, H. Wen, W. Wang, X. Li, Y. Yuan, G. Liu, J. Liu, W. Xu, X. Wang, Y. Sun *et al.*, "Personal llm agents: Insights and survey about the capability, efficiency and security," *arXiv preprint arXiv:2401.05459*, 2024.

[47] Z. Deng, Y. Guo, C. Han, W. Ma, J. Xiong, S. Wen, and Y. Xiang, "Ai agents under threat: A survey of key security challenges and future pathways," *arXiv preprint arXiv:2406.02630*, 2024.

[48] X. Li, S. Wang, S. Zeng, Y. Wu, and Y. Yang, "A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges," *Vicinagearth*, vol. 1, no. 1, p. 9, 2024.

[49] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao *et al.*, "Exploring large language model based intelligent agents: Definitions, methods, and prospects," *arXiv preprint arXiv:2401.03428*, 2024.

[50] H. Guo, C. Mu, Y. Chen, C. Shen, S. Hu, and Z. Wang, "Multi-agent, human-agent and beyond: A survey on cooperation in social dilemmas," *arXiv preprint arXiv:2402.17270*, 2024.

[51] D. Hillier, C. Tan, and J. Jiang, "Social learning through interactions with other agents: A survey," *CoRR*, 2024.

[52] H. Ding, Y. Li, J. Wang, and H. Chen, "Large language model agent in financial trading: A survey," *arXiv preprint arXiv:2408.06361*, 2024.

[53] M. C. Ramos, C. J. Collison, and A. D. White, "A review of large language models and autonomous agents in chemistry," *arXiv preprint arXiv:2407.01603*, 2024.

[54] X. Xu, Y. Wang, C. Xu, Z. Ding, J. Jiang, Z. Ding, and B. F. Karlsson, "A survey on game playing agents and large models: Methods, applications, and challenges," *arXiv preprint arXiv:2403.10249*, 2024.

[55] S. Hu, T. Huang, F. Ilhan, S. Tekin, G. Liu, R. Kompella, and L. Liu, "A survey on large language model-based game agents," *arXiv preprint arXiv:2404.02039*, 2024.

[56] Z. Guo and X. Liu, "How artificial intelligence cooperating with agent-based modeling for urban studies: A systematic review," *Transactions in GIS*, vol. 28, no. 3, pp. 654–674, 2024.

[57] S. Gao, A. Fang, Y. Huang, V. Giunchiglia, A. Noori, J. R. Schwarz, Y. Ektefaie, J. Kondic, and M. Zitnik, "Empowering biomedical discovery with ai agents," *Cell*, vol. 187, no. 22, pp. 6125–6151, 2024.

[58] J. Liu, K. Wang, Y. Chen, X. Peng, Z. Chen, L. Zhang, and Y. Lou, "Large language model-based agents for software engineering: A survey," *arXiv preprint arXiv:2409.02977*, 2024.

[59] H. Jin, L. Huang, H. Cai, J. Yan, B. Li, and H. Chen, "From llms to llm-based agents for software engineering: A survey of current, challenges and future," *arXiv preprint arXiv:2408.02479*, 2024.

[60] J. Gao, H. Qiao, and Y. Zhang, "Intelligent recognition of tool wear with artificial intelligence agent," *Coatings*, vol. 14, no. 7, p. 827, 2024.

[61] X.-y. Chen and L. Zhang, "Revolutionizing bridge operation and maintenance with llm-based agents: An overview of applications and insights," *arXiv e-prints*, pp. arXiv–2407, 2024.

[62] X. Mou, X. Ding, Q. He, L. Wang, J. Liang, X. Zhang, L. Sun, J. Lin, J. Zhou, X. Huang, and Z. Wei, "From individual to society: A survey on social simulation driven by large language model-based agents," 2024. [Online]. Available: https://arxiv.org/abs/2412.03563

[63] D. Fuentes, J. Oden, K. Diller, J. Hazle, A. Elliott, A. Shetty, and R. Stafford, "Computational modeling and real-time control of patient-specific laser treatment of cancer," *Annals of biomedical engineering*, vol. 37, pp. 763–782, 2009.

[64] J. M. Epstein, *Generative social science: Studies in agent-based computational modeling.* Princeton University Press, 2012.

[65] A. Eyni, M. J. E. Skardi, and R. Kerachian, "A regret-based behavioral model for shared water resources management: Application of the correlated equilibrium concept," *Science of the Total Environment*, vol. 759, p. 143892, 2021.

[66] H. Chan, M. T. Irfan, and C. V. Than, "Schelling models with localized social influence: a game-theoretic framework," in *AAMAS Conference proceedings*, 2020.

[67] R. Axelrod, "The dissemination of culture: A model with local convergence and global polarization," *Journal of conflict resolution*, vol. 41, no. 2, pp. 203–226, 1997.

[68] X. Zhou, H. Zhu, L. Mathur, R. Zhang, H. Yu, Z. Qi, L.-P. Morency, Y. Bisk, D. Fried, G. Neubig *et al.*, "Sotopia: Interactive evaluation for social intelligence in language agents," *arXiv preprint arXiv:2310.11667*, 2023.

[69] Y. Nakajima, "Babyagi," https://github.com/yoheinakajima/babyagi.

[70] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon *et al.*, "Constitutional ai: Harmlessness from ai feedback," *arXiv preprint arXiv:2212.08073*, 2022.

[71] L. P. Argyle, E. C. Busby, N. Fulda, J. R. Gubler, C. Rytting, and D. Wingate, "Out of one, many: Using language models to simulate human samples," *Political Analysis*, vol. 31, no. 3, pp. 337–351, 2023.

[72] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[73] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[74] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[75] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[76] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[77] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[78] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large

language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[79] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *Advances in Neural Information Processing Systems*, vol. 36, pp. 68 539–68 551, 2023.

[80] W. Zhong, L. Guo, Q. Gao, H. Ye, and Y. Wang, "Memorybank: Enhancing large language models with long-term memory," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 724–19 731.

[81] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 8634–8652, 2023.

[82] Significant Gravitas, "AutoGPT." [Online]. Available: https://github.com/Significant-Gravitas/AutoGPT

[83] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[84] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.

[85] S. Qiao, Y. Ou, N. Zhang, X. Chen, Y. Yao, S. Deng, C. Tan, F. Huang, and H. Chen, "Reasoning with language model prompting: A survey," *arXiv preprint arXiv:2212.09597*, 2022.

[86] Z. Kaiya, M. Naim, J. Kondic, M. Cortes, J. Ge, S. Luo, G. R. Yang, and A. Ahn, "Lyfe agents: Generative agents for low-cost real-time social interactions," *arXiv preprint arXiv:2310.02172*, 2023.

[87] Z. Wang, Y. Y. Chiu, and Y. C. Chiu, "Humanoid agents: Platform for simulating human-like generative agents," *arXiv preprint arXiv:2310.05418*, 2023.

[88] Y. Lan, Z. Hu, L. Wang, Y. Wang, D. Ye, P. Zhao, E.-P. Lim, H. Xiong, and H. Wang, "Llm-based agent society investigation: Collaboration and confrontation in avalon gameplay," *arXiv preprint arXiv:2310.14985*, 2023.

[89] Z. Zhang, D. Zhang-Li, J. Yu, L. Gong, J. Zhou, Z. Hao, J. Jiang, J. Cao, H. Liu, Z. Liu *et al.*, "Simulating classroom education with llm-empowered agents," *arXiv preprint arXiv:2406.19226*, 2024.

[90] Y. Yu, Z. Yao, H. Li, Z. Deng, Y. Jiang, Y. Cao, Z. Chen, J. Suchow, Z. Cui, R. Liu *et al.*, "Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making," *Advances in Neural Information Processing Systems*, vol. 37, pp. 137 010–137 045, 2024.

[91] Y. Liu, X. Chen, X. Zhang, X. Gao, J. Zhang, and R. Yan, "From skepticism to acceptance: Simulating the attitude dynamics toward fake news," *arXiv preprint arXiv:2403.09498*, 2024.

[92] X. Li, Y. Xu, Y. Zhang, and E. C. Malthouse, "Large language model-driven multi-agent simulation for news diffusion under different network structures," *arXiv preprint arXiv:2410.13909*, 2024.

[93] X. Zhang, J. Lin, X. Mou, S. Yang, X. Liu, L. Sun, H. Lyu, Y. Yang, W. Qi, Y. Chen *et al.*, "Socioverse: A world model for social simulation powered by llm agents and a pool of 10 million real-world users," *arXiv preprint arXiv:2504.10157*, 2025.

[94] H. Hattori, A. Kato, and M. Yoshizoe, "Integrating large language models into agent models for multi-agent simulations: Preliminary report," in *2024 Winter Simulation Conference (WSC)*. IEEE, 2024, pp. 230–241.

[95] A. Zhang, Y. Chen, L. Sheng, X. Wang, and T.-S. Chua, "On generative agents in recommendation," in *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, 2024, pp. 1807–1817.

[96] K. Sreedhar, A. Cai, J. Ma, J. V. Nickerson, and L. B. Chilton, "Simulating cooperative prosocial behavior with multi-agent llms: Evidence

and mechanisms for ai agents to inform policy decisions," in *Proceedings of the 30th International Conference on Intelligent User Interfaces*, 2025, pp. 1272–1286.

[97] W. Zheng and X. Tang, "Simulating social network with llm agents: An analysis of information propagation and echo chambers," in *International Symposium on Knowledge and Systems Sciences*. Springer, 2024, pp. 63–77.

[98] L. Zhang, Y. Hu, W. Li, Q. Bai, and P. Nand, "Llm-aidsim: Llm-enhanced agent-based influence diffusion simulation in social networks," *Systems*, vol. 13, no. 1, p. 29, 2025.

[99] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong *et al.*, "Chatdev: Communicative agents for software development," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 15 174–15 186.

[100] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, C.-M. Chan, H. Yu, Y. Lu, Y.-H. Hung, C. Qian, Y. Qin, X. Cong, R. Xie, Z. Liu, M. Sun, and J. Zhou, "Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=EHg5GDnyq1

[101] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang, "Autogen: Enabling next-gen LLM applications via multi-agent conversation," in *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024. [Online]. Available: https://openreview.net/forum?id=uAjxFFing2

[102] X. Zhou, Z. Su, T. Eisape, H. Kim, and M. Sap, "Is this the real life? is this just fantasy? the misleading success of simulating social interactions with llms," *arXiv preprint arXiv:2403.05020*, 2024.

[103] T. Sumers, S. Yao, K. Narasimhan, and T. Griffiths, "Cognitive architectures for language agents," *Transactions on Machine*

Learning Research, 2024, survey Certification. [Online]. Available: https://openreview.net/forum?id=1i6ZCvflQJ

[104] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=ehfRiF0R3a

[105] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "PaLM-e: An embodied multimodal language model," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 8469–8488. [Online]. Available: https://proceedings.mlr.press/v202/driess23a.html

[106] M. Sap, R. Le Bras, D. Fried, and Y. Choi, "Neural theory-of-mind? on the limits of social intelligence in large LMs," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3762–3780. [Online]. Available: https://aclanthology.org/2022.emnlp-main.248/

[107] Q. Zhao, J. Wang, Y. Zhang, Y. Jin, K. Zhu, H. Chen, and X. Xie, "Competeai: Understanding the competition dynamics of large language model-based agents," in *Forty-first International Conference on Machine Learning*.

[108] H. Zhang, J. Yin, M. Jiang, and C. Su, "Can agents spontaneously form a society? introducing a novel architecture for generative multi-agents to elicit social emergence," *arXiv preprint arXiv:2409.06750*, 2024.

[109] Y.-S. Chuang, A. Goyal, N. Harlalka, S. Suresh, R. Hawkins, S. Yang, D. Shah, J. Hu, and T. T. Rogers, "Simulating opinion dynamics with networks of llm-based agents," *arXiv preprint arXiv:2311.09618*, 2023.

[110] R. C. Atkinson, "Human memory: A proposed system and its control processes," *The psychology of learning and motivation*, vol. 2, 1968.

[111] Y. Liu, Z. Song, X. Zhang, X. Chen, and R. Yan, "From a tiny slip to a giant leap: An llm-based simulation for fake news evolution," *arXiv preprint arXiv:2410.19064*, 2024.

[112] A. Georgiou, M. Katkov, and M. Tsodyks, "Retroactive interference model of forgetting," *The Journal of Mathematical Neuroscience*, vol. 11, pp. 1–15, 2021.

[113] G. D. Brown and S. Lewandowsky, "Forgetting in memory models: Arguments against trace decay and consolidation failure," in *Forgetting*. Psychology Press, 2010, pp. 63–90.

[114] J. T. Wixted and E. B. Ebbesen, "Genuine power curves in forgetting: A quantitative analysis of individual subject forgetting functions," *Memory & cognition*, vol. 25, pp. 731–739, 1997.

[115] R. B. Anderson and R. D. Tweney, "Artifactual power curves in forgetting," *Memory & Cognition*, vol. 25, pp. 724–730, 1997.

[116] X. Han, Z. Wu, and C. Xiao, """ guinea pig trials" utilizing gpt: A novel smart agent-based modeling approach for studying firm competition and collusion," *arXiv preprint arXiv:2308.10974*, 2023.

[117] R. Liu, R. Yang, C. Jia, G. Zhang, D. Yang, and S. Vosoughi, "Training socially aligned language models on simulated social interactions," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=NddKiWtdUm

[118] W. Hua, L. Fan, L. Li, K. Mei, J. Ji, Y. Ge, L. Hemphill, and Y. Zhang, "War and peace (waragent): Large language model-based multi-agent simulation of world wars," *arXiv preprint arXiv:2311.17227*, 2023.

[119] J. S. Park, L. Popowski, C. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Social simulacra: Creating populated prototypes for social computing systems," in *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 2022, pp. 1–18.

[120] H. Kim, J. Hessel, L. Jiang, P. West, X. Lu, Y. Yu, P. Zhou, R. Bras, M. Alikhani, G. Kim *et al.*, "Soda: Million-scale dialogue distillation with social commonsense contextualization," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 12 930–12 949.

[121] Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang *et al.*, "Lora: Low-rank adaptation of large language models."

[122] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. [Online]. Available: https://aclanthology.org/2021.emnlp-main.243/

[123] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning.* PMLR, 2019, pp. 2790–2799.

[124] Y. Zhou, L. Song, B. Wang, and W. Chen, "Metagpt: Merging large language models using model exclusive task arithmetic," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 1711–1724.

[125] C. Zhang, K. Yang, S. Hu, Z. Wang, G. Li, Y. Sun, C. Zhang, Z. Zhang, A. Liu, S.-C. Zhu *et al.*, "Proagent: building proactive cooperative agents with large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 591–17 599.

[126] J. Yang, A. Prabhakar, K. Narasimhan, and S. Yao, "Intercode: Standardizing and benchmarking interactive coding with execution feedback," *Advances in Neural Information Processing Systems*, vol. 36, pp. 23 826–23 854, 2023.

[127] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought rea-

soning by large language models," *arXiv preprint arXiv:2305.04091*, 2023.

[128] Y. Liang, C. Wu, T. Song, W. Wu, Y. Xia, Y. Liu, Y. Ou, S. Lu, L. Ji, S. Mao *et al.*, "Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis," *Intelligent Computing*, vol. 3, p. 0063, 2024.

[129] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *Forty-first International Conference on Machine Learning*, 2023.

[130] S. Abdelnabi, A. Gomaa, S. Sivaprasad, L. Schönherr, and M. Fritz, "Cooperation, competition, and maliciousness: Llm-stakeholders interactive negotiation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 83 548–83 599, 2024.

[131] Y. Yu, Q. Zhang, J. Li, Q. Fu, and D. Ye, "Affordable generative agents," *arXiv preprint arXiv:2402.02053*, 2024.

[132] J. Piao, Y. Yan, J. Zhang, N. Li, J. Yan, X. Lan, Z. Lu, Z. Zheng, J. Y. Wang, D. Zhou *et al.*, "Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society," *arXiv preprint arXiv:2502.08691*, 2025.

[133] S. Du and X. Zhang, "Helmsman of the masses? evaluate the opinion leadership of large language models in the werewolf game," *arXiv preprint arXiv:2404.01602*, 2024.

[134] R. Dong, Z. Liao, G. Lai, Y. Ma, D. Ma, and C. Fan, "Who is undercover? guiding llms to explore multi-perspective team tactic in the game," *arXiv preprint arXiv:2410.15311*, 2024.

[135] T. Sato, S. Ozaki, and D. Yokoyama, "An implementation of werewolf agent that does not truly trust llms," *arXiv preprint arXiv:2409.01575*, 2024.

[136] Y. Xu, S. Wang, P. Li, F. Luo, X. Wang, W. Liu, and Y. Liu, "Exploring large language models for communication games: An empirical study on werewolf," *arXiv preprint arXiv:2309.04658*, 2023.

[137] H. Qiu and Z. Lan, "Interactive agents: Simulating counselor-client psychological counseling via role-playing llm-to-llm interactions," *arXiv preprint arXiv:2408.15787*, 2024.

[138] J. S. Park, C. Q. Zou, A. Shaw, B. M. Hill, C. Cai, M. R. Morris, R. Willer, P. Liang, and M. S. Bernstein, "Generative agent simulations of 1,000 people," *arXiv preprint arXiv:2411.10109*, 2024.

[139] S. Jiang, L. Wei, and C. Zhang, "Donald trumps in the virtual polls: Simulating and predicting public opinions in surveys using large language models," *arXiv preprint arXiv:2411.01582*, 2024.

[140] S. Guo, H. Wang, H. Bu, Y. Ren, D. Sui, Y.-M. Shang, and S. Lu, "Large language models as rational players in competitive economics games," 2023.

[141] S. Jinxin, Z. Jiabao, W. Yilei, W. Xingjiao, L. Jiawen, and H. Liang, "Cgmi: Configurable general multi-agent interaction framework," *arXiv preprint arXiv:2308.12503*, 2023.

[142] J. Ji, Y. Li, H. Liu, Z. Du, Z. Wei, W. Shen, Q. Qi, and Y. Lin, "Srap-agent: Simulating and optimizing scarce resource allocation policy with llm-based agent," *arXiv preprint arXiv:2410.14152*, 2024.

[143] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "Camel: Communicative agents for" mind" exploration of large language model society," *Advances in Neural Information Processing Systems*, vol. 36, pp. 51 991–52 008, 2023.

[144] Y. Chen, T. X. Liu, Y. Shan, and S. Zhong, "The emergence of economic rationality of gpt," *Proceedings of the National Academy of Sciences*, vol. 120, no. 51, p. e2316205120, 2023.

[145] S. Ren, Z. Cui, R. Song, Z. Wang, and S. Hu, "Emergence of social norms in generative agent societies: principles and architecture," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024, pp. 7895–7903.

[146] B. Y. Lin, Y. Fu, K. Yang, F. Brahman, S. Huang, C. Bhagavatula, P. Ammanabrolu, Y. Choi, and X. Ren, "Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks," *Advances*

in Neural Information Processing Systems, vol. 36, pp. 23 813–23 825, 2023.

[147] Z. Gu, X. Zhu, H. Guo, L. Zhang, Y. Cai, H. Shen, J. Chen, Z. Ye, Y. Dai, Y. Gao *et al.*, "Agent group chat: An interactive group chat simulacra for better eliciting collective emergent behavior," *arXiv preprint arXiv:2403.13433*, 2024.

[148] J. Tang, H. Gao, X. Pan, L. Wang, H. Tan, D. Gao, Y. Chen, X. Chen, Y. Lin, Y. Li *et al.*, "Gensim: A general social simulation platform with large language model based agents," *arXiv preprint arXiv:2410.04360*, 2024.

[149] X. Pan, D. Gao, Y. Xie, Z. Wei, Y. Li, B. Ding, J.-R. Wen, and J. Zhou, "Very large-scale multi-agent simulation in agentscope," *arXiv preprint arXiv:2407.17789*, 2024.

[150] Z.-H. Zhou, *Machine learning.* Springer nature, 2021.

[151] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[152] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[153] G. De Marzo, L. Pietronero, and D. Garcia, "Emergence of scale-free networks in social interactions among large language models," *arXiv preprint arXiv:2312.06619*, 2023.

[154] S. Thudium, F. Cimini, R. Vyas, K. Sullivan, L. Petro, A. Zhu, and C. Callison-Burch, "Outwit, outplay, out-generate: A framework for designing strategic generative agents in competitive environments."

[155] N. Ghaffarzadegan, A. Majumdar, R. Williams, and N. Hosseinichimeh, "Generative agent-based modeling: Unveiling social system dynamics through coupling mechanistic models with generative artificial intelligence," *arXiv preprint arXiv:2309.11456*, 2023.

[156] R. Williams, N. Hosseinichimeh, A. Majumdar, and N. Ghaffarzadegan, "Epidemic modeling with generative agents," *arXiv preprint arXiv:2307.04986*, 2023.

[157] C. Chen and L. Parker, "Composable contracts for multi-agent coordination," in *Agentic Markets Workshop at ICML 2024*.

[158] X. Zhao, K. Wang, and W. Peng, "An electoral approach to diversify llm-based multi-agent collective decision-making," *arXiv preprint arXiv:2410.15168*, 2024.

[159] J. Shao, T. Yuan, T. Lin, X. Cao, and B. Luo, "Cognitive insights and stable coalition matching for fostering multi-agent cooperation," *arXiv preprint arXiv:2405.18044*, 2024.

[160] L. Wang, H. Gao, X. Bo, X. Chen, and J.-R. Wen, "Yulan-onesim: Towards the next generation of social simulator with large language models," *arXiv preprint arXiv:2505.07581*, 2025.

[161] Y. Zhou, H. Wang, Q. Ai, Z. Wu, and Y. Liu, "Simulating prosocial behavior and social contagion in llm agents under institutional interventions," *arXiv preprint arXiv:2505.15857*, 2025.