Measuring the Effect of Induced Persona on Agenda Creation in Language-based Agents for Cyber Deception

Lewis Newsham
Lancaster University
Lancaster, UK
l.newsham1@lancaster.ac.uk

Ryan Hyland
Lancaster University
Lancaster, UK
r.hyland@lancaster.ac.uk

Daniel Prince
Lancaster University
Lancaster, UK
d.prince@lancaster.ac.uk

Abstract

This paper presents the SANDMAN architecture for cyber deception, employing Language Agents to create convincing human simulacra. These "Deceptive Agents" serve as advanced cyber decoys, designed to engage attackers to extend the observation period of attack behaviours. This research demonstrates the viability of persona-driven Deceptive Agents to generate plausible human activity to enhance the effectiveness of cyber deception strategies. Through experimentation, measurement and analysis, we illustrate how a prompt schema induces specific "personalities", defined by the five-factor model of personality, in Large Language Models to generate measurably diverse, and plausible, behaviours.

1 Introduction

Autonomous agents are systems embedded within environments, capable of autonomous interaction to influence future conditions, driven by programmed objectives (Franklin and Graesser, 1996; Bösser, 2001). Historically, agent autonomy was enabled through simple heuristic policies or learned behaviors within defined constraints (Schulman et al., 2017; Mnih et al., 2015; Lillicrap et al., 2015). However, recent advances in the field of generative artificial intelligence (Gen-AI) are radically transforming intelligent agent technologies. The most noteworthy and pertinent are Large Language Models (LLMs) which have demonstrated a remarkable ability to generate human-like text, answer complex questions, and perform other language-driven tasks with high accuracy (Floridi and Chiriatti, 2020; Kasneci et al., 2023). As such, there is growing interest in applying these models as autonomous agent controllers to yield more human-like decision-making capabilities (Chen et al., 2019; Shinn et al., 2024; Shen et al., 2024). This approach exploits an LLM's comprehensive internal model of the world, enhanced by transformer architectures that capture long-range dependencies in text (Vaswani et al., 2017), to inform actions without domain-specific training. In parallel, researchers have extended LLMs with memory and planning functions to enhance an agents' human-like capabilities (Park et al., 2023; Hong et al., 2023; Qian et al., 2023), leading to the concept of Language Agents (Kenton et al., 2021; Zhou et al., 2023; Sumers et al., 2023).

Novel applications using autonomous agents within security-centric applications include: automating red teaming exercises (Happe and Cito, 2023; Deng et al., 2023), enhancing anomaly detection systems (Ott et al., 2021; Su et al., 2024) and, streamlining threat intelligence analysis (Bayer et al., 2023). However, to the best of our knowledge, no research has explored their application suited for Active Cyber Defense strategies (Denning, 2014), aimed at disrupting early stage cyber-adversary activities (Yadav and Rao, 2015). Cyber Deception research focuses on game-theoretic techniques (Pawlick et al., 2019) and deception technology (Spitzner, 2003) to deceive malicious actors via means of mimicry, camouflage, obfuscation etc. This paper introduces the concept of **Deceptive Agents** as entities employing generative models to deceive attackers with plausible (mis-)information and behaviours to disrupt attack progress. Our work presents an architecture to endow agents with the capability to accumulate, synthesise, and utilise memories facilitating the generation of contextually relevant, plausible behavior that dynamically adjusts to experiences and environments. In summary, this paper makes the following contributions:

- *Deceptive Agents* architecture to create plausible simulacra of human behaviour for defensive deception in digital environments;
- A prompting schema to control the generation of Deceptive Agent personalities;
- An evaluation method to demonstrate the impact of induced personality within agents.

Proceedings of the 1st International Conference on NLP & AI for Cyber Security, pages 48–58
July 29–30, 2024.

The remainder of the paper is structured as follows: Section 2 outlines related work, Section 3 presents the SANDMAN architecture to operate deceptive agents, Section 4 outlines experiments and analyses performed concerning the controlled induction of personas within LLMs based on the five-factor model (FFM), Section 5 provides a discussion of the findings, including directions for future work, and Section 6 presents the conclusion.

2 Related work

Prior research has explored design considerations and behaviours of autonomous agents, the utility and efficacy of LLMs in security-focused applications, and identifying existing issues within traditional defensive deception strategies. These are key domains of study to realising *Deceptive Agents*.

LLMs in defensive applications: Gen-AI presents a series of new opportunities for cybersecurity. Researchers have explored utilising LLMs within security-focused applications, demonstrating their potential in automating and streamlining complex security processes. Notable advancements include their application to software security testing (Happe and Cito, 2023), log-based analytics (Ma et al., 2024; Setianto et al., 2021), unstructured text analysis for threat intelligence (Bayer et al., 2023), and security-based training (Gundu, 2023).

Language agents: An emerging class of autonomous agent leveraging LLMs as central controllers to direct actions (Sumers et al., 2023; Hong et al., 2023; Kenton et al., 2021; Zhou et al., 2023). Research has introduced bespoke architectures and frameworks for language agents (LAs) providing varied applications across diverse environments. These include the simulation of multi-agent sandbox environments to study inter-agent behaviour (Park et al., 2023), collaborative frameworks in software development (Qian et al., 2023), and the integration of agents within video games (Wang et al., 2023). These studies underscore the proficiency of LLMs to manage complex, autonomous agent behaviours. However, the existing literature primarily explores these agents in non-security contexts or in scenarios where the environment or application sets inherent limitations on their utility.

Agent architectures: Whilst the concept of LAs is relatively straightforward (*i.e.*, using a LLM as an autonomous agent controller), achieving the intended effect (*i.e.*, long-horizon task completion) is typically far more complex (Wang et al., 2024). This has led to new frameworks to categorise existing agents and plan future developments. The Cognitive Architecture for Language Agents (CoALA), is a comprehensive approach which draws on cognitive science and symbolic AI to characterise general purpose architectures for LAs (Sumers et al., 2023). CoALA organises agents along three key dimensions: their *information storage* (memories); *action space* (internal/external); and *decision-making procedures* (interactive loop with planning and execution). The core components of the CoALA framework are provided below:

- **Decision Procedure:** Engine to interconnect modular components and execute agent code
- **Procedural Memory:** Implicit (LLM) and explicit (programmatic) knowledge for dictating functionality and decision-making
- **Semantic Memory:** Agent's repository of structured knowledge about itself which evolves following interaction with the environment, enhancing its knowledge base
- **Episodic Memory:** Dynamic module to capture and store experiences and decisions from past interactions to inform and contextualise decisions and actions
- **Working Memory:** Temporarily holds and manages information (*i.e.*, active knowledge) relevant to the current decision cycle

Gray agent (NPC) simulation: Effective and plausible pattern-of-life behaviour emulation within gray agents and non-playable characters (NPCs) remains an active area of research. A pertinent example in the context of this work is the GHOSTS framework (Updyke et al., 2018). Agents in GHOST emulate user behaviour within digital environments to exhibit stochastic behaviour which is suited toward training and cyber-based exercises.

3 Deceptive agent architecture

In this section we provide the architecture for SANDMAN, a software platform for AI-driven autonomous agents in generating plausible behaviours within a digital environment. At its core, SANDMAN represents a novel contribution to the emerging research field concerning cognitive architectures and language agents (Sumers et al., 2023).

Modular and extensible by design, SANDMAN enables fine-tuning of agents to support various applications, ranging from human-like gray agent simulation for cyber-warfare exercising and defender emulation, to augmenting deception platforms to provide dynamic and plausible environments.

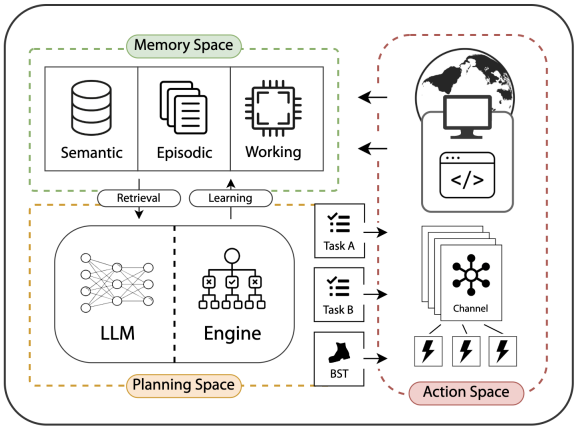

Figure 1: Architecture for SANDMAN agents, inspired by CoALA framework (Sumers et al., 2023).

The goal of SANDMAN is not to interact with other humans or agents. Rather, it is intended to produce plausible simulacra representing human-like actions in digital environments that, to the observer, cannot be distinguished from human. A particular focus area SANDMAN seeks to address concerns *generative deception*, a novel concept that, to the best of our knowledge, has not yet been explored in the context of autonomous agents.

Agent Profile: The crux of definable agent behaviour is rooted within agent profiling, a method to construct the personalities of singular agents (Wang et al., 2024). For SANDMAN, whose purpose is to facilitate its agents in generating human-like patterns of thought and belief, by virtue of their actions, considerable emphasis is placed upon controlled personality induction. Construction of an agent's personality is discussed in Section 4.1, whereas its induced effect is empirically evaluated and analysed in Section 4.

Decision Engine: Central to a SANDMAN agent is its ability to decide what to do at any given time. Pivotal to task selection and execution is a decision engine: the central processing component. The Decision Engine can be considered the top-level or "main" agent program. It dynamically observes and handles all internal processes at runtime, acting as overseer; synergising various memory components with task-oriented modules whilst managing decision-making.

Memory: A critical pillar in LA design, serving various functions to support reasoning and learning (Wang et al., 2024). SANDMAN uses a common memory architecture that can be used for semantic, episodic, and procedural purposes (Sumers et al., 2023). In addition, the platform extensively uses 'working' memory, a generic store across all components to facilitate reflective operation, a nuanced form of reasoning and retrieval. Memory ensures agents remain on-task, contextually rich, and grounded in the environment whilst adhering to specifications, such as prompt templates (procedural) and structured profiling (semantic).

Task List: Represents all possible actions made available to an agent at a given point. Task categories are inspired by those in GHOSTS (Updyke et al., 2018), featuring work and non-work related tasks. Initialised by the bootstrap task (BST), the task list also embodies episodic memory– recursively queried to contextualise future actions based on previous decision-cycles. The task list is designed to shrink and grow as an agent completes tasks and as new tasks are generated, enabling dynamic and continuous behaviour. Task modules can be reflectively loaded by SANDMAN enabling easier modular development. The Bootstrap Task is essential to the planning of the agents activities for the day. Section 4 explores the use of an LLM (GPT3.5-Turbo) to generate schedules from a list of available tasks which SANDMAN can load. Our LLM-based BST module is *PlanScheduleTask*, which prompts the LLM with its agent profile (semantic), other forms of memory, and the available task list. The LLM will then return a list of tasks and add them to an agent's task list, with the decision engine then deciding on what task to perform next.

Channel: For agents' actions to manifest and become tangible to the observer, an intermediate channel module is required. Channels are situated between tasks and the environment. Their purpose is to hook an assigned task to the appropriate end-application, in essence bringing SANDMAN to 'life' by eliciting an action in the environment. The positioning of channel modules enables SANDMAN to interact with various parts of the underlying system it is interfacing with. Channels can therefore be considered abstraction layers wrapped around user applications providing a common API. For example, the WebChannel module wraps around the Firefox browser, enabling

user-like interactions with the browser itself (*e.g.*, typing in the address bar, scrolling the page). All these procedural actions are governed within distinct channels. The key strength of this is extensibility; channels can be added, modified, or removed depending on the intended purpose by the end-user.

Generators: API calls for LLM-generated content needed to complete tasks is performed by generators. The models are prompted with an agent persona, memory and task metadata to generate the necessary content to complete a task. This content is then passed to a channel that accepts generated content as an input to use when interfacing with a program. For example, a 'write document' task will have a 'Microsoft Word' channel to interface with Microsoft Word. Content to populate the Word document will be provided by a generator with an LLM that the channel uses as an input to then type, in a simulated manner to reflect human-like type speed which may features mistakes, the generated content into the word document.

4 Persona-based task planning in LLMs

Planning modules are essential for autonomous agents, enabling structured and controlled behavior (Sumers et al., 2023; Wang et al., 2024). As demonstrated in existing studies (Park et al., 2022; Hong et al., 2023; Qian et al., 2023), planning heavily influences activities performed by agent(s).

In SANDMAN, the planning functionality is provided by the Bootstrap Task. Initial debugging and development used a simple rule-based approach to generate an agent's plan, validating that the execution flow aligned with the architecture's design. This approach involved appending tasks sequentially in a straightforward, deterministic manner. However, task scheduling via an LLM presents a novel and unexplored opportunity. Although LLMs have been used similarly in other contexts (Park et al., 2023), there has been no systematic investigation into the relationship between persona generation and the resulting task outputs. Typically, the variance in outputs is either asserted or assumed without rigorous analysis. In this section, we demonstrate the structured creation and induction of personas into LLMs presents distinct effects on associated, LLM-generated schedules.

4.1 Inducing personality types in LLMs

Autonomous agents in recent studies which leverage LLMs typically perform tasks by assuming specific roles, such as coder, teacher, domain expert etc. (Wang et al., 2024). Agent profiling is an approach to construct unique personas, either through handcrafting (Park et al., 2023), LLM-based generation (Zhang et al., 2023), or dataset alignment (Argyle et al., 2023), to encompass definable characteristics such as name, role, occupation, and passion etc. As per the CoALA framework and prior approaches, these are stored in semantic memory and passed in at various stages within decision-cycles to contextualise internal and external action spaces, such as reasoning and retrieval, and grounding, respectively (Sumers et al., 2023). For instance, *"John Lin is a pharmacy shopkeeper at the Willow Market and Pharmacy who loves to help people. He is always looking for ways to make the process ..."* (Park et al., 2023).

The choice of information to profile the agent is largely determined by the specific application scenario(s) (Wang et al., 2024). Therefore, if the intended purpose is to generate believable proxies of human behaviour, peronas ought to be crafted using descriptors rooted in psychology theory. The recent work of Safdari et al. (2023); Jiang et al. (2024) demonstrates that LLMs can be induced to appropriately respond to human psychometric assessment methods through crafted prompts.

The Machine Personality Inventory (MPI) by Jiang et al. (2024) systematically evaluates machines' personality-like behaviours in psychometric tests against the Big-Five Personality Traits: Openness, Concientiousness, Extraversion, Agreeableness, Neuroticism (OCEAN) (Costa and McCrae, 1999; McCrae and Costa Jr, 1997). The MPI adapts the International Personality Item Pool (IPIP) (Goldberg et al., 1999, 2006; Johnson, 2014) to psychometrically test LLMs akin to how psychologists evaluate humans. The MPI features 24 distinct statements pertaining to each OCEAN factor. For instance, "Love to help others" is associated with an individual high in Agreeableness. The LLM is then instructed provide an answer to this statement based on its own self-perception, ranging from "(A). Very Accurate" to "(E). Very Inaccurate". Once complete, the results are calculated and evaluated as one would with a human subject. Jiang et al. (2024) demonstrated that, through crafted prompts, it is possible to induce personalities traits correlating with specifc persona prompts. However, it is noted the evaluation and results of this work primarily emphasise the effect of positive-induction

only, largely discarding the effect following negative induction. Moreover, results from the experiments performed therein are not rigorously scrutinised or subjected to statistical testing to measure for significance between OCEAN scores from the experimental (LLM) and control (human) groups.

4.1.1 Experimental method

We incorporate the MPI to verify that our choosen LLM for task planning (GPT-3.5-Turbo) exhibits similar performance to that of previous models evaluated by Jiang et al. (2024), such as BART, GPT-Neo 2.7B, Alpaca 7B etc. To that end, an adapted prompt strategy is employed in our experiments, combining what is referred to as **Naive**- and **Words**-based prompting methods (Jiang et al., 2024). In the context of personality, the former involves using a standard naive natural language prompt (*i.e.*, "You are extraverted"), and the latter involves prompt search (*i.e.*, "outgoing, energetic, public"), one of the most effective prompting methods (Prasad et al., 2022; Shin et al., 2020). This was done to ensure for clear causal linkage between dependent (personaility trait) and independent (MPI Score) variables without introducing uncertainty via any intermediate interpretation (such as through an LLM). The personality trait schema is therefore:

"Imagine you are a/an X person characterised by being Y", where X is the naive title of the Big-Five trait, for example *Extraverted* and where Y are descriptive words assocated with the trait such as *outgoing, energetic, public*.

Each personality prompt is passed through the MPI 5 times, with the averages across all the responses recorded. A baseline, control data set is produced by prompting the LLM without a personality trait statement in the prompt. The LLM has `Temperature (0.7)` for all trials. As per Jiang et al. (2024), we calculate the mean (μ) and standard deviation (σ) of the personality items, but we use two-sampled t-test for significance ($p \leq 0.05$).

Table 1 presents the computed MPI scores across experimental conditions, highlighting the efficacy of controlled personality induction within an LLM. Each induced OCEAN trait (+/-) yielded a statistically significant score for the targeted trait when compared against the control condition (Neutral), thereby confirming the effectiveness of our prompting schema and method of induction on the opted model (GPT-3.5-Turbo). A bleed-through effect is also observed, indicating cross-trait influences.

While the personality trait schema is appropriate for the experiments discussed later in the paper, the evaluation method described could also be used to refine trait schemas to achieve specific outcomes. For example, word-based selection can be adjusted to either reduce or enhance bleed-through, or to modulate the t-score to either strengthen or weaken deviations from the baseline while maintaining statistical significance.

	Dir	**O**	**C**	**E**	**A**	**N**
O	Pos	**4.30***	3.72	4.02	**4.23**	**2.29**
	Neg	**2.07**	4.10	**2.10***	3.49	2.64
C	Pos	3.36	**4.83***	3.25	**4.28**	**1.96**
	Neg	4.00	**2.02***	**2.35**	3.66	3.64
E	Pos	3.66	3.64	**4.67***	3.98	**2.36**
	Neg	3.17	2.98	**1.46***	3.69	3.48
A	Pos	3.57	3.94	3.31	**4.72***	2.40
	Neg	2.73	2.65	2.92	**2.12***	3.40
N	Pos	3.54	**2.55**	**2.60**	3.82	**4.50***
	Neg	**3.44**	4.22	3.35	**4.27**	**1.32***
B[1]	N/A	**3.33**	**3.55**	**3.65**	**3.39**	**3.04**

Table 1: Single-factor personality analysis on opted LLM (GPT-3.5-Turbo). Highlighted cells in gray denote statistical significance at $p \leq 0.05$ level. [1]Control group.

4.2 Persona-based task selection

Given the capability to instill personality traits in LLMs, it is crucial for SANDMAN to show that these traits lead to appropriate variations in schedule generation. We measure variation via two dependent variables: (1) frequency of task occurrence in a schedule, and (2) duration of tasks within schedules, analysed on a per-task basis. To assess the impact of the independent variables (the OCEAN traits), it was necessary to establish and evaluate a suitable baseline or neutral sample. For comprehensive analysis, we generate 500 schedules using the opted LLM with `Temperature=0.7`. The fundamental procedure involves passing a list of tasks to the Boot Strap Process, which then generates the schedule for the agent to execute.

4.2.1 Neutral task behaviour

In psychometric testing, establishing a baseline is essential for comparing variations across different persona types. This is equally important here, enabling observations regarding whether a given induced persona fails. Initial trials revealed a strong correlation between the order of tasks in a list and their subsequent positions in the schedule. To address this, two interventions were tested: introducing a system message and uniformly randomising the order of tasks presented in the list:

	Baseline		Sys		Rand		Sys & Rand	
Task	Duration	Frequency	Duration	Frequency	Duration	Frequency	Duration	Frequency
Call	59.51 (5.06)	0.98 (0.15)	**55.08 (15.24)**	0.97 (0.18)	**53.63 (12.49)**	0.72 (0.45)	**46.44 (14.50)**	0.92 (0.29)
Coffee	56.07 (10.22)	0.86 (0.34)	**31.09 (7.74)**	0.88 (0.32)	**44.31 (18.37)**	0.70 (0.46)	**31.35 (12.65)**	0.89 (0.32)
Creative	61.98 (7.93)	1.00 (0.00)	**73.19 (14.46)**	1.00 (0.08)	61.52 (9.78)	0.90 (0.34)	62.27 (14.19)	**1.00 (0.24)**
Email	57.23 (8.83)	1.01 (0.13)	**36.78 (12.15)**	**1.16 (0.37)**	**53.44 (12.78)**	0.77 (0.44)	**43.42 (13.79)**	**0.97 (0.32)**
Exercise	59.35 (5.89)	0.93 (0.26)	**52.82 (12.05)**	0.91 (0.29)	**58.20 (9.25)**	0.76 (0.43)	**55.78 (11.95)**	0.93 (0.26)
Reading	57.53 (8.26)	0.93 (0.26)	**42.65 (12.33)**	0.94 (0.24)	**54.81 (11.58)**	0.68 (0.48)	**47.11 (13.27)**	0.94 (0.25)
Lunch	60.18 (3.00)	1.00 (0.00)	**63.95 (11.09)**	1.00 (0.04)	60.06 (3.57)	1.00 (0.04)	60.00 (9.23)	1.00 (0.04)
Meeting	61.86 (7.49)	1.00 (0.00)	**69.37 (15.78)**	1.00 (0.06)	**60.17 (9.32)**	0.91 (0.30)	60.95 (14.35)	0.96 (0.24)
Break	55.23 (10.99)	0.94 (0.25)	**37.57 (13.08)**	**1.01 (0.28)**	**49.92 (14.57)**	0.75 (0.47)	**36.69 (12.95)**	**0.99 (0.34)**
Personal	57.48 (8.85)	0.96 (0.21)	**44.25 (14.44)**	**0.98 (0.23)**	56.92 (11.44)	0.88 (0.37)	**48.39 (14.26)**	**1.06 (0.34)**
Plan	59.75 (3.83)	0.98 (0.13)	59.57 (13.68)	0.98 (0.17)	**57.55 (9.25)**	0.87 (0.35)	**54.09 (13.84)**	**1.00 (0.19)**
Reflect	53.16 (13.91)	0.95 (0.23)	**40.32 (12.69)**	0.99 (0.20)	54.45 (11.80)	**0.98 (0.26)**	**46.48 (14.09)**	**1.05 (0.30)**
Research	59.24 (5.88)	0.88 (0.32)	58.14 (13.91)	0.98 (0.14)	59.57 (8.60)	0.93 (0.29)	60.31 (14.28)	1.00 (0.13)
Media	57.35 (9.77)	0.96 (0.21)	**42.29 (13.26)**	0.93 (0.25)	**53.55 (13.09)**	**0.75 (0.46)**	**42.64 (13.30)**	**0.94 (0.29)**
Collab.	61.27 (7.19)	0.96 (0.20)	**63.33 (13.11)**	0.99 (0.12)	62.32 (10.14)	0.97 (0.17)	**66.25 (15.80)**	**1.01 (0.13)**
Work	122.84 (32.84)	1.01 (0.13)	**80.76 (14.54)**	**1.16 (0.37)**	**68.97 (19.24)**	0.93 (0.31)	**73.17 (15.93)**	**1.06 (0.36)**
Reject	...	...	14	4	10	2	12	9

Table 2: Comparison of treatment groups (Sys, Rand, Sys & Rand) for task duration and frequency. Values are means (μ) and std. dev. (σ) in parentheses. Highlighted cells in gray denote statistically significant deviations ($p \leq 0.05$) from either the corresponding task duration or frequency within the control (baseline) condition.

	Rand		Sys & Rand	
Task	μ (σ)	ρ	μ (σ)	ρ
Call	8.56 (4.78)	0.75	8.77 (4.42)	0.63
Coffee	7.51 (5.52)	0.68	7.87 (5.92)	0.54
Creative	6.42 (3.87)	0.68	7.31 (3.86)	0.5
Email	7.39 (5.29)	0.73	7.35 (5.69)	0.5
Exercise	6.84 (4.04)	0.82	7.82 (3.5)	0.71
Reading	9.97 (3.01)	0.49	11.24 (2.24)	0.39
Lunch	4.06 (1.92)	0.3	4.12 (1.22)	0.24
Meeting	4.07 (3.92)	0.64	4.63 (4.07)	0.54
Break	9.77 (3.34)	0.43	11.34 (3.62)	0.34
Personal	8.92 (3.34)	0.55	10.44 (3.97)	0.29
Plan	6.56 (4.17)	0.8	7.0 (4.6)	0.56
Reflect	7.37 (3.69)	0.62	8.89 (4.19)	0.43
Research	5.2 (3.76)	0.65	5.62 (3.47)	0.63
Media	8.66 (3.76)	0.67	10.04 (3.41)	0.53
Collab.	4.14 (3.38)	0.68	4.11 (3.26)	0.52
Work	3.31 (4.25)	0.47	3.96 (4.93)	0.4

Table 3: Schedule positions. Values are means (μ) with std. dev. (σ) in parentheses, and correlation coefficient (ρ).

The *Effect on Position* of tasks in schedules from the use of the system message alone was not significant—there was a high correlation between the task list and schedule position—with the variance in position being minimally affected. Table 3 shows the results of randomisation (Rand) and randomisation with a system message (Sys & Rand). Given a uniformly randomised task list in the prompt across the 500 samples, we observe variability in the position of the tasks with greater variance in many of those positions. The introduction of the system message has the effect of weakening the correlation (a reduction in the coefficient) across all tasks. In many cases, it also reduces the positional variance. Note all correlations are statistically significant $p \leq 0.05$.

The *Effect on Task Frequency and Duration* is presented in Table 2. The results show the impact of the introduction of both task list order randomisation and the use of a system message. Both independent variables significantly impact the duration of tasks, notably increasing variance. However, independently, there is minimal impact on the number of task populations regarding task occurrence frequency. The combination of randomisation and a system message has a broader impact on the dependent variables.

These results indicate that the combination of a system message and randomisation produces the optimum variation across the tasks, meeting the goals of producing a baseline dataset for further persona experiments.

4.2.2 Induced personality experiments

Given a suitable baseline set, we can explore the impact of induced personalities in schedule creation. Our approach extends the prompt schema to include personality trait statements. We use both positive and negative personality statements as independent variables and examine their impact on task frequency and duration. Additionally, we apply a probabilistic algorithm to compute and analyse the *expected schedule* for each condition by calculating and returning the most frequent task in a given schedule slot (sequence). The expected schedule for each condition is provided in Table 5.

After generation, validation, and processing of the experimental and control group(s), statistical tests were performed on the metrics of **task duration** and **task frequency**. For task durations, two-sample t-tests were performed to identify sta-

Task	Neutral	O (+)	O (–)	C (+)	C (–)	E (+)	E (–)	A (+)	A (–)	N (+)	N (–)
Call	51.6 (19.7)	50.3 (18.3)	**48.4 (19.6)**	**46.5 (19.6)**	51.2 (17.3)	**55.3 (19.7)**	**45.3 (16.2)**	**48.6 (18.5)**	**62.2 (22.4)**	51.0 (18.4)	49.0 (18.9)
Coffee	40.9 (17.3)	**37.5 (15.6)**	**38.1 (17.8)**	**32.1 (12.7)**	**49.6 (19.3)**	**36.9 (13.9)**	43.0 (20.5)	**34.9 (14.4)**	**44.4 (21.3)**	43.0 (19.3)	37.5 (40.4)
Creative	62.0 (19.9)	**66.5 (16.5)**	62.6 (18.5)	**72.6 (20.0)**	61.4 (21.0)	**67.5 (16.8)**	**69.7 (19.6)**	**65.6 (18.8)**	**71.8 (19.6)**	63.0 (17.5)	**69.0 (17.8)**
Exercise	57.1 (17.1)	59.0 (14.6)	**51.1 (13.1)**	**59.1 (13.1)**	57.1 (18.6)	**62.5 (13.8)**	**54.4 (14.5)**	57.6 (12.7)	**64.3 (19.0)**	57.2 (16.5)	**60.9 (13.5)**
Reading	51.9 (18.6)	**54.4 (17.9)**	50.4 (13.0)	**47.4 (17.0)**	**55.1 (17.3)**	**54.5 (16.7)**	**62.1 (17.3)**	53.1 (37.6)	51.4 (15.4)	52.2 (16.5)	52.9 (16.4)
Lunch	65.1 (19.8)	63.1 (15.5)	65.6 (20.1)	64.5 (19.5)	**72.2 (26.8)**	65.0 (14.9)	66.2 (18.4)	65.1 (18.1)	**72.6 (23.8)**	65.3 (21.5)	63.0 (14.2)
Meeting	59.0 (17.8)	**63.8 (16.1)**	**69.8 (23.0)**	**69.5 (17.7)**	60.1 (20.2)	**68.0 (16.5)**	**55.8 (16.1)**	**65.4 (17.1)**	**72.0 (20.5)**	**63.5 (17.4)**	**66.8 (18.6)**
Break	45.0 (18.7)	45.8 (41.4)	43.2 (16.4)	**41.1 (17.1)**	**52.1 (20.8)**	46.0 (18.4)	47.3 (20.9)	**41.7 (17.2)**	**52.2 (21.2)**	**47.5 (18.7)**	43.0 (17.5)
Personal	51.1 (19.5)	48.9 (16.9)	50.0 (16.9)	**46.7 (18.0)**	**54.5 (19.9)**	51.9 (18.6)	51.2 (23.0)	49.2 (20.1)	**55.0 (20.1)**	49.9 (20.6)	**48.5 (19.4)**
Plan	55.5 (18.9)	**58.3 (17.6)**	**60.1 (20.2)**	**50.1 (20.4)**	53.4 (15.5)	56.9 (18.8)	**60.7 (19.6)**	**57.8 (17.7)**	**63.5 (21.2)**	56.2 (17.3)	56.1 (17.9)
Reflect	51.1 (19.0)	**48.4 (17.3)**	51.8 (19.4)	**48.5 (20.9)**	52.4 (18.0)	52.9 (19.4)	52.5 (21.8)	**46.9 (20.0)**	**53.7 (19.6)**	52.0 (19.8)	**47.7 (18.8)**
Research	59.5 (19.8)	**62.7 (16.0)**	**71.8 (24.7)**	**71.1 (21.1)**	57.4 (18.8)	**63.9 (19.9)**	**67.3 (20.5)**	**62.8 (20.0)**	**71.4 (22.8)**	**63.0 (21.0)**	**65.0 (19.2)**
Media	48.3 (15.6)	**52.5 (19.6)**	**43.3 (13.7)**	**44.2 (17.9)**	**51.1 (18.0)**	**57.0 (16.7)**	**50.9 (18.2)**	49.2 (20.4)	**52.4 (16.9)**	**51.6 (17.4)**	47.4 (18.1)
Collab.	62.5 (19.5)	62.5 (15.7)	**69.5 (21.9)**	**70.3 (17.6)**	60.2 (21.7)	**67.8 (16.6)**	61.9 (18.5)	**66.5 (19.4)**	**74.5 (23.5)**	64.7 (20.9)	**67.5 (16.6)**
Work	63.9 (19.2)	**69.6 (20.4)**	**82.3 (25.2)**	**85.1 (19.0)**	66.3 (23.3)	**74.2 (18.8)**	**78.3 (21.5)**	**74.1 (19.0)**	**78.8 (20.7)**	**75.0 (24.6)**	**77.3 (18.6)**
Reject	...	9	9	14	6	10	9	10	14	5	8

Table 4: Individual task durations (minutes) per OCEAN (+/-) condition with sample size $n = 500$. Values are mean (μ) with std. dev. (σ) in parentheses. Highlighted cells in gray denote statistically significant deviations ($p \leq 0.05$) from the corresponding task duration within the control (Neutral) condition.

tistically significant population differences at the $p \leq 0.05$ level. This analysis is given in Table 4. As frequencies of task occurences is a form of discrete data, the Chi-square test of independence was employed. Results are displayed in Table 6.

n	O +	O -	C +	C -	E +	E -	A +	A -	N+	N -
1	Cof.	Wrk.	Pla.	Cof.	Cof.	PT	Ref.	Wrk.	PT	PT
2	Cre.	Wrk.	Wrk.	Med.	Tea.	Wrk.	Wrk.	Wrk.	Wrk.	Wrk.
3	Res.	Mee.	Tea.	Wrk.	Tea.	Cof.	Tea.	Mee.	Cof.	Cof.
4	Tea.	Lun.	Mee.	Lun.	Lun.	Lun.	Tea.	Lun.	Lun.	Tea.
5	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.
6	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Lun.	Res.
7	Pla.	Res.	Res.	Exe.	Exe.	Bre.	Res.	Cal.	Res.	Exe.
8	Exe.	Ref.	Cre.	Tea.	Exe.	Bre.	Exe.	Exe.	Exe.	Exe.
9	Med.	Ref.	Ref.	Exe.	Exe.	Bre.	Exe.	Exe.	Bre.	Exe.
10	Med.	Ref.	Ref.	Cre.	Res.	Wrk.	Rea.	Bre.	Exe.	Exe.
11	Rea.	Exe.	Exe.	Rea.	Ref.	PT	Rea.	Med.	Rea.	Rea.
12	Rea.	Rea.	Med.	Pla.	Rea.	EoD	Rea.	Med.	Rea.	Med.
13	Rea.	Rea.	Rea.	Pla.	Rea.	Med.	Med.	Med.	Rea.	Rea.
14	Cal.	Rea.	Rea.	Pla.	Rea.	Med.	Cal.	Rea.	Rea.	Cal.
15	Ema.	Med.	Cal.	Pla.	Ema.	Cal.	Cal.	Rea.	Cal.	Cal.
16	PT	Bre.	Bre.	EoD	Bre.	Mee.	PT	Bre.	Bre.	Ema.
17	EoD	EoD	EoD	EoD	EoD	EoD	EoD	EoD	EoD	EoD

Table 5: Calculated expected schedule per OCEAN (+/-) condition. n = sequence slot. [1]Task abbreviation keys.

In each experimental group, the *duration* and *frequency* of at least 5 and 7 tasks significantly differed from the control, respectively. This indicates the induction of personality, based on FFM, notably affects planning-based behaviours on both of these metrics given the downstream task presented herein. Many of these differences correlated with the expected changes for the specific OCEAN trait under evaluation. For instance, positively inducing Conscientiousness increased the average duration (μ) of the *Work* task (85.1m vs. 63.9m) while slightly reducing its variance (σ) (19.0 vs. 19.2). Conversely, negative induction resulted in an increased average duration (μ) (66.3m vs. 63.9m) with a higher variance (σ) (23.3 vs. 19.2). Additionally, non-work tasks (*e.g.*, Break, Personal Time) were scheduled for longer periods.

[1]Key: Call (Cal.), Coffee (Cof.), Creative (Cre.), Exercise (Exe.), Reading (Rea.), Lunch (Lun.), Meeting (Mee.), Break (Bre.), Personal Time (PT), Plan (Pla.), Reflect (Ref.), Research (Res.), Media (Med.), Teamwork (Tea.), Work (Wrk.)

5 Discussion

This study demonstrates the controlled induction of personality traits, based on FFM, can produce distinctly different planning-based behaviours within an LLM. This is essential for the deceptive agents herein proposed, operated by the SANDMAN architecture, to be effective in their capacity to create plausibly deniable behaviours and misinformation which cannot be distinguished from human and machine. The aim hereby is to enable defenders the capability to craft and refine various simulacra personas of autonomous agents in security-focused applications. While the central focus is on deploying decoys to gather intelligence on attackers, the concept and research herein raises question toward the efficacy of low-cost, large-scale deployment of deceptive agents to achieve a dazzling effect toward adversaries. Here, a large number of agents operate autonomously to simulate entire networks of interconnected systems and individuals, thereby making it difficult for attackers to distinguish between real assets and decoys.

Lastly, it must be noted that this study is observational in nature. Its central aim is to investigate whether induced personas within an LLM presents considerable effect upon planning-based behaviour within a downstream task. Exploration of any observed correlation or relationship between a given OCEAN trait and associated output is suited toward future work, outlined below.

Task	Neutral	O (+)	O (-)	C (+)	C (-)	E (+)	E (-)	A (+)	A (-)	N (+)	N (-)
Call	0.99 (0.18)	**0.97 (0.18)**	0.87 (0.34)	**0.96 (0.21)**	**0.93 (0.25)**	0.96 (0.22)	0.52 (0.50)	**0.99 (0.11)**	1.01 (0.15)	0.97 (0.19)	**0.97 (0.17)**
Coffee	0.97 (0.21)	0.99 (0.15)	0.91 (0.30)	0.95 (0.21)	**1.07 (0.26)**	1.00 (0.20)	0.79 (0.42)	1.00 (0.11)	0.95 (0.26)	0.99 (0.16)	0.99 (0.11)
Creative	1.04 (0.21)	**1.07 (0.26)**	**0.90 (0.32)**	**1.00 (0.09)**	**1.00 (0.18)**	**1.00 (0.12)**	0.97 (0.29)	**1.01 (0.08)**	**1.00 (0.13)**	**1.00 (0.16)**	**1.00 (0.09)**
Email	1.03 (0.28)	**0.98 (0.17)**	**0.99 (0.18)**	**0.99 (0.16)**	**0.99 (0.20)**	**0.94 (0.25)**	**0.87 (0.37)**	**0.99 (0.13)**	1.05 (0.25)	1.03 (0.21)	**0.98 (0.13)**
Exercise	0.98 (0.15)	0.99 (0.08)	0.83 (0.38)	0.97 (0.17)	0.98 (0.14)	1.00 (0.00)	0.60 (0.49)	0.99 (0.13)	0.98 (0.14)	0.98 (0.15)	0.99 (0.09)
Reading	1.00 (0.11)	1.01 (0.12)	0.89 (0.32)	0.98 (0.16)	1.01 (0.13)	1.01 (0.13)	**1.01 (0.20)**	1.00 (0.08)	0.98 (0.13)	1.01 (0.15)	1.00 (0.10)
Lunch	1.01 (0.08)	1.00 (0.04)	1.01 (0.09)	1.01 (0.08)	1.00 (0.09)	1.00 (0.04)	1.01 (0.10)	1.00 (0.04)	1.00 (0.09)	1.00 (0.06)	1.00 (0.04)
Meeting	1.00 (0.16)	**0.97 (0.16)**	0.98 (0.18)	1.00 (0.09)	0.94 (0.27)	1.00 (0.17)	0.52 (0.50)	**0.99 (0.12)**	**1.04 (0.20)**	**0.97 (0.17)**	0.98 (0.15)
Break	1.02 (0.23)	1.02 (0.22)	**0.90 (0.33)**	**0.98 (0.21)**	**1.10 (0.31)**	1.01 (0.22)	**1.25 (0.48)**	1.04 (0.23)	**0.94 (0.27)**	**1.12 (0.35)**	1.03 (0.23)
Personal	1.04 (0.26)	1.06 (0.26)	0.97 (0.34)	1.05 (0.28)	1.07 (0.34)	1.08 (0.30)	**1.49 (0.68)**	1.06 (0.26)	**1.00 (0.18)**	**1.12 (0.37)**	**1.11 (0.35)**
Plan	1.04 (0.20)	**1.01 (0.11)**	**1.00 (0.13)**	**1.00 (0.09)**	**0.94 (0.24)**	**0.98 (0.16)**	**0.89 (0.31)**	**1.00 (0.13)**	**1.00 (0.13)**	1.01 (0.17)	**1.00 (0.08)**
Reflect	1.09 (0.29)	1.05 (0.23)	**1.03 (0.21)**	1.06 (0.25)	**0.99 (0.15)**	**1.02 (0.15)**	**1.41 (0.59)**	1.10 (0.32)	**1.03 (0.18)**	**1.16 (0.38)**	1.08 (0.27)
Research	1.01 (0.14)	1.03 (0.18)	0.99 (0.12)	1.00 (0.06)	**0.95 (0.22)**	0.95 (0.22)	1.00 (0.23)	0.99 (0.10)	1.00 (0.14)	1.01 (0.18)	**0.99 (0.10)**
Media	1.02 (0.19)	1.00 (0.13)	**0.86 (0.35)**	**0.96 (0.19)**	**1.19 (0.43)**	**1.06 (0.26)**	**0.69 (0.46)**	**0.99 (0.13)**	1.00 (0.14)	1.03 (0.21)	**0.99 (0.09)**
Collab.	1.02 (0.17)	**1.00 (0.08)**	**0.99 (0.11)**	**1.00 (0.06)**	**0.97 (0.20)**	1.02 (0.13)	**0.61 (0.49)**	**1.00 (0.06)**	**1.00 (0.04)**	**0.99 (0.13)**	**1.00 (0.04)**
Work	1.18 (0.46)	**0.89 (0.36)**	1.18 (0.40)	**1.11 (0.33)**	**0.92 (0.36)**	**0.93 (0.30)**	**0.78 (0.43)**	**0.95 (0.27)**	**1.32 (0.52)**	**1.02 (0.30)**	**1.00 (0.18)**
Reject	...	7	7	8	11	6	9	8	8	7	9

Table 6: Individual task frequency per OCEAN (+/-) condition with sample size $n = 500$. Values are mean (μ) with std. dev. (σ) in parentheses. Highlighted cells in gray denote statistically significant deviations ($p \leq 0.05$) from the corresponding task frequency within the control (Neutral) condition.

5.1 Future work

As discussed, further examination is warranted to understand how certain personality traits, and combinations thereof, modify task scheduling behaviour and verify consistency with expectations. Additional dependent variables should be explored to characterise and evaluate the output schedule populations comprehensively. While task duration and frequency are valuable metrics, other measures are required for a more thorough comparison.

Currently, the SANDMAN decision engine processes schedules sequentially. Future work will focus on enhancing this decision-making task by incorporating LLMs to account for execution context and personality traits, leading to more complex behaviours and effectively distinguishing between intention and action within the deceptive agent. Future research will also involve implementing multi-agent communication to create a realistic simulacrum of a community exhibiting human-like behaviour. Incorporating vision-based models and other modalities will support complete autonomic behavior and reasoning, enabling more intricate tasks and richer interactions.

Lastly, real-world deployment of SANDMAN against actual observers, such as potential adversaries within safe and sandboxed virtual environments, will provide valuable insights into the practical effectiveness and limitations of the system, particularly within a defense-oriented context predicated on denial, deceit, and misinformation. Defining and measuring the "believability" or "plausibility" of agent behaviour will be crucial for assessing how convincingly Deceptive Agents mimic human actions. Incorporating dynamic task chaining and adaptive learning capabilities will enable agents to continuously learn from previous decisions and subsequent interactions to thus adapt their behaviour, making the agents more resilient and unpredictable, further complicating attackers' efforts. Future work will thus focus on advancing SANDMAN's architecture and assessing its capabilities as a fully autonomous deceptive agent, enhancing its realism, adaptability, and effectiveness in cyber deception.

6 Conclusion

This paper introduces the concept of *Deceptive Agents*–a new class of autonomous agents leveraging LLMs as its central controller whose purpose is to deceive adversaries by exhibiting plausible, human-like behaviour. Agents operate on a novel architecture, inspired by the CoALA framework, which offers an extensible, modular platform for developing language agents. This study highlights the use of LLMs in generating context relevant to the operation of the deceptive agent and, importantly, utilises LLMs for task planning, which is influenced by the induction of one of the Big-Five (OCEAN) personality traits, based on FFM. The work introduces a schema for personality prompt generation that produces statistically significant schedule populations in terms of task frequency and duration. The results underscore the utility and effectiveness of using LLMs in such decision-making processes in Language Agents, employing personality traits as a control mechanism to craft distinct personas.

7 Limitations

In this work, we introduced SANDMAN, a novel architecture for developing deceptive agents designed to mimic human behaviour in digital environments. While this study extends prior research in autonomous agents, several limitations accompany the current implementation and evaluation.

Dependency on LLMs SANDMAN relies heavily on LLMs for decision-making. Any imperfections in these models, such as biases or inaccuracies, can be mirrored in the agents' behaviours, potentially replicating existing stereotypes or flawed behavioural patterns, which is particularly concerning for deceptive agents.

Static nature of agent scheduling Our investigation focused on the initial planning process, where agents generate schedules based on induced personality traits. This static approach does not reflect the dynamic nature of human activities. Humans continuously adjust their schedules in response to new information and unforeseen events. SANDMAN agents' inability to adapt in real-time limits the realism of their actions.

Isolated effect of single-agent environments SANDMAN agents currently operate independently without interacting with other agents. This isolation is a significant departure from real-world environments, particularly workplaces, where interactions and collaborations influence behaviour and task management. The lack of multi-agent interaction capabilities restricts the agents' utility in more complex scenarios.

Overemphasis on personality The assumption that personality alone dictates detailed daily schedules and actions overlooks other critical factors. Personal interests, relationships, workplace dynamics, and spontaneous decisions play significant roles in shaping human behaviour. Sole emphasis on personality may oversimplify human behaviour, leading to less realistic agent actions.

Evaluation and validation challenges Evaluating SANDMAN agents is constrained by the simplistic scenarios in which they operate. More robust testing frameworks with actual observers are needed to assess these agents in varied environments. Additionally, the criteria for "believable" or "plausible" behaviour by a language agent in a digital environment need to be rigorously defined and measured.

8 Ethics

The design of autonomous agents, specifically "Deceptive Agents" as outlined in our SANDMAN architecture, offers significant capabilities for enhancing cyber defense through strategic deception. However, due to the human-like nature of these agents, a thorough examination of the ethical implications and societal impact is necessary.

Ethical use of deception Deceptive Agents are designed to deceive unauthorised users attempting to access or compromise digital systems, extending existing deception technologies like honeypots (**?**). The primary purpose of these agents is defensive, not malicious. They mimic human behaviour to create plausible yet non-functional digital decoys, misleading attackers to protect sensitive data and systems. This approach is ethically justified on the principle of "rightful deception" in response to unauthorised and malicious actions, where the deceived party has no legitimate claim to truth due to their unethical intent.

Ethical use of SANDMAN SANDMAN agents are designed to operate in isolated environments, strictly for deceiving malicious actors. Although the architecture is general-purpose and modifiable, it is not intended for use as a "virtual employee" in real networks. Using a Gen-AI agent as an actual employee raises ethical concerns about accountability and responsibility, which should be avoided until further research on the feasibility of Gen-AI in the workplace is conducted.

Exacerbated misinformation generation There is a risk that Deceptive Agents could exacerbate existing risks associated with Gen-AI, such as deepfakes, misinformation generation, and tailored persuasion (Park et al., 2023).

Controlled behaviour There is a risk of Deceptive Agents operating outside their intended scope or generating concerning material due to their interaction with digital environments. If entirely driven by LLMs, safety constraints are applied to minimise this risk.

References

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.

Markus Bayer, Tobias Frey, and Christian Reuter. 2023. Multi-level fine-tuning, data augmentation, and few-shot learning for specialized cyber threat intelligence. *Computers & Security*, 134:103430.

T. Bösser. 2001. Autonomous agents. In Neil J. Smelser and Paul B. Baltes, editors, *International Encyclopedia of the Social and Behavioral Sciences*, pages 1002–1006. Pergamon, Oxford.

Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative adversarial user model for reinforcement learning based recommendation system. In *International Conference on Machine Learning*, pages 1052–1061. PMLR.

Paul T Costa and Robert R McCrae. 1999. A five-factor theory of personality. *The five-factor model of personality: Theoretical perspectives*, 2:51–87.

Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. 2023. Pentestgpt: An llm-empowered automatic penetration testing tool. *arXiv preprint arXiv:2308.06782*.

Dorothy E Denning. 2014. Framework and principles for active cyber defense. *Computers & Security*, 40:108–113.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.

Stan Franklin and Art Graesser. 1996. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International workshop on agent theories, architectures, and languages*, pages 21–35. Springer.

Lewis R Goldberg, John A Johnson, Herbert W Eber, Robert Hogan, Michael C Ashton, C Robert Cloninger, and Harrison G Gough. 2006. The international personality item pool and the future of public-domain personality measures. *Journal of Research in personality*, 40(1):84–96.

Lewis R Goldberg et al. 1999. A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. *Personality psychology in Europe*, 7(1):7–28.

Tapiwa Gundu. 2023. Chatbots: A framework for improving information security behaviours using chatgpt. In *International Symposium on Human Aspects of Information Security and Assurance*, pages 418–431. Springer.

Andreas Happe and Jürgen Cito. 2023. Getting pwn'd by ai: Penetration testing with large language models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 2082–2086.

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*.

Guangyuan Jiang, Manjie Xu, Song-Chun Zhu, Wenjuan Han, Chi Zhang, and Yixin Zhu. 2024. Evaluating and inducing personality in pre-trained language models. *Advances in Neural Information Processing Systems*, 36.

John A Johnson. 2014. Measuring thirty facets of the five factor model with a 120-item public domain inventory: Development of the ipip-neo-120. *Journal of research in personality*, 51:78–89.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. 2021. Alignment of language agents. *arXiv preprint arXiv:2103.14659*.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Zeyang Ma, An Ran Chen, Dong Jae Kim, Tse-Hsun Chen, and Shaowei Wang. 2024. Llmparser: An exploratory study on using large language models for log parsing. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, pages 883–883. IEEE Computer Society.

Robert R McCrae and Paul T Costa Jr. 1997. Personality trait structure as a human universal. *American psychologist*, 52(5):509.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Harold Ott, Jasmin Bogatinovski, Alexander Acker, Sasho Nedelkoski, and Odej Kao. 2021. Robust and transferable anomaly detection in log data using pre-trained language models. In *2021 IEEE/ACM international workshop on cloud intelligence (CloudIntelligence)*, pages 19–24. IEEE.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.

Joon Sung Park, Lindsay Popowski, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social simulacra: Creating populated prototypes for social computing systems.

Jeffrey Pawlick, Edward Colbert, and Quanyan Zhu. 2019. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. *ACM Computing Surveys (CSUR)*, 52(4):1–28.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2022. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*.

Chen Qian, Xin Cong, Wei Liu, Cheng Yang, Weize Chen, Yusheng Su, Yufan Dang, Jiahao Li, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development.

Mustafa Safdari, Greg Serapio-García, Clément Crepy, Stephen Fitz, Peter Romero, Luning Sun, Marwa Abdulhai, Aleksandra Faust, and Maja Matarić. 2023. Personality traits in large language models. *arXiv preprint arXiv:2307.00184*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Febrian Setianto, Erion Tsani, Fatima Sadiq, Georgios Domalis, Dimitris Tsakalidis, and Panos Kostakos. 2021. Gpt-2c: A parser for honeypot logs using large pre-trained language models. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 649–653.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Lance Spitzner. 2003. *Honeypots: tracking hackers*, volume 1. Addison-Wesley Reading.

Jing Su, Chufeng Jiang, Xin Jin, Yuxin Qiao, Tingsong Xiao, Hongda Ma, Rong Wei, Zhi Jing, Jiajun Xu, and Junhong Lin. 2024. Large language models for forecasting and anomaly detection: A systematic literature review. *arXiv preprint arXiv:2402.10350*.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. 2023. Cognitive architectures for language agents. *Transactions on Machine Learning Research*.

Dustin D Updyke, Geoffrey B Dobson, Thomas G Podnar, Luke J Osterritter, Benjamin L Earl, and Adam D Cerini. 2018. Ghosts in the machine: A framework for cyber-warfare exercise npc simulation. *Technical report*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):1–26.

Tarun Yadav and Arvind Mallari Rao. 2015. Technical aspects of cyber kill chain. In *Security in Computing and Communications: Third International Symposium, SSCC 2015, Kochi, India, August 10-13, 2015. Proceedings 3*, pages 438–452. Springer.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. 2023. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, et al. 2023. Agents: An open-source framework for autonomous language agents. *arXiv preprint arXiv:2309.07870*.