EcoLANG: Efficient and Effective Agent Communication Language Induction for Social Simulation

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated an impressive ability to role-play humans and replicate complex social dynamics. While large-scale social simulations are gaining increasing attention, they still face significant challenges, particularly regarding high time and computation costs. Existing solutions, such as distributed mechanisms or hybrid agent-based model (ABM) integrations, either fail to address inference costs or compromise accuracy and generalizability. To this end, we propose **EcoLANG**: Efficient and Effective Agent Communication Language Induction for Social Simulation. EcoLANG operates in two stages: (1) language evolution, where we filter synonymous words and optimize sentence-level rules through natural selection, and (2) language utilization, where agents in social simulations communicate using the evolved language. Experimental results demonstrate that EcoLANG reduces token consumption by over 20%, enhancing efficiency without sacrificing accuracy.

1 Introduction

Social simulation has emerged as a powerful methodology for understanding the complex societal systems (Squazzoni et al., 2014). It explores the dynamics and emergent behaviors of societal systems by modeling the interactions between individuals, which is previously implemented by agent-based models (ABMs) (Bianchi and Squazzoni, 2015). However, traditional ABMs often rely on oversimplified agent behaviors, overlooking the context-dependent nature of human decision-making. Recent advancements in large language models (LLMs) have opened new possibilities for social simulation by enabling agents to exhibit more human-like behaviors (Mou et al., 2024a). The LLM-driven agents can vividly role-play specific persons (Argyle et al., 2023; Park et al., 2024), collaborate to complete tasks (Hong et al., 2023; Qian et al., 2024) and interact to replicate real-world phenomenas (Mou et al., 2024b; Li et al., 2024; Zhang et al., 2024a).

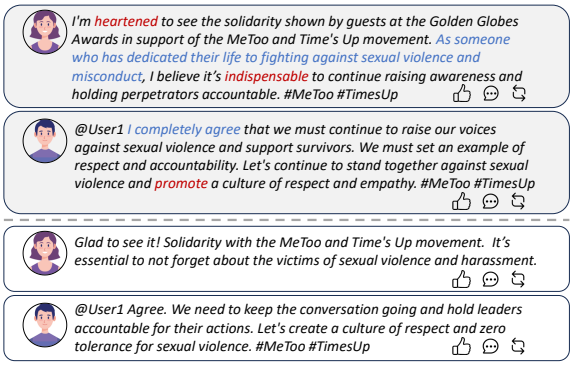

Figure 1: Responses generated by LLM-driven agents (top) and those generated by the same agents using more efficient expression (bottom) when discussing the Metoo movement. There is information redundancy in the vanilla setting, such as long but unnecessary sentences (in blue) and advanced but uncommon words (in red).

Due to the vast number of individuals in society, large-scale social simulations are becoming increasingly important in practical applications. Although LLMs have demonstrated potential to replicate human behaviors, conducting large-scale simulations remains challenging. The sheer number of agents and their interactions result in excessively high time and computational costs for the simulations (Gao et al., 2024). Currently, efforts to address this issue primarily fall into two categories: (1) Some works have improved simulation efficiency through distributed mechanisms (Pan et al., 2024; Yang et al., 2024), but they **have not fundamentally addressed the issue of inference costs**. (2) Other efforts have attempted to propose efficient simulation frameworks, such as integrating with ABM models (Chopra et al., 2024; Mou et al., 2024b) or reusing certain strategies (Yu et al., 2024), which **simplify modeling for some agents but may compromise simulation accuracy and lack gener-**

alizability across different scenarios. To better understand the inefficiencies in current simulations, we analyze the communication patterns of LLM-driven agents. From Figure 1, we observe that there is **communication redundancy** in current LLM-driven multi-agent social interactions. Agents tend to use fancy vocabulary and longer, complex sentence structures, leading to token wastage. In contrast, the principle of least effort (Zipf, 2016) show that humans tend to achieve effective communication with minimal effort, which includes using simple, common, and easy-to-understand words to reduce cognitive load, as well as employing concise sentences to shorten communication length and save time. Similarly, for LLM-driven agents, adopting common words and reducing vocabulary size can decrease GPU memory usage during simulation, and reducing the number of inference tokens can lower computational costs.

Inspired by this, we aim to develop an agent language designed for large-scale social simulations, enabling agents to communicate efficiently. We introduce **EcoLANG**: Efficient and Effective Agent Communication Language Induction for Social Simulation. EcoLANG operates in two stages: language evolution and language utilization. First, inspired by the principle of least effort (Zipf, 2016), we filter synonymous words based on word frequency and length to create a new vocabulary, thereby reducing the size of the vocabulary of existing LLMs. Then, through a natural selection paradigm, we prompt agents to use different rules for communication in dialogue-intensive scenarios, iteratively optimizing the rules. This ultimately evolves efficient sentence-level rules, i.e., "grammar" for the new language. In the language utilization stage, we urge the agents in large-scale social simulations to communicate using the acquired language by modifying the inference model's vocabulary and incorporating rule-based prompts. Since this language is induced through communication and does not rely on any task-specific architecture, it is framework-agnostic, allowing it to adapt seamlessly to various scenarios.

We conduct extensive experiments on the open-sourced Llama-3.1-8B-Instruct (Dubey et al., 2024). We evolve the language on twitter corpus and the synthetic-persona-chat dataset (Jandaghi et al., 2023), and we validate the effectiveness of this language in social simulations on PHEME (Zubiaga et al., 2016) and Metoo and Roe datasets of HiSim (Mou et al., 2024b). The experiment results show that EcoLANG can significantly reduce token consumption and improve the efficiency of social simulations without compromising simulation accuracy, demonstrating advantages over baseline methods such as structured languages and traditional agent communication languages like KQML (Finin et al., 1994). Overall, the contributions of this paper can be summarized as follows:

- We introduce EcoLANG, a two-stage paradigm consisting of language evolution and language utilization. EcoLANG can induce efficient and effective language for LLM-driven social simulations.
- We derive an agent language using EcoLANG on twitter corpus and synthetic-persona-chat dataset, that can directly generalize to different downstream social simulation scenarios.
- We conduct extensive experiments on different scenarios. The results demonstrate that EcoLANG can reduce inference costs while keeping the simulation accuracy.

2 Related Work

2.1 LLM-driven Social Simulation

Recently, LLMs have been used to construct agents to empower social simulations, aiming to reveal and explain emergent behaviors and the outcomes of interactions among numerous agents (Mou et al., 2024a). In such simulations, each agent role-plays a person in society and participates in social interactions, with the goal of modeling complex phenomena such as opinion dynamics (Chuang et al., 2024), epidemic modeling (Williams et al., 2023), and macroeconomic activities (Li et al., 2024). Initial researches construct virtual spaces supporting such simulations (Park et al., 2022, 2023). Further studies focus on alignment on specific scenarios, validating whether real-world behaviors and phenomena can be replicated by such simulations (Gao et al., 2023; Liu et al., 2024). Although LLMs show potential in mimicking human, their integration into large-scale simulation remains challenging. Some work has improved simulation efficiency by deploying open-source models-driven agents through distributed mechanisms (Pan et al., 2024; Yang et al., 2024), but it has not addressed the fundamental issue of computational costs and communication efficiency. Other work seeks to combine with agent-based models (ABMs), simplifying the

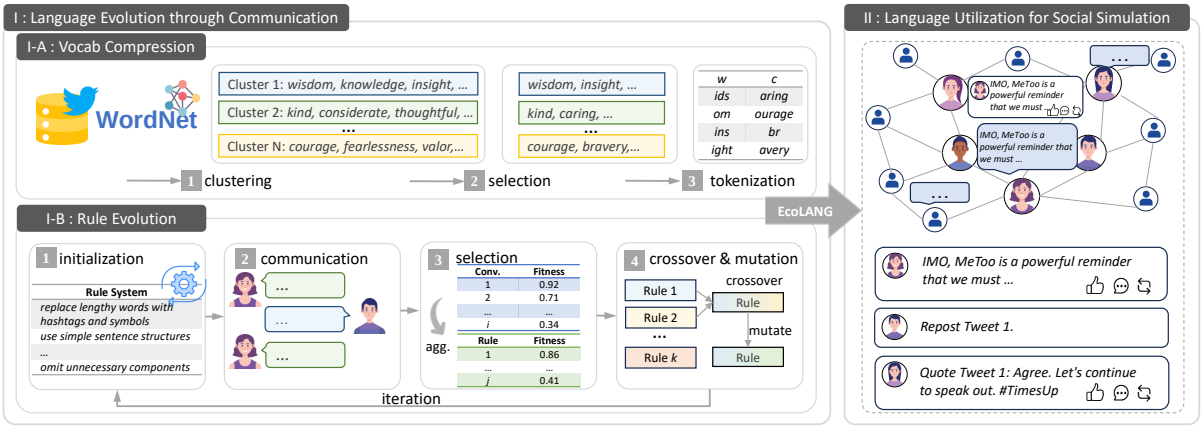

Figure 2: Overview of the EcoLANG framework. We get the language through vocabulary compression and rule evolution in dialogue-intensive scenarios. Then, we enable agents to use this language in social simulations.

modeling of certain agents, which may sacrifice some simulation effectiveness (Mou et al., 2024b; Chopra et al., 2024).

2.2 Multi-Agent Communication

Before the rise of LLMs, some studies focused on how multi-agent systems could use language to cooperate in completing tasks or solving problems (Havrylov and Titov, 2017; Lazaridou and Baroni, 2020; Lazaridou et al., 2020), typically developing effective communication protocols with task success as a training signal. In current LLM-driven multi-agent systems, communication is mainly conducted through natural language. Some research has highlighted the redundancy in communication, leading to suggestions that agents autonomously choose structured languages like JSON for communication (Chen et al., 2024a; Marro et al., 2024) or further fine-tune models to improve this communication (Chen et al., 2024b). Meanwhile, other studies have approached communication optimization from the perspective of its structure, aiming to enhance efficiency by pruning the spatial-temporal message graph (Zhang et al., 2024b). However, most existing work focuses on task-solving rather than social simulation, which more urgently needs to address the challenges of large-scale simulation.

3 Methodology

3.1 Overview

To address the issues of cost and efficiency in social simulation and to reduce the generation of unnecessary content, we propose a two-stage paradigm EcoLANG. First, we reconstruct the vocabulary based on the principle of least effort (Sec 3.2) and evolve a set of rules through natural selection (Sec 3.3). After obtaining this new language, we apply it to social simulation, encouraging agents to use this language for communication (Sec 3.4). The overall process is shown in Figure 2.

3.2 Vocabulary Compression

The development of a new language begins with defining its basic elements, i.e., the vocabulary. Since LLMs are primarily trained on natural languages, our goal is not to introduce an entirely novel symbolic language or completely replace the existing vocabulary. Instead, we focus on compressing the current vocabulary while preserving its foundational structure. In natural languages, many words share similar meanings, such as synonyms, which allows for potential optimization. This idea aligns with the principle of least effort, also known as Zipf's Law (Zipf, 2016), which suggests that people naturally tend to use the minimal effort required to communicate effectively. This principle is evident in the frequency of word usage, where low-frequency words are often replaced by more common synonyms (Mohammad, 2020). Drawing inspiration from this phenomenon, we propose a method to compress the vocabulary of LLMs, as outlined in the following steps and illustrated in part I-A of Figure 2.

Semantic Clustering To ensure that the language can still support the expression of various semantics, the new vocabulary should encompass words covering a wide range of meanings. To achieve this, we begin by clustering all words in a given corpus according to their semantic similarities, followed by further filtering within each semantic cluster.

Specifically, instead of performing clustering from scratch, we leverage existing synsets from WordNet (Miller, 1995) and assign each word in the corpus to the most relevant synset based on embedding similarity. For each word w, we compute the similarity between the word embedding e_w and the center embedding of each synset e_{S_j}, and assign w_i to the synset with the highest similarity:

$$S(w_i) = \arg\max_j \left(\text{sim}\left(\mathbf{e}_{w_i}, \mathbf{e}_{S_j}\right)\right), \quad (1)$$

This approach not only enhances controllability but also minimizes noise in the clustering process.

Intra-Cluster Selection Within each cluster, we further filter words by assigning a score to each word based on two key factors: word frequency and word length. On the one hand, as previously mentioned, more frequently used words tend to efficiently convey the speaker's intent and are likely to be better trained due to their higher occurrence. On the other hand, we prioritize retaining shorter words to minimize the length of generated content. With these considerations, we define the following scoring function:

$$R(w_i) = \lambda_{freq} F(w_i) + \lambda_{token}(1 - L(w_i)), \quad (2)$$

where $F(w_i)$ and $L(w_i)$ represent the percentile scores of the word's frequency and token lengths respectively. λ_{freq} and λ_{token} are hyperparameters. Based on these scores, we retain the top words within each cluster according to a predefined retention ratio r_w.

Tokenization While people use words as the basic units of communication, LLMs process text in units of tokens. Therefore, after identifying the words to retain, we tokenize them to determine which tokens should be kept. Although these tokens may form additional words beyond our initial selection, the overall vocabulary size of the LLMs is still effectively reduced through our method. Furthermore, we ensure that special tokens, which are essential for the model's correct generation, are preserved throughout this process.

3.3 Language Rule Evolution

Once vocabulary, the fundamental elements of a language are identified, another core aspect is how these elements are organized, i.e., grammar, or the rule system. Previous research in linguistics (Pinker and Bloom, 1990; Nowak and Krakauer, 1999) has suggested that grammar is a simplified system of rules that has evolved through natural selection, with the purpose of reducing errors in communication. Inspired by this, we design the language using the principles of evolutionary algorithms (EAs). The task is formulated as identifying a rule or prompt to enable agents to communicate both effectively and efficiently. The process primarily involves the following steps, which are also shown in part I-B of Figure 2.

Initialization Evolution typically begins with an initial population of N solutions, i.e., rule prompts $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$, which are then iteratively refined to generate new solutions. To initialize the rule system, we employ a combination of manually crafted prompts and those generated by LLMs, to leverage the wisdom of humans and LLMs (Guo et al.). These prompts are designed to instruct agents to express concisely, where the details can be found in Appendix A.2.1.

Communication Language is used and evolves through communication in social interactions. To observe how individuals using specific rules communicate, we simulate dialogues between LLM-driven agents. Given a set of dialogue scenarios $\mathcal{D}$ between two agents, for each scenario $d_i \in \mathcal{D}$, we generate M dialogue trajectories $\left\{\tau_i^j\right\}_{j=1}^{M}$, each with a randomly selected rule from $\mathcal{P}$ appended to the original prompt to the agents.

Selection To select high-quality rules, we evaluate the dialogue trajectories using a fitness function. First, the language should be both effective and efficient. *Efficiency* can be measured by token count. For effectiveness, previous work in multi-agent task-solving often uses task success rate as a metric (Lazaridou et al., 2020). However, in social simulation, there is no specific task. Instead, we believe that *alignment*—how well the agent embodies the persona it is role-playing, is the cornerstone of social simulation. Thus, we include an alignment score to indicate effectiveness. Additionally, *expressiveness* (Galke et al., 2022) is crucial to prevent the language from becoming overly abstract and to maintain fluency. Taking these factors into account, we define the following fitness function:

$$\begin{aligned} F(\tau_i^j) = \lambda_{align} Align(\tau_i^j) + \lambda_{eff} Eff(\tau_i^j) \\ + \lambda_{exp} Exp(\tau_i^j), \end{aligned} \quad (3)$$

where the alignment score $Align(\tau_i^j)$ and the

expressiveness score $Exp(\tau_i^j)$ are given by external judge LLMs, and $Eff(\tau_i^j)$ is the normalized token count $\frac{\#\,\text{Tokens}(\tau_i^j)}{\max_k(\{\#\,\text{Tokens}(\tau_i^k)\}_k)}$. λ_{align}, λ_{eff} and λ_{exp} are hyperparameters. After calculating the fitness score for each dialogue trajectory, we aggregate and average these scores based on the rules used, which allows us to derive the overall fitness score for each rule.

Crossover and Mutation To diversify the rules, we perform crossover and mutation operations on the high-quality rules following the selection process. Specifically, we select parent rules from the population based on their fitness values, which determine the probability of selection. Then, we prompt LLMs to generate new rules using the prompts from (Guo et al.).

Update and Iteration In each iteration, we use the elitism strategy of genetic algorithm to update the population. We retain the top $N/2$ rules from the current population based on fitness values, and generate another $N/2$ rules through crossover and mutation, ensuring the population size remains constant at N. In applications, the best rule is used as the rule for the new language. The overall process can be described as Algorithm 1.

3.4 Language Utilization in Social Simulation

Once we acquire the vocabulary and rule system of a new language, we enable agents to communicate in that language by modifying the decoding range of the LLMs that drive them and incorporating rules into their original contextual prompts. While the evolution and use of a language could intuitively occur within the same scenario, we adopt a **transfer setting** for two key reasons: (1) the sparsity of large-scale social simulation data, and (2) the natural emergence of new languages from everyday communication. Specifically, we evolve the language on general multi-turn dialogue data, where communication is more intensive, to enhance the efficiency of language evolution. We then apply the evolved language to specific social interaction scenarios. Moreover, since the language is evolved on general communication data, this design is inherently task-agnostic.

4 Experiment Settings

As mentioned before, we evolve and utilize language in different scenarios. We filter vocabulary in Twitter corpus and acquire the rule in dialogue-intensive scenarios and apply the evolved language in social simulation scenarios, i.e., PHEME (Zubiaga et al., 2016) and Metoo and Roe of HiSim (Mou et al., 2024b). PHEME aims to simulate the propagation and discussion of potential rumors, while HiSim focuses on modeling the evolution of opinion dynamics following the release of triggering news related to specific social movements.

4.1 Language Evolution Settings

Twitter Corpus for Vocabulary Compression Since we partly rely on word frequency to filter words in Sec 3.2, we need a corpus to count words. While it would be ideal to include all tweets, this is not feasible. Therefore, we have chosen to analyze and gather statistics from existing tweets relevant to the topics of our social simulation scenarios. Specifically, for the PHEME, which aims to model rumor, we use tweets from Twitter15 (Liu et al., 2015) and Twitter16 (Ma et al., 2016), resulting in 35,211 words from 41,736 tweets. For HiSim, we use tweets related to the corresponding movements (Maiorana et al., 2020; Chang et al., 2023; Mou et al., 2024b), resulting in 1,662,657 words from 52,967,084 tweets.

Scenarios for Communication in Rule Evolution During the rule evolution process, we use the synthetic-persona-chat dataset (Jandaghi et al., 2023) to generate conversations between agents following specific rules. This dataset provides a collection of dialogues between two users with diverse personalities, along with their corresponding personality descriptions. We provide these profiles to LLMs, instructing them to role-play these individuals communicating, and obtain dialogue trajectories for further selection.

Implementation Details The agents are powered by Llama-3.1-8B-Instruct (Dubey et al., 2024). For vocabulary compression, we set the hyperparameters $\lambda_{freq} = 1$, $\lambda_{token} = 1$. The reservation ratio r_w for each semantic cluster is 0.6 for PHEME and 0.2 for HiSim, yielding vocabulary sizes of 32.6K (25.4% of Llama-3.1's vocabulary) and 48.2K (37.5% of Llama-3.1's vocabulary), respectively. For rule evolution, we set the number of initial rules $N = 10$. We use the development set of the synthetic-persona-chat dataset, which contains 1,000 chatting scenarios for communication simulation during the evolution process. For selection, GPT-4o (Achiam et al., 2023) serves as

Method	PHEME						HiSim						
	stance↑	belief↑	belief_JS↓	$token_r$↓	$token_p$↓	$token_c$↓	stance↑	content↑	$\Delta bias$ ↓	Δdiv ↓	$token_r$↓	$token_p$↓	$token_c$↓
Base	66.21	42.44	0.137	2.61K	84.43K	8.44K	70.30	30.23	0.093	0.027	13.02K	1.92M	283.79K
Summary	66.07	41.55	0.133	2.41K	84.27K	8.01K	**70.95**	32.31	0.089	0.023	10.62K	1.90M	269.73K
AutoForm	63.72	40.50	0.136	2.00K	85.02K	7.69K	69.92	32.04	**0.082**	0.029	10.66K	1.89M	252.09K
KQML	57.66	42.09	0.130	3.01K	91.10K	9.18K	70.16	32.47	0.093	0.032	12.06K	1.96M	279.17K
AgentTorch	-	-	-	-	-	-	67.87	31.81	0.098	0.024	**2.5K**	**0.48M**	**94.34K**
Vocab	65.73	44.65	0.131	2.67K	84.78K	8.70K	70.34	30.48	0.086	0.023	11.37K	1.91M	286.41K
Rule	**66.86**	45.14	0.128	**1.98K**	**82.08K**	**7.52K**	70.63	32.25	0.091	0.027	9.07K	1.84M	242.43K
EcoLANG	66.34	**45.50**	**0.128**	2.08K	82.26K	7.70K	70.60	**32.57**	0.083	**0.020**	9.80K	1.83M	236.83K

Table 1: Experimental results of different methods. The average results of 3 runs are reported. We report the best performance in **bold** format and the second best in underlined format.

the judge to evaluate alignment and expressiveness based on reference dialogues. The weight hyperparameters are set to $\lambda_{align} = 1$, $\lambda_{eff} = 0.6$ and $\lambda_{exp} = 0.6$. In each iteration, We retain the top-5 parent rules and generate 5 new rules through crossover and mutation, with parents randomly selected according to their scores. The number of iterations is set to 5. Please refer to Appendix A for more details.

4.2 Language Utilization Settings

Datasets We collect 196 real-world instances from PHEME (Zubiaga et al., 2016), each involving 2 to 31 users discussing a source tweet, to examine whether agents can mimic user responses towards rumors. We use the second events of #Metoo and #Roeoverturned movements from HiSim (Mou et al., 2024a), each with 1,000 users discussing the topic-related news over time, to study the opinion dynamics following social interactions.

Metrics For PHEME, we focus on content-related metrics. We measure consistency between each agent's initial stance on the source tweet and real users' stances, categorized into four types as in (Derczynski et al., 2017) and annotated by GPT-4o-mini. Following (Liu et al., 2024), we also label each agent's final belief as belief, disbelief, or unknown using GPT-4o-mini, and compute belief consistency and JS divergence (Lin, 1991) of belief distribution with real-world data.

For HiSim, we follow (Mou et al., 2024b) to report stance and content consistency between initial agent responses and real users' initial posts, labeled by GPT-4o-mini. We also report $\Delta bias$ and Δdiv. to measure the difference in average opinion bias and diversity between simulated and real user groups over time.

For both datasets, we evaluate communication efficiency by reporting the average number of tokens in generated tweet responses per scenario (# $tokens_r$), as well as the total token consumption per scenario, which includes both prompt tokens (# $tokens_p$) and completion tokens (# $tokens_c$).

Baselines We have the following baselines for comparison, including different means of communication: (1) *Base*: conduct simulations without adding any additional rule prompt; (2) *Summary*: prompt agents to summarize their opinions when responding, as concise expression resembles a summarization task; (3) *AutoForm* (Chen et al., 2024a): prompt agents to automatically choose a structured format to respond, such as JSON and logical expression; (4) *KQML* (Finin et al., 1994): prompt agents to use a traditional agent communication language KQML; (5) *Vocab*: a variant of our method that only compresses the vocabulary of the LLMs; (6) *Rule*: a variant of our method that only applies the evolved communication rules. We also include an efficient hybrid simulation method that does not focus on communication optimization: (7) *AgentTorch* (Chopra et al., 2024): uses LLM archetypes to represent all the agents, simulate the actions of archetypes and map their response to other agents.

Implementation Details Agents are driven by Llama-3.1-8B-Instruct (Dubey et al., 2024). All the simulations are conducted in OASIS framework (Yang et al., 2024). We run each simulation 3 times and report the averaged results. We use GPT-4o-mini to label the stance, belief and content of responses and apply Textblob to calculate the opinion score. Details can be found in Appendix B.

5 Experiment Results

5.1 Overall Performance

The overall results are presented in Table 1. We have the following observations.

(1) Can reducing communication redundancy improve simulation efficiency? Compared to *Base*, all methods of simplified communication

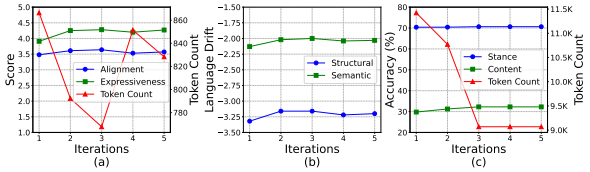

Figure 3: (a) Average fitness score change and (b) language drift change on synthetic-persona-chat simulated dialogues across iterations; (c) Performance and token consumption in HiSim using the best language rules acquired across iterations.

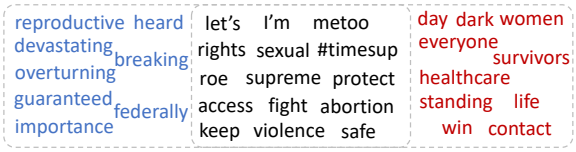

Figure 4: Top words in responses generated by agents in HiSim simulation without (left) and with (right) vocabulary compression. The center part presents their intersection.

PHEME				HiSim			
Ratio	# Vocab	stance↑	belief↑	Ratio	# Vocab	stance↑	content↑
0.2	31.5K	63.80	44.16	0.2	48.2K	70.34	30.48
0.4	31.8K	63.13	43.57	0.4	49.3K	70.41	30.09
0.6	32.6K	64.10	44.25	0.6	50.9K	69.64	29.11
0.8	34.0K	65.73	44.65	0.8	52.8K	69.26	29.55
Llama-3.1	128.3K	66.21	42.44	Llama-3.1	128.3K	70.30	30.23

Table 2: Performance of the simulations when using different vocabularies. *Ratio* represents the reserving ratio for each semantic cluster when filtering words. We at least keep one word for each synonym set.

have significantly reduced token generation. This improvement in efficiency is not only reflected in $token_r$ but also cumulatively transmitted to $token_p$ and $token_c$ due to the generated content serving as context for other agents, agents' memory mechanisms, and so on. Among these, our proposed *Rule* and *EcoLANG* are the most prominent, capable of reducing generated tokens by over 20%. However, we have also observed that approaches like *AgentTorch*, which modify the simulation paradigm rather than simplifying communication, can more significantly reduce token consumption, albeit often at the cost of reduced simulation accuracy.

(2) Will simplifying communication compromise the effectiveness of the simulation? Some baselines such as *AutoForm* and *KQML*, despite enhancing efficiency, reduced the accuracy of the simulation. This may suggest that while these structured languages can improve the efficiency and effectiveness of task-solving, they might not be suitable for social simulation, as humans generally communicate using natural language. By contrast, benefiting from the considerations of both efficiency and alignment during the process of language evolution, our method is able to enhance efficiency while maintaining leading simulation accuracy.

(3) Does vocabulary compression enhance performance or efficiency? We observed that simulations can still achieve comparable, or even better, results after vocabulary compression, e.g., in HiSim, indicating that the vocabulary in LLMs may be redundant for these simulations. Theoretically, removing these tokens in LLM's vocabulary could enhance the model's inference efficiency, using less GPU memory. However, vocabulary compression does not have a significant impact on token consumption. This is not surprising since the change in the length of individual words has a minimal effect on the overall sentence length.

5.2 Analysis of Rule Evolution

To explore the evolution process of language rules, we analyze the changes in dialogue scores and language shifts during the iterations, as well as the impact of the acquired rules on downstream social simulations. Figure 3(a) and (b) illustrate the changes in metrics on the synthetic-person-chat dialogues during the evolution. In addition to the fitness scores defined in Sec 3.3, we have also calculated the dialogues' structural drift and semantic drift (Lazaridou et al., 2020), where structural drift measures the fluency and grammaticality in relation to natural language, while semantic drift measures the adequacy in relation to the literal semantics of the target. The results indicate that as iterative evolution progresses, the fitness of the language initially shows an upward trend, with alignment and expressiveness increasing while token consumption decreases. The decline in language drift further corroborates the improvement in language quality, even though we did not directly optimize these metrics during the evolution process. This improvement is also reflected in the impact of the acquired rules on downstream social simulations, where simulation accuracy improves and token consumption is reduced, as shown in Figure 3(c). However, after a certain number of iterations, the fitness score no longer improves, potentially indicating some form of overfitting, and the optimal rules provided for the simulation tasks no longer change.

Method	PHEME						HiSim						
	stance↑	belief↑	belief_JS↓	$token_r$↓	$token_p$↓	$token_c$↓	stance↑	content↑	$\Delta bias$ ↓	Δdiv ↓	$token_r$↓	$token_p$↓	$token_c$↓
Qwen	63.35	49.25	0.1426	1.93K	78.21K	7.36K	71.63	26.06	0.1025	0.0246	17.68K	1.81M	214.11K
Qwen w/ Rule	62.95	51.65	0.1475	1.81K	78.36K	7.09K	72.04	26.77	0.0978	0.0255	14.71K	1.77M	188.96K
Mistral	62.98	52.39	0.1529	3.10K	96.94K	11.60K	72.02	31.78	0.1220	0.0536	26.91K	2.36M	416.15K
Mistral w/ Rule	63.84	60.00	0.1484	2.28K	94.76K	10.39K	72.39	32.57	0.0963	0.0352	22.76K	2.29M	358.23K

Table 3: Results of simulations driven by Qwen2.5 and Mistral with and without the evolved rule of Llama3.1.

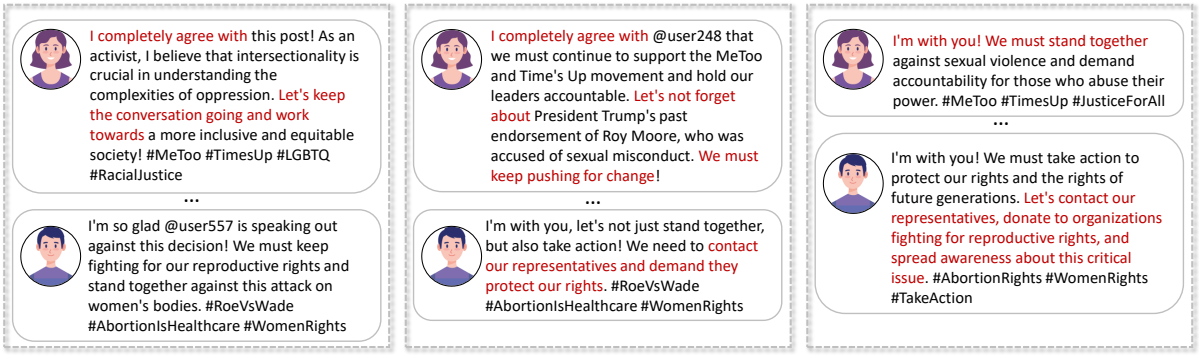

Figure 5: Case study: responses of agents without any communication optimization and with the best evolved rule at iteration 1 and 5. In most cases, agents express more concisely while sometimes fail to follow instructions.

5.3 Analysis of Vocab Size

We further explore the impact of the vocabulary on the simulation. As shown in Table 2, since it is necessary to ensure that at least one word is retained for each semantic cluster, changing the retention ratio has subtle impact on the size of the vocabulary. Nevertheless, it can be observed that the influence of vocabulary size on performance exhibits different trends across simulations. For PHEME, a larger vocabulary is better, possibly because it covers a more diverse range of topics and discussions, requiring more words for support. In contrast, for HiSim, due to the more focused discussion topics Metoo and Roe, using fewer but more commonly used words can achieve ideal results.

We also compare the top 50 frequent words generated by the agents with and without vocabulary compression. Figure 4 shows that after vocabulary compression, agents have indeed reduced the use of cumbersome words like "devastating", opting instead for simpler and more commonly used terms such as "dark" and "day", while still retaining the usage of other common words.

5.4 Transferability of Language Rule Across Different LLMs

Can the evolved language be used on other models, or do we need to reacquire the language for each model? To answer this question, we applied the acquired language rules to other models, i.e., Qwen2.5-7b-Chat (Team, 2024) and Mistral-7b-Instruct-v0.3 (Jiang et al., 2023). Table 3 show that the rules evolved on Llama-3.1 can also enable other models to communicate efficiently, again demonstrating the transferability of EcoLANG.

5.5 Case Study and Error Analysis

Figure 5 showcases some exemplary instances of efficient communication and bad cases. Benefiting from the evolved rule, agents can speak more concisely using words like "I'm with you" to replace "I completely agree with you". However, sometimes the agents may fail to simplify their expression and disclose excessive details. This may be the result of the model's insufficient ability to follow instructions. A potential solution is to further fine-tune the models using the efficient communication dialogues from the language evolution process.

6 Conclusion

We introduced EcoLANG, a novel two-stage paradigm comprising language evolution and utilization, designed to acquire efficient and effective language for large-scale social simulations. We derive the language by vocabulary compression and rule evolution and demonstrate its applicability across social simulation scenarios. Experiment results highlight EcoLANG's ability to reduce inference costs while maintaining simulation accuracy.

Limitations

EcoLANG induces an efficient agent communication language that improves simulation efficiency and reduces inference costs while maintaining simulation accuracy. Despite our careful design, some limitations still exist.

- Although EcoLANG improves efficiency, the extent of this improvement is not yet transformative. This is because we focus on reducing token generation but do not address the reduction of the number of inference times. In the future, we plan to integrate it with hybrid frameworks that optimize the number of inference steps, thereby further enhancing efficiency and reducing costs to a greater extent.

- Due to the limited available large-scale social simulation datasets for validation, we have currently only tested EcoLANG in PHEME and HiSim, which may raise concerns about its generalizability. In the future, it will be necessary to advance the construction of benchmarks for diverse social simulation scenarios.

- Due to the lack of objective and unified evaluation frameworks and metrics for existing LLM-driven social simulations, as compared to task-solving scenarios, we currently partly rely on LLMs to get the fitness value during the selection process, which may introduce potential bias. We will continue to explore more reliable evaluation frameworks for social simulation.

Ethics Statement

This paper aims to evolve an efficient communication language for social simulation. Like most work in social simulation, it may raise potential considerations and we urge the readers to approach it with caution.

- When employing LLMs for social simulation, concerns arise regarding the fidelity and interpretability of the results. If not carefully managed, the risk of bias could exacerbate real-world problems. However, our experiments demonstrate that EcoLANG does not amplify incorrect predictions related to misinformation (PHEME) or opinion polarization (HiSim).

- Ensuring the ethical handling of any real-world datasets, including anonymization and consent, is crucial. During our social simulations, all user content was anonymized to minimize privacy risks.

- Although EcoLANG is designed to evolve efficient language, misuse, such as promoting uncivil language, could pose risks. Therefore, strict governance and ethical guidelines should be implemented.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. 2023. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351.

Federico Bianchi and Flaminio Squazzoni. 2015. Agent-based models in sociology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(4):284–306.

Rong-Ching Chang, Ashwin Rao, Qiankun Zhong, Magdalena Wojcieszak, and Kristina Lerman. 2023. # roeoverturned: Twitter dataset on the abortion rights controversy. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 17, pages 997–1005.

Weize Chen, Chenfei Yuan, Jiarui Yuan, Yusheng Su, Chen Qian, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2024a. Beyond natural language: LLMs leveraging alternative formats for enhanced reasoning and communication. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10626–10641, Miami, Florida, USA. Association for Computational Linguistics.

Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2024b. Optima: Optimizing effectiveness and efficiency for llm-based multi-agent system. *arXiv preprint arXiv:2410.08115*.

Ayush Chopra, Shashank Kumar, Nurullah Giray-Kuru, Ramesh Raskar, and Arnau Quera-Bofarull. 2024. On the limits of agency in agent-based models. *arXiv preprint arXiv:2409.10568*.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy Rogers. 2024. Simulating opinion dynamics with networks of llm-based agents. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3326–3346.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. 1994. Kqml as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*, pages 456–463.

Lukas Galke, Yoav Ram, and Limor Raviv. 2022. Emergent communication for understanding human language evolution: What's missing? *arXiv preprint arXiv:2204.10590*.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S^3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30.

Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Pegah Jandaghi, XiangHai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2023. Faithful persona-based conversational dataset generation with large language models. *arXiv preprint arXiv:2312.10007*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Angeliki Lazaridou and Marco Baroni. 2020. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. 2020. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7663–7674.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. 2024. 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)*, pages 15523–15536.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1867–1870.

Yuhan Liu, Xiuying Chen, Xiaoqing Zhang, Xing Gao, Ji Zhang, and Rui Yan. 2024. From skepticism to acceptance: simulating the attitude dynamics toward fake news. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Zachary Maiorana, Pablo Morales Henry, and Jennifer Weintraub. 2020. #metoo Digital Media Collection - Twitter Dataset.

Samuele Marro, Emanuele La Malfa, Jesse Wright, Guohao Li, Nigel Shadbolt, Michael Wooldridge, and Philip Torr. 2024. A scalable communication protocol for networks of large language models. *arXiv preprint arXiv:2410.11905*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Saif Mohammad. 2020. Wordwars: A dataset to examine the natural selection of words. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3087–3095.

Xinyi Mou, Xuanwen Ding, Qi He, Liang Wang, Jingcong Liang, Xinnong Zhang, Libo Sun, Jiayu Lin, Jie Zhou, Xuanjing Huang, et al. 2024a. From individual to society: A survey on social simulation driven by large language model-based agents. *arXiv preprint arXiv:2412.03563*.

Xinyi Mou, Zhongyu Wei, and Xuanjing Huang. 2024b. Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4789–4809, Bangkok, Thailand. Association for Computational Linguistics.

Martin A Nowak and David C Krakauer. 1999. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033.

Xuchen Pan, Dawei Gao, Yuexiang Xie, Yushuo Chen, Zhewei Wei, Yaliang Li, Bolin Ding, Ji-Rong Wen, and Jingren Zhou. 2024. Very large-scale multi-agent simulation in agentscope. *arXiv preprint arXiv:2407.17789*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2022. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–18.

Joon Sung Park, Carolyn Q Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S Bernstein. 2024. Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109*.

Steven Pinker and Paul Bloom. 1990. Natural language and natural selection. *Behavioral and brain sciences*, 13(4):707–727.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. 2024. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186.

Flaminio Squazzoni, Wander Jager, and Bruce Edmonds. 2014. Social simulation in the social sciences: A brief overview. *Social Science Computer Review*, 32(3):279–294.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Ross Williams, Niyousha Hosseinichimeh, Aritra Majumdar, and Navid Ghaffarzadegan. 2023. Epidemic modeling with generative agents. *arXiv preprint arXiv:2307.04986*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Martin Ma, Bowen Dong, Prateek Gupta, et al. 2024. Oasis: Open agents social interaction simulations on one million agents. In *NeurIPS 2024 Workshop on Open-World Agents*.

Yangbin Yu, Qin Zhang, Junyou Li, Qiang Fu, and Deheng Ye. 2024. Affordable generative agents. *arXiv preprint arXiv:2402.02053*.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024b. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

George Kingsley Zipf. 2016. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio books.

Arkaitz Zubiaga, Geraldine Wong Sak Hoi, Maria Liakata, and Rob Procter. 2016. Pheme dataset of rumours and non-rumours.

A Implementation Details of Language Evolution

A.1 Vocabulary Compression

Twitter Corpus for Word Frequency Counting Since it's infeasible to get a corpus of all tweets to count words, we have chosen to analyze and gather statistics from existing tweets relevant to the topics of social simulation. Since some tweet links are no longer accessible, we crawled 41,736 tweets from the Twitter 15 and 16 datasets (Liu et al., 2015; Ma et al., 2016) and 52,967,084 tweets from the HiSim dataset (Mou et al., 2024b), resulting in 35,211 and 1,662,657 words, respectively.

Semantic Clustering We experimented with both top-down clustering, which involves assigning words from the corpus to synsets in WordNet (Miller, 1995), and bottom-up clustering, which encodes each word and groups them into clusters using methods like KMeans or spectral clustering. We found that the top-down approach is more controllable and less likely to group unrelated words into the same cluster, so we adopted the former method. Specifically, we first remove non-English words, and we compute the center embedding e_{S_j} of each synset S_j in WordNet and calculate the cosine similarity between each candidate word w_i and the center of every synset. The word is then assigned to the synset whose center has the highest similarity, as shown in Eq. 1.

Due to the fine-grained division of synonym sets in WordNet, many sets contain only one or two words. Therefore, we further merge similar sets using a similarity threshold of 0.8, resulting in 16,545 clusters for PHEME and 47,339 clusters for HiSim.

Intra-Cluster Selection Within each semantic cluster, we reserve words with the highest scores calculated by the score function in Eq. 2. With different reservation ratio r_w for each cluster, we can get vocabularies of different sizes, as shown in Table 2.

Tokenization To ensure normal generation by LLMs, in addition to retaining tokens corresponding to the selected words, we also preserve tokens for the LLM's special tokens, punctuation, abbreviations, and emojis.

A.2 Rule Evolution

A.2.1 Initialization

We initialize the language rules by human crafting and LLM generation. We calculate the information density of each tweet in the Twitter corpus, and summarize rules that can reflect the characteristics of these tweets. For LLMs, we ask GPT-4o how to issue rule instructions to enable efficient communication. Specifically, we obtained the following rule prompts:

Initial Rules for Evolution

1. Please respond concisely.
2. Provide a brief summary of your response.
3. Feel free to replace lengthy words or phrases with hashtags and symbols, like emojis.
4. Please use simple sentence structures.
5. Please omit unnecessary components such as subjects or predicate verbs.
6. Try using abbreviations or slang to shorten your sentences.
7. Identify your main point and communicate it directly without unnecessary details.
8. Avoid repeating ideas and removing unnecessary filler words.
9. Get to the point quickly and clearly, without over-explaining.
10. Remove words like "very" or "really" that don't add value.

A.2.2 Communication

We user the synthetic-persona-chat dataset for communication simulation. We append the sampled language rule behind the profile of agents in their system prompts. In practice, we use AutoGen (Wu et al., 2023) to generate dialogues between agents, and the system prompt used is as follows.

Prompt of Agents for Communication

You are {agent_name}. {agent_profile}
{few-shot chat history for initialization}
What will you, {agent_name}, speak next?
{rule}

A.2.3 Selection

For the fitness function in selection value, we set the hyperparameters $\lambda_{align} = 1$, $\lambda_{eff} = 0.6$ and $\lambda_{exp} = 0.6$, learning from previous work (Chen

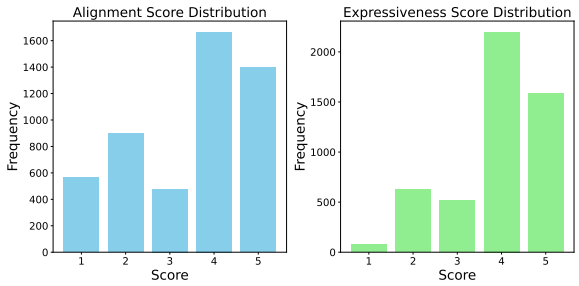

Figure 6: Alignment and expressiveness score distribution in the first iteration.

et al., 2024b). We use the following prompts to instruct GPT-4o to give the alignment score and expressiveness score to the dialogues.

Prompt for Alignment Evaluation

Please evaluate whether the agents' responses align with the persona reflected in the reference response.
Please focus on the aspects of content, emotion and atttude, and ignore differences in language structure, e.g., word choice, sentence length, emoji usage and syntax.
Agent's response: {simulated_dialog}
Reference response: {reference_dialog}
Please rate on a scale of 1 to 5, with 1 being most inconsistent and 5 being most like the persona.
Please write a short reason and strictly follow the JSON format for your response:
{{"reason": <str>, "score": <int>}}

Prompt for Expressiveness Evaluation

Please evaluate whether the agents' language is clear and easy to understand.
Agents' language: {simulated_dialog}
Please rate on a scale of 1 to 5, with 1 being most unclear and 5 being most clear.
Please write a short reason and strictly follow the JSON format for your response:
{{"reason": <str>, "score": <int>}}

Figure 6 shows the score distribution of dialogues in iteration 1, indicating that the judge model GPT-4o is capable of assigning differentiated scores. In addition, we sampled 50 dialogues for human annotation and found that GPT-4o is more consistent (Cohen's Kappa: 0.48) with human judgments than GPT-4o-mini. Therefore, we chose GPT-4o as the judge model.

A.2.4 Crossover & Mutation

We use the following prompts to conduct crossover and mutation on parent rules.

Prompt for Crossover

Please cross over the following prompts and generate a new prompt bracketed with <prompt> and </prompt>.
Prompt 1: {rule_prompt1}
Prompt 2: {rule_prompt2}

Prompt for Mutation

Mutate the prompt and generate a new prompt bracketed with <prompt> and </prompt>
Prompt: {rule_prompt}

A.2.5 Update and Iteration

In each iteration, we adopt the elitism strategy of genetic algorithm to reserve the top-5 rules in current population and generate 5 new rules through crossover and mutation. The overall process for the evolution can be described in Algorithm 1.

A.2.6 Evolved Rules

Based on the vocabularies of PHEME and HiSim, we perform rule evolution using the synthetic-persona-chat dataset. In each iteration, we obtain the following best rules:

Best Rules for PHEME

iter 1: Please use simple sentence structures.
iter 2: Respond briefly, removing unnecessary words.
iter 3: Eliminate repetitive ideas, unnecessary fillers, and respond concisely.
iter 4: Eliminate repetitive ideas, unnecessary fillers, and respond concisely.
iter 5: Remove redundancy, filler words, and respond briefly.

Algorithm 1 Evolution of the language rules

Require: Initial rules $\mathcal{P}_1 = \{p_1, p_2, \ldots, p_N\}$, size of rule population N, a set of scenarios for dialogue simulation $\mathcal{D} = \{d_i\}$, number of sampled rules for each scenario M, a predefined number of iterations T, fitness function for each dialogue F, crossover and mutation operation $Opr(\cdot)$, update strategy $Upd(\cdot)$

1: **for** t in 1 to T **do**
2: **Communication**: sample and assign rules to each scenario d_i and use LLM-driven agents to generate dialogues $\left\{\tau_i^j\right\}_{j=1}^{M}$ in these scenarios
3: **Selection**: use the fitness function to evaluate the dialogues $s_i^j \leftarrow F(\tau_i^j)$, and average the scores of the dialogues based on rules used to get fitness of each rule
4: **Crossover and Mutation**: select a certain number of rules as parent rules $p_{r_1}, \ldots, p_{r_k} \sim \mathcal{P}_t$, and generate new rules based on the parent rules by leveraging LLMs to perform crossover and mutation $\{p_i'\} \leftarrow Opr\left(p_{r_1}, \ldots, p_{r_k}\right)$
5: **Update**: update the set of rules $\mathcal{P}_{t+1} \leftarrow Upd(\mathcal{P}_t, \{p_i'\})$
6: **end for**
7: **return** the best rule p_t^* at each iteration t

Hyperparameter	Value
model	Llama-3.1-8B-Instruct
temperature	0
max_tokens	512
num_steps	max depth of each (non)rumor

Table 4: Hyperparameters of PHEME Simulation.

Best Rules for HiSim

iter 1: Avoid repeating ideas and removing unnecessary filler words.
iter 2: Please use simple sentence structures.
iter 3: Eliminate redundancy, cut filler, and be concise.
iter 4: Eliminate redundancy, cut filler, and be concise.
iter 5: Eliminate redundancy, cut filler, and be concise.

B Implementation Details of Language Utilization (Simulation)

B.1 Implementation Details

All the simulations are conducted in OASIS framework (Yang et al., 2024). We run the simulator on a Linux server with 8 NVIDIA GeForce RTX 4090 24GB GPU and an Intel(R) Xeon(R) Gold 6226R CPU. We run each simulation three times and report the average results to reduce randomness.

B.2 PHEME Simulation

We initialize the agents with user profiles and network information acquired from the PHEME dataset. We prompt GPT-4o-mini to write a short description given each user's biography on Twitter. For each instance in PHEME, we only retain replies with content for simulation and validation. The action space prompt for PHEME in OASIS simulation is as follows and the hyperparameters are shown in Table 4. Other parameters and mechanisms, such as the memory mechanism, are set to the defaults in the OASIS framework.

Action Space Prompt for PHEME in OASIS

You're a Twitter user, and I'll present you with some posts. After you see the posts, choose some actions from the following functions.
Suppose you are a real Twitter user. Please simulate real behavior.

- do_nothing: Most of the time, you just don't feel like reposting or liking a post, and you just want to look at it. In such cases, choose this action "do_nothing"
- quote_post: Quote a specified post with given content.
  Arguments:
  - "post_id" (integer) - The ID of the post to be quoted.
  - "quote_content" (string) - The content of the quote. You can 'quote_post' when you want to share a post while adding your own thoughts or context to it.

{rule_prompt}

B.3 PHEME Evaluation

For simulation results on PHEME, we include the following metrics to evaluate simulation effectiveness:

- Stance Consistency: we label the stance of each agent's and real user's *initial* response towards the source tweet given the tree-like threads, with the label space being *support*, *deny*, *query* and *comment* from (Derczynski et al., 2017).
- Belief Consistency: Following (Liu et al., 2024), we label the belief of agents and real users at the *end* of simulation. Since we observed that some agents or users did not explicitly express belief or disbelief, we added an additional category, *unknown*, alongside the existing *belief* and *disbelief* labels.
- Belief JS Divergence: To measure the belief distribution of the user group regarding fake news, we additionally incorporated the JS divergence of the belief distribution to assess the effectiveness of the simulation at the group level.

The prompts for stance and belief annotation are as follows.

Prompt for PHEME Stance Labeling

Given threads discussing a news, please label the stance of the question tweet on the source news tweet.

Treads: {threads}
Question tweet: {tweet}

Please choose from the following options:
1. support: the author of the response supports the veracity of the news.
2. deny: the author of the response denies the veracity of the news.
3. query: the author of the response asks for additional evidence in relation to the veracity of the news.
4. comment: the author of the response makes their own comment without a clear contribution to assessing the veracity of the news.

Please strictly follow the JSON format for your response:
{{"stance": <str>}}

Prompt for PHEME Belief Labeling

Please determine whether the author of the final tweet believes the source news.

Source News:{source_tweet}
Final Tweet:{final_tweet}

If the author does not believe the source news, questions the AUTHENTICITY of the source news or queries for more information about the AUTHENTICITY of the news, please label it as disbelief.
If the author expresses opinions or call for actions under the assumption that the news is true, please label it as belief.
If the author discusses something unrelated to the source news, please label it as unknown. Please label 0 for disbelief, 1 for belief and 2 for unknown.

Please write a short reason and strictly follow the JSON format for your response:
{{"reason": <str>, "label": <int>}}

B.4 HiSim Simulation

Metoo and Roe datasets in HiSim provide profiles and historical tweets of 1,000 users respectively, as well as their social networks in Twitter. We use this information to initialize the agents in the OASIS platform. To reduce the randomness introduced by the OASIS platform, we ban the recommendation systems and only enable agents to get information from external news and who they are following. The action space prompt for PHEME in OASIS simulation is as follows. The hyperparameters are shown in Table 5. Other parameters and mechanisms, such as the memory mechanism, are set to the defaults in the OASIS framework.

Action Space Prompt for HiSim in OASIS

You're a Twitter user, and I'll present you with some posts. After you see the posts, choose some actions from the following functions.
Suppose you are a real Twitter user. Please simulate real behavior.

- do_nothing: Most of the time, you just don't feel like reposting or liking a post, and you just want to look at it. In such cases, choose this action "do_nothing"
- create_post: Create a new post with the given content.
  - Arguments: "content" (str): The content of the post to be created.
- repost: Repost a post.
  - Arguments: "post_id" (integer) - The ID of the post to be reposted. You can 'repost' when you want to spread it.
- quote_post: Quote a specified post with given content.
  - Arguments:
    - "post_id" (integer) - The ID of the post to be quoted.
    - "quote_content" (string) - The content of the quote. You can 'quote_post' when you want to share a post while adding your own thoughts or context to it.

{rule_prompt}

B.5 HiSim Evaluation

For simulation results on HiSim, we follow (Mou et al., 2024b) to include the following metrics to evaluate simulation effectiveness:

Hyperparameter	Value
model	Llama-3.1-8B-Instruct
temperature	0
max_tokens	512
num_steps	14

Table 5: Hyperparameters of HiSim Simulation.

Dim.	Consistency
stance	0.94
belief	0.78

Table 6: Consistency of GPT-4o-mini judging the stance and belief when taking human evaluations as the ground-truth reference.

- Stance Consistency: we classify the *initial* response or agents and real users into three categories: *support*, *neutral* and *oppose*, towards the given target *#Metoo movement* and *the protection of abortion rights*, and compute the consistency between agents and users.
- Content Consistency: we classify the *initial* response or agents and real users into 5 types, i.e., *call for action*, *sharing of opinion*, *reference to a third party*, *testimony* and *other*.
- $\Delta bias$ and Δdiv: bias is measured as the deviation of the mean attitude from the neutral attitude, while diversity is quantified as the standard deviation of attitudes. These metrics are calculated at each time step and averaged over time. The differences between the simulated and real-world measures, denoted as $\Delta bias$ and Δdiv are reported.

The prompts for stance and content labeling are borrowed from (Mou et al., 2024b).

B.6 Evaluation Bias

Since we partially rely on LLMs for evaluation, this approach may introduce some evaluation bias. To address this, we sample 100 simulation instances and instruct two human annotators to label the stance and belief of the responses, providing them with the same information as given to GPT. Table 6 shows the consistency between the annotations of GPT-4o-mini and those of the human annotators.