EFFECTS OF PERSONA PROMPTING ON LANGUAGE AGENTS IN A COMPETITIVE HIDDEN INFORMATION GAME

Collisteru

A thesis submitted to the
Faculty of the Engineering School of the
University of Colorado in partial fulfillment
of the requirement for the degree of
Bachelor of Science
Department of Computer Science
2025

The University of Colorado at Boulder

October 2025

To William H. Carter Sr. for his spiritual support, to Robert N. Allton for his material support, and to all my family for their enduring love.

Abstract

Language agents have become an important use case for LLMs. While prompting effects on one language agent have been well studied, little attention has been paid to how prompting affects inter-agent interactions. We develop a novel experimental technique based on the Myers-Briggs Type Indicator to test the effects of persona prompting on multiple language agents in the hidden information game Werewolf. In this observational study, personas affect game performance in complex ways. Agents that are told to exploit later and listen carefully to interlocutor utterances outperformed others. The interaction of agents with multiple personas led to unintuitive and complex results, revealing the multifaceted nature of persona prompting. This model can be extended for further analysis and will have deep implications as language agents become part of the workforce.

1 Background

The AI community has experienced a surge of interest in transformer-based large language models (LLMs). These are neural network architectures built with multi-headed attention mechanisms that process natural language as a series of tokens and respond with a sequence prediction derived from a massive corpus of training data [3]. LLMs have astounded researchers and laymen alike with their long-lasting scaling effects as their capabilities have grown in tandem with their size and training data volume [15]. Recent LLMs have demonstrated capabilities of abstraction, comprehension, memorization, and creativity [4, 21, 30], although they still have limitations [7, 47].

Traditional LLM implementations are limited by unimodality— their input and output channels process only text. This is a fundamental difference from natural intelligences (NIs) such as humans, which are embedded in the natural environment. NIs learn from interacting with, influencing, and being influenced by, the environment around them. Compared to natural senses such as vision, hearing, and touch, text is a narrow channel of information [26]. The text-only training methodology of current leading LLMs limits them and contributes to their tendency to hallucinate [49].

To bridge this gap, researchers seeking to reproduce the behavior and capabilities of NI are increasingly embedding LLMs in environments. This requires giving the LLM a "body" it can use to interact with the environment, which usually comes in the form of structural tools. These can include a profile module that grants the agent its foundational identity, a memory module that stores agent experiences, a planning module that allows the agent to form intentions, and tools that allow the agent to perform actions. These modules are implemented in various ways that make use of querying the LLM as well as algorithms that process information about the surrounding environment [32, 42]. In some cases, these LLMs have been equipped with self-adaptation abilities and can modify or improve themselves in response to experiences [28]. The resulting agents are called language agents [10].

Guo et al. [10] identify two major categories of application for language agents: problem-solving and simulation, each with multiple subcategories. Language agents have already been deployed for uses as diverse as mental health support, studies of political science and economy, social simulation, documentation and data management, and embodied artificial intelligence [42]. In this paper, we analyze their behavior through the lens of social science.

2 Introduction and Motivation

As LLMs grow in power and capability, they are becoming part of our daily lives. Language agents will similarly become more common. They will be increasingly applied to solve problems, provide companionship, and populate virtual spaces. The study of language agents gives us hints on how this technology will be built and used.

It has been demonstrated that language agents can interact with each other, put on personas, and hide their true intentions in large environments populated by other language agents [22]. Numerous other papers have explored the possibilities of multiple language agents interacting in a social environment [12, 19]. However, there remains a gap in understanding how language agents behave in a competitive environment. In this paper we aim to fill that gap.

We propose a limited cooperation game in which language agents interact with each other. Each agent is a member of one of two teams. Each team is given a goal that can only be fulfilled by carrying out complex strategies involving both cooperation and competition with other agents. We use this game to study persona prompting, an important technique in language agent creation.

2.1 Werewolf

Werewolf, also known as *Mafia*, is a seven-player logic puzzle and one of the world's most popular party games. In an in-person party setting, one person is designated the moderator and is tasked with enforcing the rules of the game. The other players are divided into two teams: the werewolves and the villagers. The werewolves know which other players are werewolves, but the villagers don't know anything about other players' identities.

The game cycles between two phases: day and night. During the night phase, the werewolves vote to eliminate a player from the game. The villagers cannot see anything that happens during the night phase. Next comes the day phase, which is public to all players. In the day phase, all players vote to eliminate a chosen player from the game.

The werewolves' goal is to eliminate all the villagers without being discovered, and the villagers' goal is to identify the werewolves and vote to eliminate them. When one team is completely eliminated, the other team wins [46].

Some variants of Werewolf include special roles, such as Seer, for the villagers. For

the sake of simplicity, we do not include these in our simulation.

In the context of AI, Werewolf requires a number of skills computer scientists are interested in: natural language, logical reasoning, social interaction skills, and a theory of mind, among others. As a result, the game has become a useful testing ground for the study of AI agents [35, 36, 40, 48].

2.2 Game Theory of Werewolf

Due to Werewolf's popularity in the AI literature, much work has been done on its game-theoretic considerations [44]. Traditionally, a totally random strategy has been considered optimal for both teams in the game of Werewolf without any special abilities. However, Shitong Wang proved that there is a strategy for the villagers that weakly dominates the random strategy. Hence, despite being a hidden information game, there is room for strategic thinking in Werewolf that agents with differing personalities can take advantage of.

The probability of winning a game of werewolf can be derived from the number of villagers and the number of werewolves. Let us call the number of werewolves w and the number of villagers v. Holding w constant, the probability of the villager group winning exhibits a slow stepwise increase as v increases. The villager win probability sharply decreases as w increases. To avoid complications arising from these considerations, we assume $w = 2$ and $v = 5$ during all simulations.

2.3 Persona Prompting and the Myers-Briggs Type Indicator (MBTI)

The study of prompting has emerged in tandem with the rise of transformer-based LLMs. A prompt is an input to a generative AI model that is used to guide its output [43]. A subfield has emerged in which different prompts are tested and their effects of performance evaluated. It is often the case that the precise wording and content of the prompt has an effect on LLM performance.

For example, prompters can stimulate in-context learning by adding examples of the task the model is meant to do in the prompt itself. This can make more explicit what exactly the model is supposed to do, improving results, a phenomenon known as few-shot learning.

Persona prompting is another type of prompting in which the model is cast into a role before task execution. Researchers have found that going beyond the classic "helpful assistant" role can improve output [50].

In the literature, **personas** typically supply occupation and demographic information. There's been little research into how personalities affect LLMs [1]. Furthermore, persona prompting research has so far focused on LLMs and not language agents. Since our personalities and life experiences deeply modify how we interact with the world, it is natural to wonder whether personalities will affect how language agents interact with their world.

However, to make this question scientific the notion of "personality" must be quantified. Boundaries must be drawn in order to form testable hypotheses. For this purpose we choose for this study the Myers-Briggs Type Indicator (MBTI).

The MBTI originates from the thought of Carl Jung, who believed that each person is born with an innate and unique disposition toward the world. For Jung, these dispositions are neither atomic nor arbitrary, but relate to the substructure of reality. Humans share an unconscious understanding of this substructure, and their personality is their way of relating to it [39].

The MBTI extended and specified Jung's work. Katharine Cook Briggs and her daughter Isabel Briggs Myers aimed to specify the personality types and developed a test they claimed could categorize a person on four binary dimensions [27]:

Table 1: The MBTI framework, as an extension of Jung's typology.

Dimension	**Description**
Introversion (I) / Extroversion (E)	Direction of attention: inward vs. outward
Sensing (S) / Intuition (N)	Mode of gathering information
Thinking (T) / Feeling (F)	Basis for decisions and judgments
Judging (J) / Perceiving (P)	Approach to structure and organization

It's important to note that the labels for tendencies do not necessarily correspond to colloquial usage. A *Judging* agent is not necessarily more judgemental than other agents. *Judging* is a label for an approach toward organization. A *Judging* agent will prefer having more structure and more closure in making decisions, whereas a *Perceiving* agent will wait longer before making a decision.

The validity of the MBTI as a psychological classification for humans is dubious [33]. That doesn't matter for our purposes. Our intent is not to classify humans but to build a structure for agent personalities. For us, an explicit, dimension-based, and categorical

schema like the MBTI is ideal.

For this project, we use only the $Thinking \longleftrightarrow Feeling$ ($T \longleftrightarrow F$) and $Judging \longleftrightarrow Perceiving$ ($J \longleftrightarrow P$) dimensions, seeing these as the most meaningful in the context of a short-term iterated game.

2.4 Justification

When Xu et al. [48] developed a framework for language agents playing werewolf together, they found that each agent used a variety of trust, confrontation, camouflage, and leadership strategies. However, it is still unknown how the behavior of agents during the game can be modified. How does persona prompting affect how the language agents play games?

When applied to Werewolf, this question yields data that perfect information games like chess and go do not. Language agent behavior in a game of social deception can be treated as a bellwether to the efficacy of deliberate language agent deception. This pertains to the future of human-AI cooperation, especially online, which is likely to become a part of daily life for knowledge workers. The results have implications in every domain where AI agents will be applied, including education and tutoring, work and productivity, healthcare, customer service, government, and everyday interfaces.

The results of this study help us understand the future relationship between humans and AI agents in the digital world.

3 Research Question and Hypotheses

We create a simulation in which language agents communicate with each other in natural language and player elimination is fully implemented. We create agents to play against each other in sets of forty games, each with various configurations of personality modules in order to answer the following question:

How do different MBTI personalities affect a language agent's performance in Werewolf?

Having developed a number of personas along archetypes measured by the MBTI, we will do a quantitative analysis of how these changes of personality led to differences in playstyle and win rates.

3.1 Hypotheses

We hypothesize that the Thinking (T) and Judging (J) personalities will lead to better performance for both teams. This is because of their emphasis on rational thinking and acting quickly.

Being the first player to move is an advantage in most games: this is called the advantage of the initiative [41]. It is therefore reasonable to expect that a personality that encourages an agent to act more quickly would improve its performance in Werewolf. We therefore expect Judging to outperform Perceiving, since Judging emphasizes quick and decisive decisions.

4 Methods

We create a novel language agent simulation architecture to study the effects of persona prompting on gameplay in werewolf. We embed seven language agents in a simulation of Werewolf.

We used Python as our main language. Our experimental code and data are available on GitHub at this address:

https://github.com/scarter-the-buff/werewolf-personas.

4.1 Program Architecture

A number of agent architectures exist to facilitate the use of language agents. Rather than building our own architecture, it is more efficient to choose a premade one. The most popular are CrewAI, AutoGen, LangChain, and Pydantic AI. We choose CrewAI for its relatively advanced abilities to orchestrate many agents at once. For our agents we connect CrewAI to gpt-4o-min and perform the simulation. This is the first Werewolf simulation system of its kind in the literature.

A Werewolf simulation is more complex than a typical CrewAI task since it needs to calculate internal game states and winners and losers. Furthermore, the prompt needs to be modified each round to take into account both the novel personalities and remind the agents which players are available to be eliminated. The solution involved a novel program architecture connecting both internal architecture and agent tasks.

The driver code timed itself and executed a series of forty games, as shown in the *Initialization* and *Program* rows of Fig. 1. The *Game* row represents the flow of the games themselves.

Each game has multiple rounds, and these continue until the game is over. The game ends when one of two conditions are met: either there is only one player remaining, or all the remaining players belong to the same team, which is declared the victor.

As in Werewolf, each round has two phases: the night phase (highlighted in red in Fig. 1) and the day phase. In the night phase, the werewolves only vote on eliminating a non-werewolf, which happens in the voting module (Fig. 2). We prompt the agent with both the structure of the game and an example of the format their responses will be in. This promotes in-context learning [43].

After the players that have been voted for are eliminated, the day phase occurs in which all the players vote to eliminate someone (Fig. 2). The elimination happens and the simulation checks once again for game over. Once forty games have been played this way, the simulation records the gameplay records and the time taken.

4.2 Agent Architecture

We create seven agents representing Werewolf players: five villagers and two werewolves.

Each game player is modeled as an agent with certain attributes and tasks. Each agent has a role representing which team they are on (villagers or werewolves) and a personality.

Multiple iterations of the software architecture were required to achieve a working model of the game. The first attempt was a hierarchical structure in which a manager agent oversaw a free form conversation between the other player agents. This failed because the manager agent didn't reliably delegate the program flow to the other players, and kept hallucinating the behavior of the other agents on its own and the entire flow of the game.

In the second iteration, we moved to a sequential architecture, In this, the agent is prompted in accordance with its personality and the game situation at the beginning of its turn. In this version there were eight agents – the seven player agents and a final "notetaker" agent who recorded and tallied the votes of the other agents at the end of the loop described in Fig. 2. This also didn't work because there was no direct channel

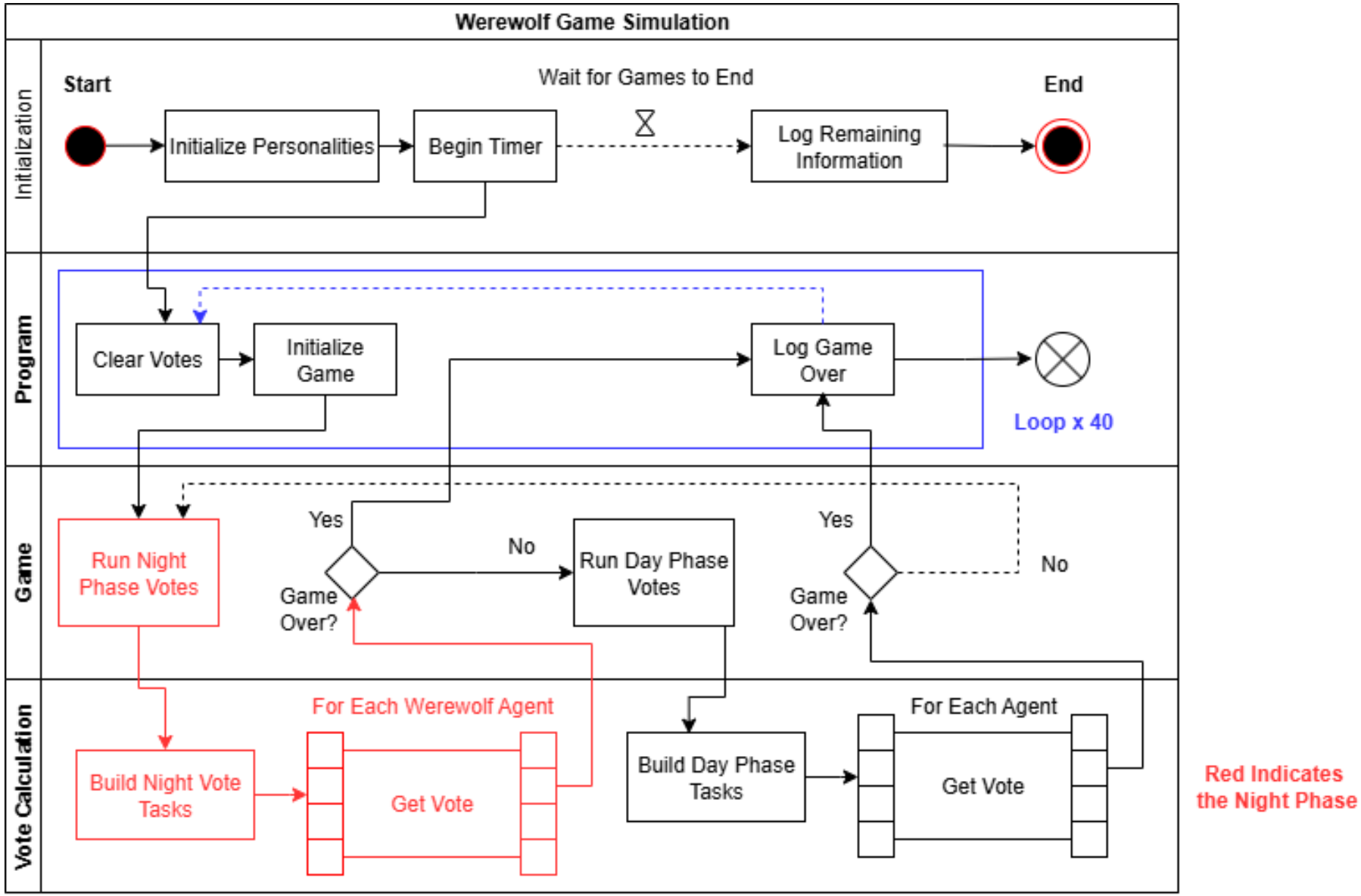

Figure 1: UML Activity Diagram for the Simulation.

of information from the other agents to the notetaker.

Finally, we removed the notetaker agent and integrated the voting more closely with CrewAI. CrewAI uses a system of "tasks" to organize the actions that an agent needs to do. Our innovation was to split each voting action into its own separate task, instead of having the entire round be one large task. This allows us to take the information from each vote directly into the Python architecture. With this architecture, the voting is finally reliable enough to play full games well. However, it comes at the cost of needing to reinitialize the agent each round, so our simulation doesn't support memory kept by the agent itself. Instead, we simulate memory by telling the agent what has happened already in the current round and which players are available this round. This simulacrum of memory is limited. Changes to the research design will be needed for us to study how language agents incorporate memory.

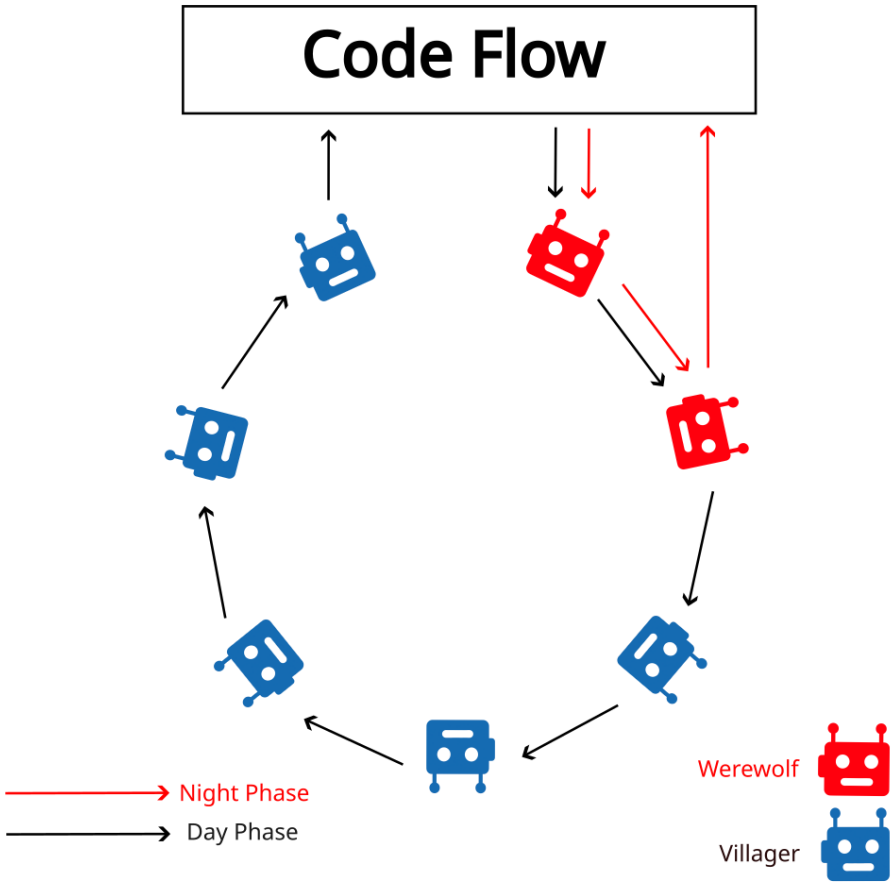

Figure 2: Diagram of the order of player votes.

Fig. 2 illustrates the voting process. In the night phase, we iterate through the night tasks, one per werewolf, and have the werewolves vote on which player they want to eliminate. Their answers are collected by the program architecture and this is used to

eliminate a player. During the day phase, the same thing happens except we iterate through all the players instead of only the werewolves.

Each task has a description and an expected output. The description parameter is customized at runtime for each agent and for each round. The agent is reminded whether they are a werewolf or a villager and which players they can choose. The expected output parameter tells them how to display their response in a way that the code can directly parse.

4.3 Personality Modules

We add MBTI-based personality modules (as explained in §2.3) to the system prompts to create a persona prompt. The agents were given this persona prompt at the beginning of each task.

Park et al. initialized each of their language agents with a short paragraph to explain their identity and background [32]. We borrow this idea to create the personality module. We investigate whether this personality module affects agent behavior and performance in the game. If they are correlated, that would make identity modules a way to program personality in language agents.

Basing our work on the MBTI [17, 34], we create four personalities from the two relevant MBTI axes we choose as per §2.3. These are p_{FP} (Feeling + Perceiving), p_{FJ} (Feeling + Judging), p_{TP} (Thinking + Perceiving), and p_{TJ} (Thinking + Judging). A personality is a combination of two positions along the two axes. These positions are called **personality components**. We write a short prompt snippet for each personality component that we give to the language agents, as shown in Table 4.3.2

These descriptions in the identity module are original but based on the MBTI Manual [27].

4.3.1 Control Conditions

To calibrate the effects of the personalities on agent behavior and success, and to confirm that the personalities had the effect we expect, we create three simple conditions that we apply in the same ways as the other personalities and did not use the MBTI:

1. The control condition, in which we apply no personality.

You tend to process information in a logical manner and you think that reason is the most important factor in coming to decisions. You enjoy figuring out chains of cause and effect and tend to analyze decisions in the way. You are concerned with objectivity and lack of bias.",. You tend to make decisions slowly and seek more information. You like to keep options open and explore rather than exploit. You tend to seem spontaneous, curious, and adaptable. You are concerned with receiving information as long as possible in an effort to miss nothing that is important... Werewolf vote. You are Player 2. Choose exactly one VILLAGER from: Player 3, Player 4, Player 5, Player 6, Player 7. Output ONLY the chosen player's exact name, e.g., 'Player 5'. No extra text.Players previously voted like so: ['voter': 'Player 1', 'vote': 'Player 4']

Figure 3: An example of a task prompt we give to a language agent before voting. This agent has personality p_{TP}.

2. The aggressive condition, in which we tell the agent to act aggressively.

3. The suppressed condition, in which we tell the agent to suppress its own capabilities by making suboptimal choices in the game.

We develop three sets of personalities: control-control, aggressive-suppressed, and suppressed-aggressive. We hypothesized that the team that received the aggressive condition will perform far better than the team that receives the suppressed condition. If the data were to prove this hypothesis correct, it would validate the idea of using agent personalities in this context.

The results we received confirmed our hypothesis that the personalities have the expected effect, as shown in Table 2.]

Villager Personality	Werewolf Personality	Personality Effect	Villager Win Rate
None	None	0	68%
Aggressive	Low-Effort	+0.28	90%
Low-Effort	Aggressive	-0.22	40%

Table 2: Villager win rates under different personality configurations.

4.3.2 MBTI Personalities Used

We pair each of the personalities with each other and measured the relative win rate of the villagers for each pair. The prompt for each component is shown in Table 3.

Personality Component	Prompt / Description
T (Thinking)	You tend to process information in a logical manner and you think that reason is the most important factor in coming to decisions. You enjoy figuring out chains of cause and effect and tend to analyze decisions in this way. You are concerned with objectivity and lack of bias.
F (Feeling)	You tend to process information by reading its emotional temperature. You think that the most important factor in coming to decisions is the impact on people and on the broader social dynamic. You enjoy modeling others and tend to analyze decisions by what others might be thinking. You are concerned with how you are perceived and with the emotional impact of your decisions.
J (Judging)	You tend to make decisions quickly and prefer to seek closure. You are concerned with planning operations and organizing activities. You tend to shut off your perception as soon as you have observed enough to make a decision, and you prefer your outer behavior to be organized, purposeful, and decisive.
P (Perceiving)	You tend to make decisions slowly and seek more information. You like to keep options open and explore rather than exploit. You tend to seem spontaneous, curious, and adaptable. You are concerned with receiving information as long as possible in an effort to miss nothing that is important.

Table 3: Prompts Defining Personality Components

These descriptions were included in every prompt given to the language agents during the game.

5 Data

5.1 Monte Carlo Simulation and Significance

In order to properly interpret the effects of the changing personalities on the simulation, we need to understand its basic behavior. While the control condition from §4.3.1 provides us with some evidence of the simulation's behavior without personalities, we need more data to form a confidence interval than the API-based simulation, which is constrained in both time and money, could provide.

We solve this problem by implementing a Monte Carlo variant of the simulation. It served as a control group with a substantially larger sample size. The variant is a replica of the actual system wherein the choices are made randomly instead of by AIs. We run the simulation 1000 times and take repeated samples of forty games to reflect our actual experimental design. In the resulting sampling distribution, $\mu = 0.63$, similar to the control group of the actual simulation ($\mu_s = 0.68$). This confirms that the

Monte Carlo variant is a good approximation of the behavior of the simulation without personalities.

We plot the average villager win rate for each sample as a histogram in Fig. 4, representing a binomial distribution approximation of a normal distribution.

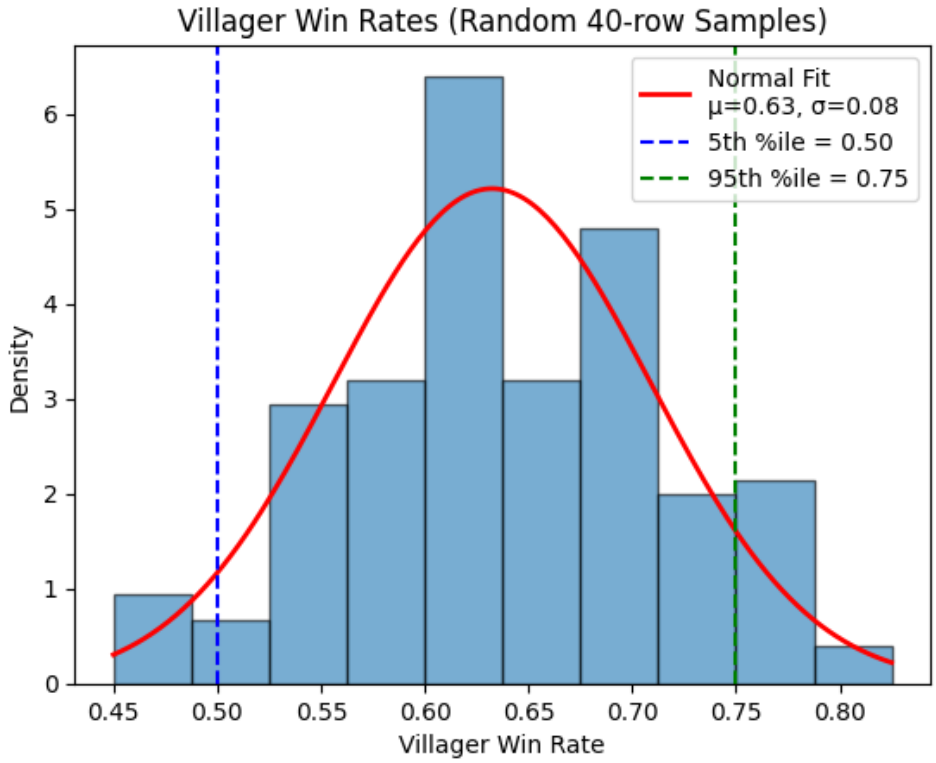

Figure 4: Distribution of win rates from Monte Carlo sampling.

5.2 Personality Pair Effects

Let $\mathcal{P}$ be a set of personalities, where each $p \in \mathcal{P}$ may be assigned to a language agent. This set includes the empty personality $\emptyset$, representing the default model without personality modification.

During our experiments, we keep the personalities in each team uniform. Specifically, the werewolves $\mathbf{W}$ and the villagers $\mathbf{V}$ are each assigned a personality. We record the win rate ρ for the villagers for each game $G(\mathbf{W}_{p_1}, \mathbf{V}_{p_2})$. To quantify the impact of the personalities, we define the effect size e as $e = \rho - \mu$, where $\mu = 0.63$ is the win rate from $G(\mathbf{W}_\emptyset, \mathbf{V}_\emptyset)$

Across multiple personality configurations we observe values of e that were significant with respect to our Monte Carlo model with $p \leq 0.05$, confirming our initial research

question. That being said, we should not interpret nonsignificant values as evidence of no effect, as they may reflect insufficient power derived from our sample size.

The effect size e for each personality pair is documented in Fig. 5. A blue tone indicates that the pairing favored the villagers (positive e), whereas a red tone indicates that the pairing favored the werewolves (negative e). The saturation of the tone captures the intensity of the advantage.

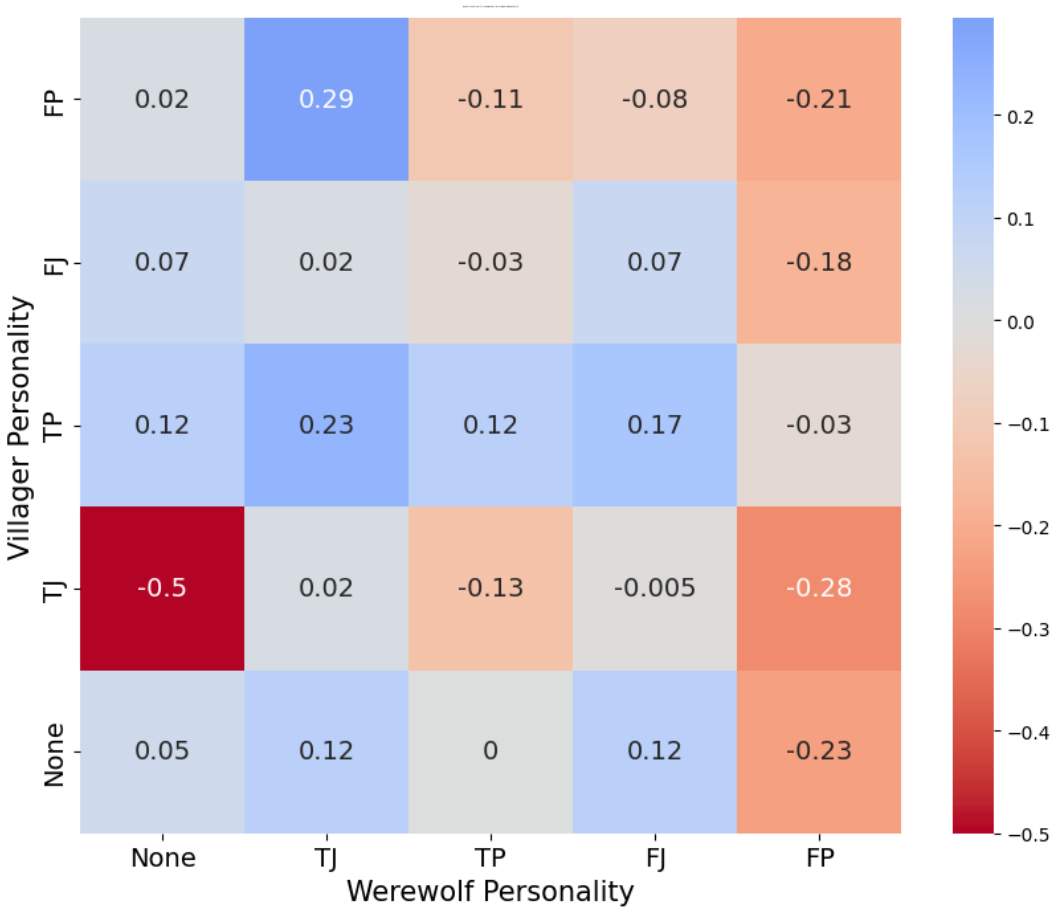

Figure 5: Matrix of Effects on the Performance of the Villagers from Every Personality Permutation

We note that ρ for the $G(\mathbf{W}_\emptyset\mathbf{V}_\emptyset)$ cell is within the margin of error from the theoretical value of μ obtained without personalities from the Monte Carlo simulation. That similarity demonstrates the validity of the Monte Carlo simulation as a proxy for the $G(\mathbf{W}_\emptyset\mathbf{V}_\emptyset)$ condition.

5.3 Average Personality Effects

To discover which personalities did the best overall, we average the advantage each personality gave to the team that possessed it across the games. This gives us the best and worst performing personalities overall, summarized in table 4:

Table 4: Average Personality Effects

Personality	**Average Advantage**
FP	+0.150
TP	+0.076
FJ	-0.032
TJ	-0.043

We observe surprising results in table 4. Firstly, p_{FP} has the best average largely due to the great boost it gives to the werewolves. This is unexpected given our hypothesis (§3.1).

p_{TP} has the second highest average, and this can be seen on Fig. 5 from the weak cross of color (§6.1) it forms in the very middle of the graph, which has a strongly blue row and a weakly red column. After this, p_{FJ} and p_{TJ} both perform poorly, although neither to as great an extent as p_{FP} and p_{TP} perform well.

We can derive the average effect per component by taking the average of the effects of each of the two personalities that share that component:

Table 5: Average Effect for Each Personality Component

Personality	**Average Advantage**
P	+0.118
F	+0.060
T	+0.006
J	-0.038

We must be careful when interpreting table 5 since its entries are averages of averages. In it the details that can be seen when looking at a finer grain of number, in particular the effects of the other personality components, are lost. That being said, the derivation and error propagation are linear thanks to the equal sample size the component statistics.

Table 5 tell us that Perceiving performs the best of the personality components, and Judging the worst. This confirms the idea that, contrary to expectations, the $Judging \longleftrightarrow Perceiving$ axis is more important than the $Thinking \longleftrightarrow Feeling$ axis.

5.4 Prompt Sentiments

The effects of prompt engineering can arise from spurious properties of the prompts [8]. The first place to look is the sentiment of each of the personality prompts. We utilize the sentiment analysis tools provided in the Python Natural Language Toolkit (NLTK) for this analysis.

We start with the polarity scores for the personality prompts. Compound sentiment polarity measures how positive or negative a text "sounds." A higher number indicates that the text seems more positive. The scores are given in the table below in descending order:

Table 6: Compound Sentiment Polarity Scores for the Personalities and Personality Components

Personality	**Compund Sentiment Polarity**
FP	0.9224
FJ	0.8748
TP	0.775
TJ	0.5413
F	0.8439
T	0.3804
P	0.3552
J	0.2263

The best-performing personality, p_{FP}, also has the most positive sentiment. The worst-performing personality (p_{TJ}) also has the worst sentiment. Judging does the worst on both sentiment and gameplay effect. This is weak evidence that the sentiment of the personalities could have an effect on the performance of the agents.

For the personality components, results are more ambiguous. J is the component with the worst sentiment and also with the worst performance, but no other connections occur. The derived nature of the average advantages of the personality components could contribute to this lack of clear connection between the personality component advantages and their sentiments.

6 Analysis

Agents did not exhibit strong high-level strategic abilities. They didn't collaborate among each other, nor did they use deductive reasoning; merely a simulacrum of it.

There was no point at which a villager, for example, put together that either one of players A or B are a werewolf, but player B cannot be a werewolf due to earlier behavior, so player A must be. This is all to say that they never performed true logic.

Furthermore, the agents respected the boundaries of the prompt too carefully. Even the agents prompted with the Thinking personality didn't use a cohesive strategy. This is especially surprising since the strategies covered in §2.2 are in their training data. The agents did nothing more or less than carry out the instructions of the prompt.

6.1 Explaining Personality Pair Effects

So stimulating rationality did not seem to work in this context. What did work? When looking for a personality that did well in the game on this chart, we should look in Fig. 5 for a cross of color along a row and a column with the same personality. A cross with a blue row and a red column indicates that the personality is advantageous to the agents possessing it; likewise, a cross with a red row and a blue column indicates that the personality is detrimental to the agents who possess it.

There aren't many such crosses in the personality matrix, showing that the personalities in general had effects that were not wholly linear. However, a few patterns do suggest themselves. p_{TJ} comes the closest to having such a cross with a rather blue column and a mostly red row. In turns out that an agent with p_{TJ} is less likely to win. This is surprising and contradicts our hypothesis (§3.1).

What happened? While we believed that telling the agents to think logically (Thinking) and act quickly upon best judgment (Judging) would improve performance, it actually harmed performance for both teams. Judging may lead the agent to overly hasty decisions. By fiercely seek closure, the agents end up refusing to update in light of new evidence, which becomes harmful as the game progresses.

Another trend we notice is that the presence of p_{FP} helps the werewolves in general, no matter which team it appears in. The only exception is $G(\mathbf{V}_{FP}, \mathbf{W}_{TJ})$, which starkly advantages the villagers. Since p_{TJ} is weak, it is reasonable to assume that its opposite, p_{FP}, is strong, and when the strongest personality is paired against the weakest it would stand to reason that the team with the strongest personality would win most often. We would therefore expect $G(\mathbf{V}_{TJ}, \mathbf{W}_{FP})$ to advantage the werewolves. Indeed it does, by a large margin (-0.28, see Fig. 5) It is the pairing second-most biased towards the werewolves among them all.

The curious thing is that, except in the most extreme case of the $p_{FP} \longleftrightarrow p_{TJ}$ pairing, p_{FP} helps werewolves more than villagers. What explains this asymmetry?

In game design, a game is asymmetric when it presents different rule structures to different (groups of) players [5]. Werewolf is an asymmetric game because the two teams have different assets and different goals. The werewolves start with two players while the villagers have five. Further, the villagers need to find out who among the seven players is a werewolf, whereas the werewolves only need to eliminate the known villagers as soon as possible, ideally the cannier ones first.

So the villagers are playing a game of deduction and reasoning to a greater extent than the werewolves. In contrast, logic matters less to the werewolves than modeling what the villagers are thinking. Modeling what other entities are thinking is known as perspective-taking [20]. If p_{FP} improves the perspective-taking ability of players, this would ameliorate the performance of the werewolves more than the villagers because perspective-taking matters more for the werewolves. The villagers benefit less from a Theory of Mind because they are more concerned with who the werewolves are than with what they are thinking.

Further work (§9) can extend these hypotheses and test them using the internal thoughts of the villagers.

We also see that p_{TP} is by far the most helpful personality for the villagers, even more helpful than p_{FP}. This fits the hypothesis that the Thinking personality component is more helpful for the villagers since the villagers need logic more.

One anomalous pairing stands out: $G(\mathbf{V}_{TJ}, \mathbf{W}_{\emptyset})$ starkly favors the werewolves, despite the werewolves having no personality. $p_{\emptyset}$ doesn't seem particularly helpful in other situations, so its strength in this situation is puzzling.

What's the explanation for the Werewolf advantage in $G(\mathbf{V}_{TJ}, \mathbf{W}_{\emptyset})$? We know that p_{TJ} hurts villagers in the majority of cases. $G(\mathbf{V}_{TJ}, \mathbf{W}_{TJ})$ and $G(\mathbf{W}_{TJ}, \mathbf{V}_{TJ})$ have little effect, as we expect in symmetric games. p_{FJ} also has little effect, consistent with the observations so far that the $T \longleftrightarrow F$ axis is least important. When the villagers have *Perceiving*, they do well against $\mathbf{W}_{TJ}$. But none of this explains why the empty personality does so well against p_{TJ} for the werewolves.

It's possible that the default behavior of the werewolves plays well against p_{TJ}, and that any additional personality hobbles werewolf behavior. This is a good example of the unpredictability of persona prompting, an attribute that is well-attested in the literature [51]. The interplay of two competing personas amplifies the chaotic results of

each.

The presence of strong personality effects is our most important result. The personalities of both agent teams helps us predict the results of each competition. While some personalities are better than others, there is no personality that completely dominates. The personalities advantage the villagers and the werewolves in different ways, but the difference is one of magnitude, not sign.

6.2 Explaining Personality Component Efficacy

Why do perceiving types do better than judging types? Perceiving types were told to "make decisions slowly and seek more information," as opposed to shutting off perception once they have perceived enough to make a decisions. The distinction Perceiving and Judging is thus analogous to the explore/exploit decision in computer science [6], where Perceiving types exploit later and Judging types exploit earlier.

Perceiving types may perform well because they collect a greater amount of data before they make a decision, making them less likely to make rash mistakes. Werewolf is a linear game where mistakes are irreversible, so exploiting later is advantageous.

The Perceiving advantage may also arise from a general characteristic of language agents, which is that telling them to think longer and contemplate before responding improves output. [43]. The Perceiving personality would then be a specific form of a prompt telling the agent to double-check its work.

Finally, humans are known to be biased towards exploiting too early in explore-exploit situations [16]. LLMs may have inherited this bias and the Perceiving personality may help them overcome it.

7 Discussion

The *Judging* $\longleftrightarrow$ *Perceiving* axis has a greater effect on strategic efficacy than the *Thinking* $\longleftrightarrow$ *Feeling* axis. This was perhaps the most surprising result from this study. The $J \longleftrightarrow P$ axis mediates how early the agent acts on information. Agents with p_P were less hasty to act upon previous information and this personality may have stimulated more premeditated thought before action.

This being said, aside from the effects we covered in §6, most personalities had weak

or ambiguous effects. For example, as previously discussed, there were few prominent crosses of color in Fig. §5. Furthermore, the effects of personalities were highly mediated by their interaction with each other.

This could be in part because the personality prompts were not sufficiently strong. The personalities in this study were adapted from the MBTI manual. It is possible that stronger language in the personalities will be required to produce a more stark effect. More specific strategic language may also need to be used.

We must further note that irrelevant attributes of the personality prompt likely had an effect on game performance, as is often the case for persona prompting [1]. We studied spurious sentiment effects in §5.4 and found that the evidence is consistent with this possibility.

7.1 Agency Lessons from Werewolf

We have analyzed the results of the personalities on gameplay from a psychological perspective. What do our results tell us about the nature of language agents?

While it is clear that persona prompting has an effect on agent behavior, these effects are unpredictable and might not related to the agents themselves. This confirms that prompts alone are an insufficient way to direct the behavior of language agents, a common refrain we see in the literature [51]. This problem is overanthropomorphization: at the neuronal level, the agent is fitting together semantic components derived from training data like logic gates, but these ideas do not necessarily correspond to actual logical propositions. In many cases, the wording of a prompt can matter more than the content the wording signifies, as in section 10 of Linsey et al. in which the LLM is forced onto a response path by grammar and the specific verbiage of a jailbreak prompt [25].

The nature of our results suggests that they arise more due to effects like this than to any actual modeling of personality in LLMs. There is no evidence that large language models have a personality that can be changed by prompting or even any other method. While prompting can affect what an LLM says, we find no evidence that it affects what an LLM does.

Persona prompting changes the performance of LLMs in social deception games, but it does not necessarily change this performance in the desired direction. It is a powerful but unreliable tool.

There are other methods of agent direction are more promising. Reinforcement

Learning (RL) has been used in many games to improve performance, and these often lead the models to take novel strategies that are both well-reasoned and highly effective [23]. Language agents can only natively support token prediction. All their behavior in a simulation is an epiphenomenon of token prediction. We believe prompting alone cannot achieve agency. Only deeper interventions such as RL stimulate truly goal-directed behavior.

The power of the type of study we demonstrate in this paper is that it gives us a quantitative method for studying how prompting affects LLM agents. Personalities are a subset of the larger field of persona prompting. They are a high-level lens through which to study agent behavior.

8 Conclusion

In this paper we build a novel testing framework for embedding language agents in the hidden information game Werewolf. With this tool, we demonstrate that persona prompting effects agent performance in a hidden-information game.

There is strong evidence that agents that are told to spend longer on their work (Perceiving rather than Judging) perform better than other agents. In addition, there is weak evidence that agents that are told to pay more attention to the social dynamic and the "emotions" of other agents (Feeling rather than Thinking) outperform those who are focused on entirely rational analysis. In particular, p_{FP} statistically outperforms p_{TJ}.

While it is tempting to connect these results to human strategies arising from human personalities, we must be wary of the dangers of anthropomorphization. It's possible that the effects of personality rely not on a mediation of personality as we understand it, but from a chaotic effect hidden somewhere in the embedding space of the models. For example, the quantitative sentiment of the Judging module is less than that of the other modules, (§5.4) indicating a more negative emotional valence. It's possible that this alone causes the model to self-sabotage and play worse in personalities that include Judging. We discussed this more in §5.4. The lack of logical thinking during the simulations further supports this explanation. It's likely that the transformer technology, which was designed originally for token prediction, will need more than prompting to result in robust, competent language agents.

Finally, we are aware of the interaction effects between the personalities of different

agents. Testing the personalities in isolation, with the other team in the "None" column, produced inconsistent results, (Fig. 5). The effects of the personalities when playing against each other are more complicated than initially expected. Future studies are needed to clarify these complex effects.

9 Future Work

Personalities have complex and unexpected effects when they come into contact. Future studies can implement a few changes in the research design to control for personality interference:

1. We can randomize the turn order of the villagers and werewolves. In the current research design, each player provides their answers in the same order (Fig. 2). Randomizing the order or, better yet, building a simulation that allows free-order voting, would make the effects we see more robust.

2. When building personalities, we can switch from the MBTI to a quantitative model, such as the Five-Factor OCEAN model, which would allow us to vary the nature of the personalities on a more fine-grained level. Continuous scales will allow us to perform regression analysis and more precisely define the interaction effects between personalities of differing intensities. The disadvantage of this is that it may be more difficult to generate prompts from.

These two interventions above would help us study in further depth what happens when two personalities interact with each other. But varying the experimental design could also accomplish much more.

In future work we also want to apply these frameworks to study open-source LLMs. In addition to improving transparency and accessibility, this would increase the amount of analysis we can do since these models are open-weight and their internal deliberation can be interpreted directly. However, this poses the risk of significantly reducing agent abilities and also does not represent the cutting edge in LLM technology.

Another improvement involves the kind of data we collect. Since these simulations produce many dimensions of data, there is much opportunity for the study of more statistics in addition to win rate, such as target accuracy for the villagers and the "strength" of the win, defined as the number of players on the winning team remaining, as well as the sentiment and other attributes of agent utterances during play.

Furthermore, improvements to the agents themselves could help better study this situation. In particular, we want to integrate improvements from Kaiya et al. [14] and Xu et al. [48] to better specialize the agents to be able to play werewolf.

Finally, we recommend rebuilding the simulation infrastructure with Autogen. Autogen has support for a greater number of multiagent structures than heirarchical and sequential. This would allow the agents to freely talk with one another, better simulating Werewolf and expanding the range of communication possibilities.

Many scientists and futurists believe that AI agents will be the most impactful use of AI in the future [18]. As they become more important in the workplace, their importance in the social life of our society will also grow. Understanding how prompting affects the behavior of the agents will soon be more than a matter of mere scientific curiosity. It will be part of navigating technological infrastructure. Work in this field helps to guide our own development of AI and lays the groundwork for future people to navigate their own technological landscape. We owe this work to our descendants.

10 References

[1] de Araujo, P. H. L., Röttger, P., Hovy, D., & Roth, B. (2025). Principled Personas: Defining and Measuring the Intended Effects of Persona Prompting on Task Performance. arXiv preprint arXiv:2508.19764.

[2] R. C. Atkinson and R. M. Shiffrin. Atkinson, Richard C., and Richard M. Shiffrin. "Human memory: A proposed system and its control processes." Psychology of learning and motivation. Vol. 2. Academic press, 1968. 89-195.

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[4] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712.

[5] Neto, A., Cardoso, P., & Carvalhais, M. (2024, November). Asymmetry Bricks: A Framework for the Design of Asymmetry in Games. In International Conference on Design and Digital Communication (pp. 987-1000). Cham: Springer Nature Switzerland.

[6] Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. ACM computing surveys (CSUR), 45(3), 1-33.

[7] Dahl, M., Magesh, V., Suzgun, M., & Ho, D. E. (2024). Large legal fictions: Profiling legal hallucinations in large language models. Journal of Legal Analysis, 16(1), 64-93.

[8] Gandhi, V., & Gandhi, S. (2025). Prompt Sentiment: The Catalyst for LLM Change. arXiv preprint arXiv:2503.13510.

[9] Goldberg, L. R. (1993). The structure of phenotypic personality traits. American psychologist, 48(1), 26.

[10] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., ... & Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges. arXiv preprint arXiv:2402.01680..

[11] Hirata, Y., Inaba, M., Takahashi, K., Toriumi, F., Osawa, H., Katagami, D., & Shinoda, K. (2016, June). Werewolf game modeling using action probabilities based on play log analysis. In International Conference on Computers and Games (pp. 103-114). Cham: Springer International Publishing.

[12] Jinxin, S., Jiabao, Z., Yilei, W., Xingjiao, W., Jiawen, L., & Liang, H. (2023). Cgmi: Configurable general multi-agent interaction framework. arXiv preprint arXiv:2308.12503.

[13] Jones, C., & Bergen, B. (2024, June). Does GPT-4 pass the Turing test?. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 5183-5210).

[14] Kaiya, Z., Naim, M., Kondic, J., Cortes, M., Ge, J., Luo, S., ... & Ahn, A. (2023). Lyfe agents: Generative agents for low-cost real-time social interactions. arXiv preprint arXiv:2310.02172.

[15] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. arXiv preprint arXiv:2001.08361.

[16] Kasper, J., Fiedler, K., Kutzner, F., & Harris, C. (2023). On the role of exploitation and exploration strategies in the maintenance of cognitive biases: Beyond the pursuit of instrumental rewards. Memory & Cognition, 51(6), 1374-1387.

[17] King, S. P., & Mason, B. A. (2020). Myers-Briggs type indicator. The Wiley encyclopedia of personality and individual differences: Measurement and assessment, 315-319.

[18] Daniel Kokotajlo, Scott Alexander et. al. (2025). AI 2027. https://ai-2027.com/

[19] Kovač, G., Portelas, R., Dominey, P. F., & Oudeyer, P. Y. (2023). The socialAI school: Insights from developmental psychology towards artificial socio-cultural agents. arXiv preprint arXiv:2307.07871.

[20] Labash, A., Aru, J., Matiisen, T., Tampuu, A., & Vicente, R. (2020). Perspective taking in deep reinforcement learning agents. Frontiers in Computational Neuroscience, 14, 69.

[21] Latif, E., Zhou, Y., Guo, S., Gao, Y., Shi, L., Nayaaba, M., ... & Zhai, X. (2024). A systematic assessment of openai o1-preview for higher order thinking in education. arXiv preprint arXiv:2410.21287.

[22] Li, S., Yang, J., & Zhao, K. (2023). Are you in a masquerade? exploring the behavior and impact of large language model driven social bots in online social networks. arXiv preprint arXiv:2307.10337.

[23] Li, Z., Ni, Y., Qi, R., Jiang, L., Lu, C., Xu, X., ... & Zhang, X. (2024). Llm-pysc2: Starcraft ii learning environment for large language models. arXiv preprint arXiv:2411.05348.

[24] Lim, S., Lee, S., Min, D., & Yu, Y. (2025). Persona Dynamics: Unveiling the Impact of Personality Traits on Agents in Text-Based Games. arXiv preprint arXiv:2504.06868.

[25] Lindsey, J., Gurnee, W., Ameisen, E., Chen, B., Pearce, A., Turner, N. L., ... & Batson, J. (2025). On the biology of a large language model (2025). Transformer Circuits Thread https://transformer-circuits. pub/2025/attribution-graphs/biology. html.

[26] Richard E. Mayer. (2020). Multimedia Learning (3rd. ed.) Chapter 1. Cambridge Univ. Press, Cambridge, UK.

[27] Isabel Briggs Myers, Mary H. McCaulley, et. al. (1998). MBTI Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator. (3rd ed.) Consulting Psychologists Press, Palo Alto, CA.

[28] Nascimento, N., Alencar, P., & Cowan, D. (2023, September). Self-adaptive large language model (llm)-based multiagent systems. In 2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C) (pp. 104-109). IEEE.

[29] Nuxoll, A. M., & Laird, J. E. (2012). Enhancing intelligent agents with episodic memory. Cognitive Systems Research, 17, 34-48.

[30] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

[31] Osawa, H., Otsuki, T., Aranha, C., & Toriumi, F. (2019, August). Negotiation in hidden identity: designing protocol for werewolf game. In International Workshop on Agent-Based Complex Automated Negotiation (pp. 87-102). Singapore: Springer Singapore.

[32] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023, October). Generative agents: Interactive simulacra of human behavior. In Proceedings of the 36th annual acm symposium on user interface software and technology (pp. 1-22).

[33] Pittenger, D. J. (1993). Measuring the MBTI... and coming up short. Journal of Career Planning and Employment, 54(1), 48-52.

[34] Randall, K., Isaacson, M., & Ciro, C. (2017). Validity and reliability of the Myers-Briggs Personality Type Indicator: A systematic review and meta-analysis. Journal of Best Practices in Health Professions Diversity, 10(1), 1-27.

[35] Ri, H., Kang, X., Khalid, M. N. A., & Iida, H. (2022). The dynamics of minority versus majority behaviors: a case study of the mafia game. Information, 13(3), 134.

[36] Shibata, H., Miki, S., & Nakamura, Y. (2023). Playing the Werewolf game with artificial intelligence for language understanding. arXiv preprint arXiv:2302.10646.

[37] Soto, C. J., & Jackson, J. J. (2013). Five-factor model of personality. Journal of Research in Personality, 42, 1285-1302.

[38] Spitale, G., Biller-Andorno, N., & Germani, F. (2023). AI model GPT-3 (dis) informs us better than humans. Science Advances, 9(26).

[39] Papadopoulos, R. K. (Ed.). (2006). The handbook of Jungian psychology: Theory, practice and applications. Psychology Press.

[40] Tsunoda, I., & Kano, Y. (2019, October). AI werewolf agent with reasoning using role patterns and heuristics. In Proceedings of the 1st International Workshop of AI Werewolf and Dialog System (AIWolfDial2019) (pp. 15-19).

[41] Uiterwijk, J. W. H. M., & van den Herik, H. J. (2000). The advantage of the initiative. Information Sciences, 122(1), 43-58.

[42] Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., ... & Wen, J. (2024). A survey on large language model based autonomous agents. Frontiers of Computer Science, 18(6), 186345.

[43] Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., ... & Resnik, P. (2024). The prompt report: a systematic survey of prompt engineering techniques. arXiv preprint arXiv:2406.06608.

[44] Wang, S. T. (2024). Optimal Strategy in Werewolf Game: A Game Theoretic Perspective. arXiv preprint arXiv:2408.17177.

[45] Wang, Y., & Laird, J. E. (2006). Integrating semantic memory into a cognitive architecture. Ann Arbor, MI: University of Michigan Center for Cognitive Architecture.

[46] WikiHow. (2024)(. How to Play the Werewolf Card Game with Your Friends. Retrieved from https://www.wikihow.com/Play-Werewolf-(Party-Game)

[47] Wu, K., Wu, E., Cassasola, A., Zhang, A., Wei, K., Nguyen, T., ... & Zou, J. (2024). How well do LLMs cite relevant medical references? An evaluation framework and analyses. arXiv preprint arXiv:2402.02008.

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

[49] Ye, H., Liu, T., Zhang, A., Hua, W., & Jia, W. (2023). Cognitive mirage: A review of hallucinations in large language models. arXiv preprint arXiv:2309.06794.

[50] White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., ... & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382.

[51] Zheng, M., Pei, J., Logeswaran, L., Lee, M., & Jurgens, D. (2024, November). When" a helpful assistant" is not really helpful: Personas in system prompts do not improve performances of large language models. In Findings of the Association for Computational Linguistics: EMNLP 2024 (pp. 15126-15154).

11 Appendix

AI Disclosure: Artificial Intelligence tools were used to help in the creation of this paper. GPT-5 performed certain boilerplate coding tasks. Overleaf's built-in AI was used for light manuscript editing.

A Full Data

These tables represent the results for all personalities played against each other. The Personality Efect is calculated from the villager win rate minus 0.63, the theoretical win rate without personalities (§5.1).

Highlighted rows are statistically significant with $p \leq 0.05$

Table 7: TP Performance

Villager Personality	Werewolf Personality	Personality Effect	Villager Win Rate
TP	None	+0.12	0.75
TP	TJ	+0.23	0.865
TP	FJ	+0.17	0.8
TP	FP	-0.03	0.6
None	TP	0.00	0.675
TJ	TP	-0.13	0.5
FJ	TP	-0.03	0.6
FP	TP	-0.11	0.525
TP	TP	+0.12	0.75

Table 8: FP Performance

Villager Personality	Werewolf Personality	Personality Effect	Villager Win Rate
FP	None	+0.02	0.65
FP	TJ	+0.295	0.925
FP	FJ	-0.08	0.55
FP	TP	-0.11	0.525
None	FP	-0.23	0.4
FJ	FP	-0.18	0.45
TJ	FP	-0.28	0.35
TP	FP	-0.03	0.6
FP	FP	-0.205	0.425

=

Table 9: TJ Performance

Villager Personality	Werewolf Personality	Personality Effect	Villager Win Rate
TJ	None	-0.50	0.625
TJ	TP	-0.13	0.5
TJ	FJ	0.00	0.625
TJ	FP	-0.28	0.35
None	TJ	+0.12	0.75
TP	TJ	+0.23	0.865
FJ	TJ	+0.02	0.65
FP	TJ	+0.295	0.925
TJ	TJ	+0.02	0.65

Table 10: FJ Performance

Villager Personality	Werewolf Personality	Personality Effect	Villager Win Rate
FJ	None	+0.07	0.7
FJ	TJ	+0.02	0.65
FJ	FP	-0.18	0.45
FJ	TP	-0.03	0.6
None	FJ	+0.12	0.75
TJ	FJ	0.00	0.625
TP	FJ	+0.17	0.8
FP	FJ	-0.08	0.55
FJ	FJ	+0.07	0.70