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. However, large-scale LLM-driven simulations still face significant challenges in high time and computational costs. We observe that there exists redundancy in current agent communication: when expressing the same intention, agents tend to use lengthy and repetitive language, whereas humans naturally prefer concise expressions. To this end, we propose **EcoLANG**: Efficient and Effective Agent Communication Language Induction for Social Simulation. Inspired by how human language evolves through interactions, we induce a more compact language by identifying and preserving core communicative concepts at the vocabulary level and evolving efficient expression patterns at the sentence level through natural selection. We apply the induced language in various social simulations. Experimental results demonstrate that EcoLANG reduces token consumption by over 20%, enhancing efficiency without sacrificing simulation accuracy.

1 Introduction

Social simulation models how individuals interact to uncover the dynamics and emergent patterns of complex societies (Squazzoni et al., 2014; Mou et al., 2024a). The rise of large language models (LLMs) has greatly expanded this field, enabling agents to simulate human behavior at various levels, from mimicking well-known individuals (Argyle et al., 2023; Park et al., 2024) and reconstructing specific scenarios for task-solving (Hong et al., 2023; Qian et al., 2024) to modeling large-scale social dynamics (Mou et al., 2024b; Yang et al., 2024). Among these advances, large-scale social simulation focuses on the emergence of collective behaviors. Rather than faithfully replicate every individual's exact wording, researchers aim to capture the macro-level trends that inform applications in social governance, information flow, and crisis response (Lorig et al., 2021; Piao et al., 2025).

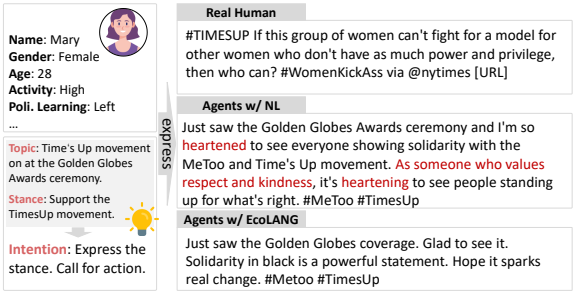

Figure 1: Responses generated by humans and agents when discussing the MeToo movement. Speakers are driven by intent when formulating their messages. While humans tend to favor efficient and concise expressions, agents using natural language (NL) often generate lengthy, redundant sentences and rely on advanced but uncommon vocabulary. This can be effectively mitigated by our proposed EcoLANG.

Despite their promise, LLM-based social simulations remain computationally expensive and inefficient. Simulating thousands of agents across millions of interactions requires significant computational resources and time, posing a major barrier to real-world deployment (Gao et al., 2024). To better understand potential inefficiencies, we revisit the characteristics of communication in both human and simulated settings. As shown in Figure 1, during communication, speakers typically form an intention, often tied to a topical stance or claim, and organize it into utterances using a variety of linguistic expressions. We observe that agents tend to express the same intention verbosely, repetitively, or with excessive detail. In contrast, humans instinctively optimize for concise and intention-preserving expression, adhering to the principle of least effort (Zipf, 2016). This raises a natural question: can agent communication be made more efficient by inducing a more compact language that reduces computational cost while maintaining core semantic content in social simulation?

In this paper, we introduce **EcoLANG**: Efficient and Effective Agent Communication Language Induction for Social Simulation. EcoLANG centers on two foundational pillars of a language system: a compact **vocabulary** guided by core communicative concepts, and efficient sentence-level **rules** evolved through interaction-driven selection. For the vocabulary, recognizing that only a limited set of concepts is pragmatically essential for communication (Wierzbicka, 1996), we construct a compressed vocabulary by clustering semantically similar words and selecting representative, high-utility terms, thereby reducing the LLM's decoding space. Then, we simulate agent dialogues with varying rule sets and apply evolutionary algorithms to iteratively refine these rules, optimizing for both efficiency and expressiveness. Once the language is induced, it is deployed in large-scale social simulations under a **transfer** setting. As EcoLANG is derived from general communication behavior and independent of task-specific architectures, it (1) reduces redundant content and lowers simulation cost, and (2) preserves agent diversity and behavioral fidelity across diverse scenarios.

We conduct extensive experiments using the open-source Llama-3.1-8B-Instruct (Dubey et al., 2024) model. The language induction process leverages both the Twitter corpus and the synthetic-persona-chat dataset (Jandaghi et al., 2023). We then validate the effectiveness of the induced language in large-scale social simulations on the PHEME (Zubiaga et al., 2016) and HiSim (Mou et al., 2024b) datasets. Experimental results demonstrate that EcoLANG significantly reduces token consumption and improves simulation efficiency without sacrificing accuracy, outperforming baseline approaches such as structured languages and conventional agent communication protocols. Our main contributions are summarized as follows:

- We propose EcoLANG, an efficient and effective agent language induction framework that features a concept-driven compact vocabulary and expression rules evolved through natural selection in communications.
- We derive a compact, generalizable agent language using EcoLANG, induced from the Twitter corpus and synthetic-persona-chat dataset, and show its generalizability across diverse downstream social simulation tasks.
- We perform extensive experiments across different scenarios, demonstrating that EcoLANG reduces inference costs while preserving simulation accuracy across different levels of granularity.

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 the outcomes of interactions among numerous agents (Mou et al., 2024a; Zhang et al., 2025). Despite these advancements, integrating LLM agents into large-scale simulations remains costly and computationally inefficient. Existing efforts to address this challenge generally fall into two categories: (1) System-level optimization. Some approaches improve efficiency by employing open-source models within distributed mechanisms (Pan et al., 2024; Yang et al., 2024), thereby reducing reliance on commercial APIs. However, these methods mainly shift the computational burden to the edge without reducing the number of inferences or response length, thus not fundamentally lowering costs. (2) Agent-level simplification. Other studies enhance efficiency by simplifying the modeling of most agents, either through integration with ABMs (Chopra et al., 2024; Mou et al., 2024b) or by reusing certain strategies (Yu et al., 2024). While these methods significantly reduce inference times, they sacrifice diversity and accuracy. By contrast, our approach reduces redundancy at the language level, improving efficiency while preserving the core content necessary for simulation.

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, 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). However, most existing work focuses on task-solving rather than social simulation, which more urgently needs to

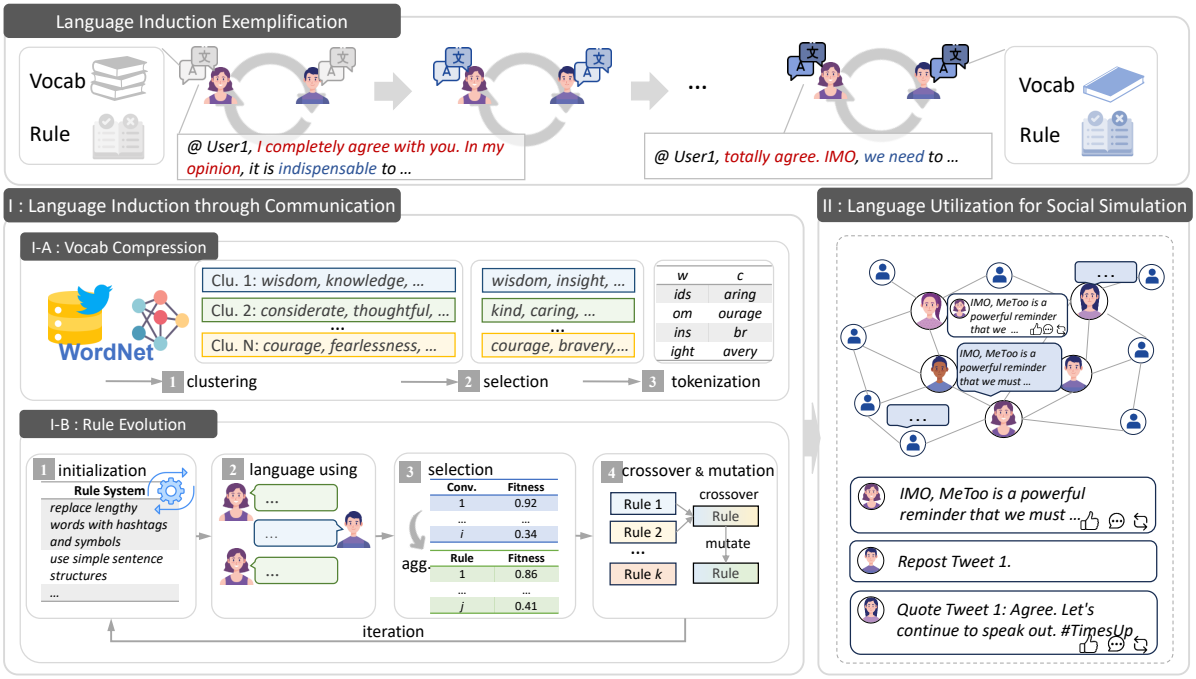

Figure 2: Overview of the EcoLANG framework. We induce the language through vocabulary compression and rule evolution in dialogue-intensive scenarios. Then, we enable agents to use this language in social simulations.

address the challenges of large-scale simulation.

3 Methodology

3.1 Problem Setup

A language system is built upon two key components: vocabulary, the set of words used to express concepts, and rules, which govern how these words are combined into meaningful sentences. To enable efficient communication, our goal is to develop a shared and streamlined vocabulary $\mathcal{V}$ and rule set $\mathcal{P}$, starting from an existing language such as natural language. The induced language should allow agents to convey intentions using simpler, more accessible terms, e.g., using *"need"* instead of the complex *"indispensable"* in Figure 2, and to adopt more concise and compact sentences overall.

Human language is shaped and evolved through use in social interactions, where communication drives individuals to iteratively refine their linguistic choices (Nowak and Krakauer, 1999), as illustrated at the top of Figure 2. This motivates our induction process: vocabulary compression to retain core communicative concepts, and rule evolution to optimize expression patterns. Together, these simulate the emergence of an efficient language shaped by communicative pressure. Given this setup, we induce language from vocabulary (Sec.3.2) and rules (Sec.3.3), and apply it to diverse social simulations (Sec. 3.4).

3.2 Vocabulary Compression

The first step in inducing a compact language system is defining its fundamental units, i.e., vocabulary. From a pragmatic perspective, effective communication does not require the full expressivity of natural language but rather depends on a limited set of core concepts that agents need to convey (Wittgenstein, 2009). Linguistic theories such as Zipf's law (Zipf, 2016) and semantic primitive theory (Wierzbicka, 1996) suggest that a small number of high-utility concepts dominate everyday communication. Inspired by this, we aim to distill a minimal yet expressive vocabulary by identifying and preserving such semantically central concepts.

To this end, we reinterpret vocabulary compression as the identification and retention of key communicative primitives. We approach this in locating conceptually coherent word groups by semantic clustering and retaining the most representative words based on pragmatic criteria, as illustrated in part I-A of Figure 2.

Semantic Clustering Natural language contains redundant words that share similar meanings but differ in style and frequency. To locate the underlying concepts, we cluster words into semantic groups. Rather than constructing clusters from scratch, we leverage WordNet (Miller, 1995) as a curated semantic hierarchy and assign each word to its most similar synset using embedding-based

similarity. This approach grounds the vocabulary in a structured conceptual space while avoiding the noise of unsupervised clustering.

Intra-Cluster Selection Within each cluster, we further filter words by assigning a score that balances two key factors: word frequency and word length. Frequent words are generally more effective at conveying intent and are better supported by the LLM due to their higher training occurrence. Meanwhile, shorter words are preferred to reduce the length of generated outputs. To integrate these considerations, we define the following scoring function:

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

where $F(w_i)$ and $L(w_i)$ denote the percentile scores of the word's frequency and token length respectively. λ_{freq} and λ_{token} are hyperparameters controlling factors' relative importance. Given these scores, we retain the top words within each cluster according to a predefined retention ratio r_w.

Tokenization Although our conceptual analysis is word-based, LLMs operate on subword tokens. We therefore map the selected words to their corresponding tokens. While these tokens may form additional words beyond our initial selection, the overall vocabulary size of LLMs is substantially reduced. We also preserve model-specific special tokens necessary for generation integrity.

3.3 Language Rule Evolution

Once the vocabulary, the basic building blocks of language, is established, the next step is to determine how these elements are organized: the rule system. In linguistics, grammar has been described as a set of simplified structures that evolved through natural selection to reduce communication ambiguity and error (Pinker and Bloom, 1990; Nowak and Krakauer, 1999). Inspired by this evolutionary perspective, we frame the construction of language rules as a search process guided by evolutionary algorithms (EAs). Our objective is to discover rule prompts that facilitate both effective and efficient communication between agents. The overall process is depicted in Part I-B of Figure 2.

Initialization The evolutionary process begins with an initial population of N candidate rule prompts $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$. These serve as guiding instructions for agents' expressions and are refined through iterative evolution. To initialize the population, we combine human-designed prompts with those generated by LLMs, following the insight that combining the wisdom of humans and LLMs yields more diverse and high-potential seeds (Guo et al.). These rules encourage agents to adopt concise expression styles. More initialization details are provided in Appendix A.2.1.

Language Using Language evolves through use. To simulate this, we let agents interact via LLM-driven dialogues under varying rule prompts. Given a set of general dialogue scenarios $\mathcal{D}$, we generate M trajectories $\tau_i^{j\,M}_{\;j=1}$ for each scenario $d_i \in \mathcal{D}$, where each trajectory is conditioned on a sampled rule $p_j \in \mathcal{P}$. These dialogues serve as the observable outcomes of applying specific language rules in communicative settings.

Selection The quality of each rule is evaluated based on how well it enables agents to communicate in a way that is both effective and efficient. Unlike task-oriented multi-agent settings that use task completion as the primary metric (Lazaridou et al., 2020), social simulation environment lacks explicit tasks. Considering that a good language should be both effective and efficient, we propose the following three dimensions: (1) *efficiency*: How efficient is the communication (i.e., token usage)? (2) *effectiveness*: How well does the agent's response reflect its assigned persona? (3) *expressiveness*: Does the agent maintain clarity and fluency without becoming overly abstract (Galke et al., 2022)? Integrating these considerations, we define the fitness of a dialogue trajectory τ_i^j as follows:

$$\begin{aligned} F(\tau_i^j) = \lambda_{align} Align(\tau_i^j) + \lambda_{eff} Eff(\tau_i^j) \\ + \lambda_{exp} Exp(\tau_i^j), \end{aligned} \quad (2)$$

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 controlling factors' relative importance. Finally, we aggregate and average the fitness scores of all trajectories associated with each rule to derive that rule's overall fitness.

Crossover and Mutation To promote linguistic diversity, we apply standard EA operations: crossover and mutation. Parent rules are sampled with probabilities proportional to their fitness

scores. Crossover involves merging components of two parent rules, while mutation is guided by prompting LLMs to creatively alter existing rules or synthesize novel variants (Guo et al.). These operations enable exploration beyond the initial prompt space.

Update and Iteration We use an elitist strategy to update the population each generation: the top $N/2$ rules are retained directly, while the remaining $N/2$ are generated through crossover and mutation. This ensures both quality preservation and exploration. After several iterations, the rule with the highest overall fitness is selected as the dominant rule of the induced language. The full evolution process is detailed in Algorithm 1.

3.4 Language Utilization in Social Simulation

With the vocabulary and rule system of the new language established, we enable agents to communicate in this language by restricting the decoding range of the underlying LLMs and integrating the evolved rules into their contextual prompts. This effectively grounds the language model's output space and behavior within the newly constructed linguistic framework. While it is possible to evolve and apply the language within the same social simulation environment, we adopt a transfer setting: the language is evolved using general multi-turn dialogue data and then transferred to downstream social simulation tasks. This decision is motivated by two key considerations: (1) large-scale social simulation data is sparse, whereas general dialogue data offers richer and more intensive communication, facilitating more efficient language induction; (2) languages emerge naturally from everyday conversations, making general dialogues a more task-agnostic and robust foundation for language development.

4 Experiment Settings

As mentioned before, we induce and utilize language in different scenarios. We filter the vocabulary using a Twitter corpus and acquire rules from dialogue-intensive scenarios. The language is then applied to social simulation scenarios, namely PHEME (Zubiaga et al., 2016) and the Metoo and Roe events from HiSim (Mou et al., 2024b). PHEME simulates the propagation and discussion of potential rumors, while HiSim models the evolution of opinion dynamics following the release of triggering news related to social movements.

4.1 Language Induction Settings

Twitter Corpus for Vocabulary Compression As our vocabulary filtering in Sec.3.2 partially relies on word frequency, we require a corpus to compute word statistics. While ideally we would analyze all tweets, this is impractical. Instead, we collect and analyze tweets related to the topics of our social simulation scenarios. Therefore, we have chosen to analyze and gather statistics from existing tweets relevant to the topics of our social simulation scenarios. Specifically, for PHEME, which models rumor propagation, we use tweets from Twitter15(Liu et al., 2015) and Twitter16 (Ma et al., 2016). For HiSim, we use tweets posted prior to the simulated events (Maiorana et al., 2020; Chang et al., 2023; Mou et al., 2024b).

Scenarios for Communication in Rule Evolution For rule evolution, we use the synthetic-persona-chat dataset (Jandaghi et al., 2023) to generate dialogues between agents adhering to specific language 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 and instruct them to role-play the corresponding individuals in conversation, thereby obtain dialogue trajectories for further selection.

Implementation Details We use Llama-3.1-8B-Instruct (Dubey et al., 2024) as the agent backbone. For vocabulary compression, we cluster semantically similar words and retain representative tokens, reducing the vocabulary to 25.4% (PHEME) and 37.5% (HiSim) of the original size of the vocabulary of Llama-3.1. For rule evolution, we simulate agent dialogues on the development set of the synthetic-persona-chat dataset and apply an evolutionary algorithm to iteratively refine expression rules. GPT-4o (Achiam et al., 2023) serves as the judge to evaluate alignment and expressiveness based on reference dialogues. Top-performing rules are retained and mutated across 5 iterations. Additional details and hyperparameters are provided in Appendix A.

4.2 Language Utilization Settings

Datasets From PHEME (Zubiaga et al., 2016), we collect 196 real-world instances, each involving 2 to 31 users discussing a source tweet, to examine whether agents can mimic user responses towards rumors. From HiSim (Mou et al., 2024b), we use

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
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 second events of #Metoo and #Roe movements, each comprising 1,000 users discussing news related to the events over time, to examine the opinion dynamics in socially interactive settings.

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) between the simulated and real-world belief distributions. More details about the experimental setup can be found in Appendix B.2.

For HiSim, we report stance and content consistency between agents' initial responses and those of real users, again using GPT-4o-mini for labeling. 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. More details about the experimental setup can be found in Appendix B.4.

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 compare our method against the following communication strategies: (1) *Base*: standard simulation without any additional rule prompts; (2) *Summary*: agents are prompted to summarize their opinions when responding, as concise expression resembles a summarization task; (3) *AutoForm* (Chen et al., 2024a): agents are prompted to automatically choose a structured format to respond, such as JSON and logical expression; (4) *KQML* (Finin et al., 1994): agents are prompted 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. Besides, we conduct additional experiments integrating these communication optimization methods with an ABM-based scalable simulation framework AgentTorch (Chopra et al., 2024), which clusters all agents into a small number of archetypes, simulates the actions of these archetypes, and then maps their responses to the corresponding agents.

Implementation Details Agents are powered by Llama-3.1-8B-Instruct (Dubey et al., 2024), and all simulations are conducted within the OASIS framework (Yang et al., 2024). Each simulation is run three times, and we report the average results. We use GPT-4o-mini to label the stance, belief and content of responses and apply Textblob to calculate the opinion score. Further implementation details can be found in Appendix B.

5 Experiment Results

5.1 Overall Performance

Table 1 presents the overall results, from which we make the following observations.

(1) Can reducing communication redundancy improve simulation efficiency? All simplified communication methods significantly reduce token usage compared to *Base*. This improvement in efficiency is not only reflected in $token_r$ but also cumulatively transmitted to $token_p$ and $token_c$ as the generated content contributes to subsequent context via memory mechanisms and social interactoions. Among the approaches, our proposed *Rule* and *EcoLANG* are the most prominent, reducing generated tokens by over 20%. Appendix B.7 further shows that combining these methods with AgentTorch, an approach that alters the simulation paradigm, can further boosts efficiency, with our method achieving the best. However, AgentTorch comes at the cost of reduced diversity and accuracy

Method	PHEME					HiSim				
	CosSim↑	Jaccard↑	word_JS↓	Δ_{l_s}↓	Δ_{l_w}↓	CosSim↑	Jaccard↑	word_JS↓	Δ_{l_s}↓	Δ_{l_w}↓
Base	0.662	0.037	0.307	31.63	0.12	0.782	**0.067**	0.321	13.60	0.15
Summary	0.651	0.039	0.306	29.94	0.10	0.769	0.062	0.320	9.98	0.14
AutoForm	0.662	0.039	0.306	23.63	0.18	0.763	0.061	0.324	5.38	0.15
KQML	0.653	0.035	0.307	32.57	0.16	0.757	0.061	0.321	8.22	**0.10**
Vocab	**0.671**	0.039	0.305	30.79	0.06	**0.789**	0.064	0.320	10.20	0.12
Rule	0.661	0.039	0.299	22.43	0.09	0.774	0.062	0.319	4.25	0.15
EcoLANG	0.661	**0.040**	**0.298**	**22.33**	**0.05**	0.775	0.065	**0.309**	**3.26**	0.13

Table 2: Comparison of semantic similarity and length between agent responses and real user responses. We report the best performance in **bold** format and the second best in underlined format.

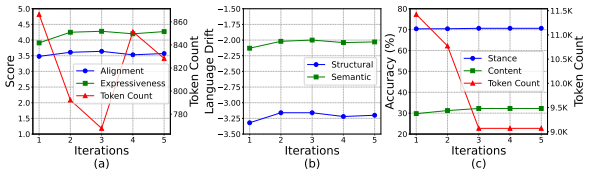

Figure 3: (a) Average fitness score change and (b) language drift change on synthetic-persona-chat simulated dialogues across iterations; (c) Performance and token consumption in HiSim using the best language rules acquired across iterations.

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 3: 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.

of agent-generated content, suggesting that such paradigms should be used with caution.

(2) Does simplifying communication compromise the simulation effectiveness? 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 induction, our method is able to enhance efficiency while maintaining leading simulation accuracy.

(3) Does vocabulary compression enhance performance or efficiency? Vocabulary compression does not significantly affect token usage, as changes in individual word lengths do not substantially alter overall sentence lengths. However, simulations still achieve comparable or even better performance after compression, e.g., in HiSim, suggesting that standard LLM vocabularies may be redundant for such social simulations. Theoretically, removing these tokens from LLM's vocabulary could enhance the model's inference efficiency and reduce GPU memory usage.

5.2 Finer-Grained Evaluation of Languages

Although the previous part has discussed the effectiveness of different methods in social simulation, where relatively macro dimensions are considered, some may be concerned that language compression could risk losing fine-grained individual semantics. To address this, we conduct a more detailed evaluation of agent language. Table 2 reports the semantic and length differences between agent-generated and real user-generated responses across different methods. The metrics include sentence embedding cosine similarity (CosSim), lexical overlap of responses (Jaccard), JS divergence of word distributions (word_JS), as well as differences in average sentence length (Δ_{l_s}) and word length ((Δ_{l_w}) in tokens. We summarize the key findings as follows:

(1) Word Usage Patterns: **Even without language compression**, i,e., *Base*, **agents' responses exhibit low Jaccard similarity with real user tweets**, highlighting the inherent divergence between LLM-generated and human-written texts. This gap is likely due to the agents' limited access to personal or contextual knowledge and the biases introduced by LLMs. Nevertheless, most approaches maintain comparable semantic similarity to *Base*, with **our method outperforming others** in preserving meaning.

(2) Response Length Patterns: Agents tend to produce longer and more complex responses than real users, who generally favor brevity and simplicity in social communication. This aligns with the redundancy issues discussed in the introduction section. Compared to baselines, **our method produces shorter and more concise utterances, which not only improve communication efficiency but also**

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

better align with the communication habits of real users in general.

5.3 Tracing the Evolution of Language Rules

To better understand how language rules evolve, we analyze both the progression of dialogue fitness scores and linguistic shifts across iterations, as well as their downstream effects on social simulations. Figures 3(a) and (b) illustrate the trends observed during the evolution process on synthetic-persona-chat dialogues. Beyond the fitness scores defined in Sec 3.3, we also track two additional metrics: structural drift and semantic drift (Lazaridou et al., 2020). Structural drift assesses fluency and grammaticality relative to natural language, while semantic drift captures how well the generated language preserves the literal meaning of intended targets. Results reveal that as evolution progresses, **language fitness improves overall: alignment and expressiveness increase, and token consumption decreases**. Simultaneously, both structural and semantic drift decline, **indicating improved language quality** despite these metrics not being directly optimized during training. **These improvements translate into downstream gains** as well: as shown in Figure 3(c), simulations guided by the evolved rules demonstrate higher accuracy and lower token usage. However, after several iterations, the fitness score no longer improves, and the optimal rules provided for the simulation tasks remain unchanged, suggesting that the evolution process may have converged.

5.4 Unpacking the Impact of Vocabulary

We further explore the impact of the vocabulary on the simulation. As shown in Table 3, since it is necessary to ensure that at least one word is retained for each semantic cluster, changing the retention ratio has a 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.

5.5 Exploring the Transferability of Language Rules 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 apply 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 4 show that the rules evolved on Llama-3.1 **can also enable other models to communicate efficiently, again demonstrating the transferability of EcoLANG**.

5.6 Case Study and Error Analysis

We analyze exemplary instances of both effective and ineffective communication in Appendix B.8. Several emergent patterns are observed in the evolved language: it adopts **more compact and expressive sentence structures**, shows a **reduced use of titles and identity labels**, and shifts from surface-level, repetitive event descriptions to more **abstract but in-depth discussions** centered around thematic vocabulary, such as *"Justice for Victims"* and *"Accountability"*. However, sometimes 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.

6 Conclusion

We introduced EcoLANG, a language induction framework, 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.

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*.

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.

Salvatore Giorgi, Sharath Chandra Guntuku, McKenzie Himelein-Wachowiak, Amy Kwarteng, Sy Hwang, Muhammad Rahman, and Brenda Curtis. 2022. Twitter data of the #blacklivesmatter movement and counter protests: 2013 to 2021.

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.

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.

Fabian Lorig, Emil Johansson, and Paul Davidsson. 2021. Agent-based social simulation of the covid-19 pandemic: A systematic review. *JASSS: Journal of Artificial Societies and Social Simulation*, 24(3).

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.

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, 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*.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, et al. 2025. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*.

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.

Anna Wierzbicka. 1996. *Semantics: Primes and universals: Primes and universals*. Oxford University Press, UK.

Ludwig Wittgenstein. 2009. *Philosophical investigations*. John Wiley & Sons.

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*.

Xinnong Zhang, Jiayu Lin, Xinyi Mou, Shiyue Yang, Xiawei Liu, Libo Sun, Hanjia Lyu, Yihang Yang, Weihong Qi, Yue Chen, Guanying Li, Ling Yan, Yao Hu, Siming Chen, Yu Wang, Jingxuan Huang, Jiebo Luo, Shiping Tang, Libo Wu, Baohua Zhou, and Zhongyu Wei. 2025. 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*.

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 Induction

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 10,673,881 tweets from the social movement dataset (Maiorana et al., 2020; Chang et al., 2023; Giorgi et al., 2022) that were posted before the simulated events in HiSim occurred, resulting in 35,211 and 1,662,657 unique 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. **??**.

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. 1. With different reservation ratio r_w for each cluster, we can get vocabularies of different sizes, as shown in Table 3.

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 use the validation set of 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

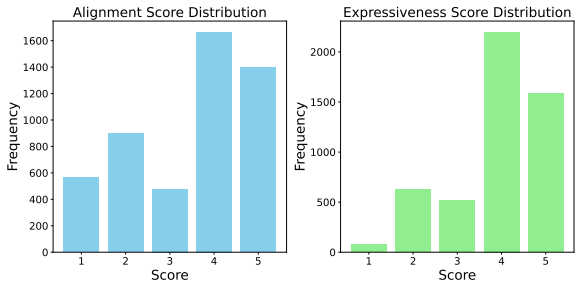

Figure 4: Alignment and expressiveness score distribution in the first iteration.

$\lambda_{exp} = 0.6$, learning from previous work (Chen 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 4 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 5: 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 (Social 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 5. 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 6. 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 6: Hyperparameters of HiSim Simulation.

Dim.	Consistency
stance	0.94
belief	0.78

Table 7: 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 of 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 of 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). Notably, we focus on the macro setting from the original HiSim paper, which involves continuous, multi-turn interactions to simulate complex social dynamics over time. However, we did not include HiSim as a baseline, as it adopts a different agent architecture based on AgentVerse from our implementation on OASIS.

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

Method	stance↑	content↑	$\Delta bias$↓	Δdiv↓	$token_r$↓	$token_p$↓	$token_c$↓	CosSim↑	Jaccard↑	word_JS↑
AgentTorch	67.87	31.81	0.098	0.024	2.5K	0.48M	94.34K	0.666	0.058	0.360
w/ Summary	68.28	32.27	0.151	0.032	1.5K	0.48M	88.58K	0.698	0.054	0.366
w/ AutoForm	65.19	32.52	0.092	0.026	1.8K	0.47M	84.09K	0.726	0.065	0.359
w/ KQML	67.64	32.20	0.110	0.018	1.5K	0.49M	88.90K	0.693	0.064	0.361
w/ Vocab	67.61	**33.56**	0.098	**0.013**	1.6K	0.47M	91.39K	0.726	**0.066**	0.359
w/ Rule	67.76	33.35	**0.086**	0.015	1.3K	0.47M	80.96K	0.735	0.066	**0.356**
w/ EcoLANG	**68.63**	33.45	0.099	0.017	**1.2K**	**0.46M**	**78.03K**	**0.740**	0.066	0.358

Table 8: The results of the communication simplification method combined with AgentTorch on HiSim dataset. Only 36 prototype agents were used in all experiments. The number of prototypes is determined by the combination of fundamental attributes such as gender, political inclination, and activity level.

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.

stance and belief of the responses, providing them with the same information as given to GPT. Table 7 shows the consistency between the annotations of GPT-4o-mini and those of the human annotators.

B.7 Integration with AgentTorch

The communication simplification methods are orthogonal to AgentTorch (Chopra et al., 2024) and can be combined with it to enhance efficiency further. To understand the potential of combining different communication simplification methods with this paradigm, we conducted experiments by integrating different communication simplification methods with AgentTorch. Since each scenario in PHEME involves a relatively small number of agents, further clustering them into a few prototypes would overly simplify the agent population, resulting in homogeneous content and limiting the generation of meaningful responses. Given these limitations, we determined that AgentTorch is not a suitable baseline for PHEME and therefore conducted experiments only on HiSim.

The results in Table 8 show that combining all methods with AgentTorch can further improve simulation efficiency, reducing token consumption by up to an additional 80% compared to Table 1. Among these, our method demonstrates advantages in both effectiveness and efficiency, highlighting its robustness. However, integrating with AgentTorch has some side effects. While using a small number of agents drastically reduces token usage, it also compromises the diversity and accuracy of agent responses, leading to noticeable shortcomings in content-related metrics, e.g., stance and CosSim, compared to results in Table 1 and Table 2.

B.8 Case Study

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.