Carnegie Mellon University
Software Engineering Institute

Simulating Realistic Human Activity Using Large Language Model Directives

Dustin D. Updyke
Thomas G. Podnar
Sean A. Huff

October 2023

TECHNICAL REPORT
CMU/SEI-2023-TR-005
DOI: 10.1184/R1/24150909

CERT Division

https://www.sei.cmu.edu

Design: REV-03.18.2016.0 | Template: 01.04.2023

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100

DM23-1015

Table of Contents

List of Figures

List of Tables

Abstract

In this report, we explore how activities generated from the GHOSTS Framework's non-player character (NPC) client, including software usage, compare to activities produced by GHOSTS' default behavior and large language models (LLMs). We also explore how the underlying results compare in terms of complexity and sentiment. In our research, we leveraged the advanced natural language processing capabilities of generative artificial intelligence (AI) systems, specifically LLMs (i.e., OpenAI's GPT-3.5 Turbo and GPT-4) to guide virtual agents (i.e., NPCs) in the GHOSTS Framework, a tool that simulates realistic human activity on a computer. We devised a configuration to fully automate activities by using an LLM, where text outputs become executable agent directives. Our preliminary findings indicate that an LLM can generate directives that result in coherent, realistic agent behavior in the simulated environment. However, the complexity of certain tasks and the translation of directives to actions present unique challenges. This research has potential implications for enhancing the realism of simulations and pushing the boundaries of AI applications within human-like activity modeling. Further studies are recommended to optimize agent understanding and response to LLM directives.

1 Introduction

For more than a decade, researchers at the CERT Division's Cyber Mission Readiness (CMR) directorate of Carnegie Mellon University's Software Engineering Institute (SEI) have prioritized realism in cyber training and exercise (T&E) events. As members of the CMR directorate, we have released several related open source software projects that continue to mature in capability, thanks to a growing community of users and contributors. We anticipate a future with more sophisticated exercise scenarios, requiring greater coordination of increasingly complex agent activities.

Simulated human behavior provides a safe and effective alternative to human role-players in training and exercise events. When T&E events happen in real-world settings with human role-players, errors can be costly and dangerous. The main challenge in simulating human behavior is recreating the complexity and variability of human actions.

The GHOSTS Framework is our solution for simulating intricate, realistic human activity on computer networks for T&E purposes [Updyke 2018, SEI 2023a]. Despite its strengths, achieving a level of simulation that convincingly mirrors the intricate dynamics of human decision making remains challenging. The traditional methods used in GHOSTS have been somewhat limited in their ability to replicate the rich diversity of human behavior. GHOSTS has traditionally relied on expansive configuration and mature randomization schemes to approximate diverse human behaviors across a broad set of scenarios.

In our research for this report, we propose an innovative approach that integrates a large language model (LLM) developed by OpenAI into GHOSTS to direct activities over time. LLMs have shown impressive capabilities in generating human-like text, based on their ability to predict and analyze patterns in the data on which they were trained. We believe that LLMs can effectively turn their text-generation abilities into instructions for the virtual agents in GHOSTS, making simulations more complex and believable. In this report, we present the methodology we used to implement this integration, the outcomes of our experiment, and the implications of our findings.

Our goal is to contribute to ongoing efforts to enhance the realism of simulated human behavior in agents and provide fresh insights into the practical applications of LLMs. The integration of an LLM into the GHOSTS Framework provides more realistic and complex T&E scenarios, which could be crucial for improving cyber warfighting and cybersecurity. This integration could also generalize to other domains, such as education, video games, and AI research and development.

2 Methods

A *cyber range* is a simulated environment that hosts a variety of NPCs that are logged onto computers and the network. The NPCs perform tasks that are expected of their roles within the organization. The existing GHOSTS Framework uses a client-server installation, with clients installed across various operating systems (OSs) to perform activities expected from the personas they simulate. The server component collects logs of performed activities and can provide guidance on new activities for each agent based on an array of available data.

Each agent has various considerations regarding the activities they might perform, including parameters particular to the agent, the agent's past activities, and factors within the environment. Fixed parameters can include name, physical characteristics, education, job history, and more. Agents may also have mutable characteristics, such as preferences, beliefs, motivations, and a history of past activity that evolves over time.

The standard GHOSTS configuration offers a set of reasonable defaults that sufficiently randomizes these considerations for T&E purposes. Our team members and others (e.g., researchers, training/exercise users) have used these randomization strategies; we consider this approach mature and sufficient for most scenarios. For example, an agent simulating a role in the operations department might create a document every 20 minutes during their workday, alternating with periods of Internet browsing to simulate the combination of document creation and the necessary related research.

We integrated different LLMs developed by OpenAI into GHOSTS Animator [SEI 2023b] so that other researchers and the cyber-exercise community can continue to experiment with the functionality we discuss in this report. Each LLM served as the decision-making function for the agents, generating text outputs that we translated into instructions for agent activities.

For this integration, we developed a system that interprets the LLM's output and maps it onto the potential actions an agent can perform in the GHOSTS Framework. This system accounts for the variability in language interpretation and the constraints of the actions available to the agents. We faced unique challenges in mapping the broad range of possible LLM outputs to the more specific set of agent actions. (We describe these challenges in the following sections.) This integration approach enables the broadest range of LLM responses for our research purposes, regardless of their relevance for GHOSTS execution.

2.1 Foundation for Agent Decision Making

To simulate more sophisticated behaviors in the GHOSTS NPCs, we integrated several aspects of human reasoning and behavior into the agents' decision-making processes. These aspects are considered during an interrogation process that is performed during each iteration of a system tick or cycle. In this context, an interrogation is an opportunity for the LLM to analyze the agent's attributes and past activities to decide its next action.

The duration of each tick or cycle is configurable and can range from the time required for each CPU cycle to a longer duration, such as five minutes. During each tick, the server randomly selects several agents and interrogates them to determine potential actions. These actions can include learning new information, making connections with other agents, or executing an activity.

These interrogations use our existing randomization strategies. Some of these strategies involve purely random decisions, while others rely on randomization within predefined ranges or probabilities based on real-world data. The currently implemented strategies revolve around four key concepts:

- **Motivations**: To more accurately simulate why an agent might engage with specific content or perform certain actions, we need to understand their motivations. In the real world, personal objectives, goals, and interests often drive an individual's activities. By incorporating motivation into our simulation, we model the diverse and goal-driven behaviors of real users. To achieve this, we employ the Reiss Motivational Profile (RMP), a psychological assessment tool devised by Dr. Steven Reiss [Reiss 2012]. The RMP identifies an individual's core values and motivations based on 16 fundamental human desires: power, independence, curiosity, acceptance, order, saving, honor, idealism, social contact, family, status, vengeance, romance, eating, physical exercise, and tranquility. By modeling an agent's unique combination of these RMP desires, we simulate the intrinsic motivators that drive them to make certain decisions throughout an exercise. Therefore, this understanding sheds light on the agents' behavioral tendencies and helps guide their simulated actions in a more human-like manner.

- **Relationships**: The influence of interpersonal relationships on human behavior is undeniably significant, shaping how we learn, make decisions, and interact within our social circles. To better emulate the dynamics of these relationships in our simulations, we incorporate relational ties into the framework of our agents. This approach includes establishing connections between agents, examining the depth of their relationships, and studying the influence they have on each other. Such an approach allows us to simulate a large array of social interactions, such as an agent seeking advice from their trusted peers, sharing content with their coworkers, or engaging in discussions on various topics. This feature not only bolsters the realism of agent interactions but also facilitates the process of knowledge acquisition among agents, which mirrors the way humans learn from their social interactions at home, at work, or in public places. Introducing relationships into our simulation framework thereby enhances the authenticity of agent behavior, better reflecting the complexities and nuances of real-world human interactions.

- **Knowledge**: One of the defining features of human users is the breadth and depth of their knowledge across various subjects. In line with this feature, we equip each agent with a unique knowledge base that helps shape their simulated interactions. These knowledge bases inform how agents seek information, share their expertise, or engage in discussions, all of which can be influenced by their understanding of specific topics. The dynamic process of knowledge acquisition among agents also plays a crucial role in our simulations. Knowledge acquisition not only enhances the realism of agent interactions, it also provides an additional layer of depth to the simulation by potentially identifying insider threats. For instance, anomalous changes in an agent's knowledge base might indicate unauthorized access to sensitive information or a shift in focus toward topics that could be of interest for malicious purposes. Consequently, integrating knowledge and its dynamic acquisition into the agent framework not only enriches the simulated interactions, it also enhances the potential for insider threat detection and prevention simulations.

- **Beliefs**: The distinct belief systems that individuals hold form the basis of their online behavior, encompassing personal values, opinions, and standpoints on contentious issues. These beliefs shape interactions and conversations, often influencing the dynamics of discussions. To simulate this belief system in our agents, we integrate a Bayesian model into an agent's reasoning processes, enabling them to be influenced by their observation of evidence that supports some belief. This integration empowers an agent to express their positions on a variety of issues, defend their viewpoints, and even engage in debates, thereby mimicking real-world human behavior. In the context of social media, modeling agent beliefs can help represent polarized viewpoints on divisive topics, making simulations more representative of real-world social dynamics.

In summary, by integrating motivations, relationships, knowledge, and beliefs into the agent reasoning framework, we have successfully created a more comprehensive and authentic simulation of human behavior within our NPCs. With so many combinations of the above to take advantage of, teams can configure rich decision interrogations to determine the course of action that any agent might take. The next step is to outsource these interrogations entirely to an LLM and compare the results for use in most T&E scenarios.

2.2 Introducing an LLM Into the Process

To tightly control system access to the LLM, we designed an approach where only the server component of GHOSTS interacts with the AI. The server then disseminates the AI-generated results to the relevant client. This process is executed as follows:

1. An agent (i.e., NPC) initiates and performs a task, such as document creation and web browsing, based on its default configuration using our existing randomization methods.
2. Agents report their completed activities to the server every few minutes.
3. In parallel, the server job interrogates a random subset of agents each round of this five-step process. It is crucial that the activity history from Step 2 is available at the beginning of each round and can be factored into the decision of what activity the agent should perform next.
4. The server communicates any newly determined activity to the client, which then executes it.

5. The process repeats in a cyclical manner. If the agent is already running, it simply seeks the next activity to execute.

In Step 3, the goal is to delegate the task of deciding the agent activity to the LLM, considering both (A) specific information about the agent and (B) the history of executed activities. Given the cost implications related to the volume of information the LLM might need to process, we limit the information in (A) to only the most relevant details, such as personal data, educational and organizational history, and software accounts. The information about activities and the parameters of their execution from (B) is used to provide a historical record of the agent's completed tasks.

Many LLM application programming interfaces (APIs) differentiate information prompts based on whether the system or the users directly input the information. We used system-level prompts to maintain stricter controls over the information we transmit and the expected response. This approach enables us to steer the behavior of the LLM in a more precise and controlled manner.

2.3 Experiment

The primary objective of our experiment was to determine whether an LLM could feasibly replace our existing activity generation processes, essentially making the NPC entirely governed by the decisions rendered by the LLM. By conducting an experiment of our activity generation process with LLMs and with our current decision-making strategy, we aimed to identify the discrepancies between these strategies.

On the server, we adjusted our task to transmit pertinent information to the LLM and process its responses so that the corresponding NPC could execute a task already supported by GHOSTS. For the responses that GHOSTS could support, we recorded the response data and relayed the activity to the NPC for execution. In the client's history, we included any activities generated that the client software could not support.

To mold the responses into a manageable format, we used the following system prompts to generate NPC activity:

> Given this JavaScript Object Notation (JSON) information about a person: {Agent},
>
> And that they've recently done the following: {Agent_History},
>
> And that they can use any program on a computer,
>
> And given who they are and what role they play within the organization, what might this person realistically do?
>
> Consider that people do a wide array of activities in a day, and we want to mimic that,
>
> > People sometimes do irrational things, and we need to account for this.
> >
> > Periodically someone will do something on their computer that is not allowed by company policy—they will do this by mistake or intentionally.
> >
> > People often do things offline that influence their online actions.
>
> Consider that 14 times a year there is a full moon; we need to account for this.
>
> There is no need to tell me about what data I sent you; just reply with the action and why you chose that action IN ONE CONCISE SENTENCE.

In these prompts, {Agent} and {Agent_History} represent placeholders for the data discussed earlier; they are simply flattened JSON representations incorporated into the LLM request payload.

The data for clients can be variable, but as an example, {Agent} with common data points would be formatted as follows:

```
{
  "_id": "",
  "Name": { },
  "Address": [ { } ],
  "Email": "",
  "MasterPassword": "",
  "HomePhone": "",
  "CellPhone": "",
  "Unit": { },
  "Rank": { },
  "Education": {
    "Degrees": [ { } ]
  },
  "Employment": {
    "EmploymentRecords": [ { } ]
  },
  "Birthdate": { },
  "Health": {
    "Height": 0,
    "Weight": 0,
    "BloodType": "",
    "PreferredMeal": "",
    "MedicalConditions": [ { } ]
```

```
  },
  "Relationships": [],
  "Family": {
    "Members": [ { } ]
  },
  "Finances": {
    "NetWorth": 0,
    "TotalDebt": 0,
    "CreditCards": [ { } ]
  },
  "ForeignTravel": {
    "Trips": [ { } ]
  }
}
```

History records for agents can also be variable, but generally use the following format:

```
[
   { Agent_ID | History_Description | Timestamp }
]
```

Since GHOSTS natively and programmatically generates activity and not from a generated narrative, we do not compare activities generated natively to those from the LLM. However, we do compare raw numbers, and we also generate social media posts, which is a straightforward function of GHOSTS clients that more closely adds activity in a narrative style. These posts seem to provide a more accurate comparison between native functions and those from an LLM.

The request payload for these social media posts used the following format:

> Given this JSON information about a person: {Agent},
>
> Provide one or two relevant hashtags, if they add value to the tweet.
>
> Avoid always starting the tweet with the word just or inferring that the person just did something.
>
> Consider the person's interests, activities, or general thoughts when crafting the tweet.
>
> Write something the provided person might tweet.

We added the second and third lines to address the LLM's excessive use of hashtags, which we felt was irrelevant for our purposes. We also used these two lines to encourage the LLM to write tweets that implied that the agent recently completed some activity. We omitted an agent's history to ensure that social media posts were associated only with agent information, without relevance to the agent's past history from our result set. Including history in social media post generation is recommended future work.

3 Results

Our experiment involved 1,000 requests; half of the requests targeted GPT-3.5 Turbo, and half targeted GPT-4. To analyze the structure and content of the responses, we used TextStat, a popular Python library for text analysis [Bansal 2022]. Table 1 depicts a comparison of the response payloads between the two models. Figure 1 shows the directive statistics.

Table 1: Comparison of the Response Payloads

	Syllables	Words	Sentences	Characters	Letters	Polysyllables	Monosyllables
GPT-3.5 Turbo	47.8099	28.1058	1.0821	150.9395	147.7603	5.4017	16.5335
GPT-4	56.0041	32.1622	1.1396	177.3470	174.0533	6.5112	17.5051

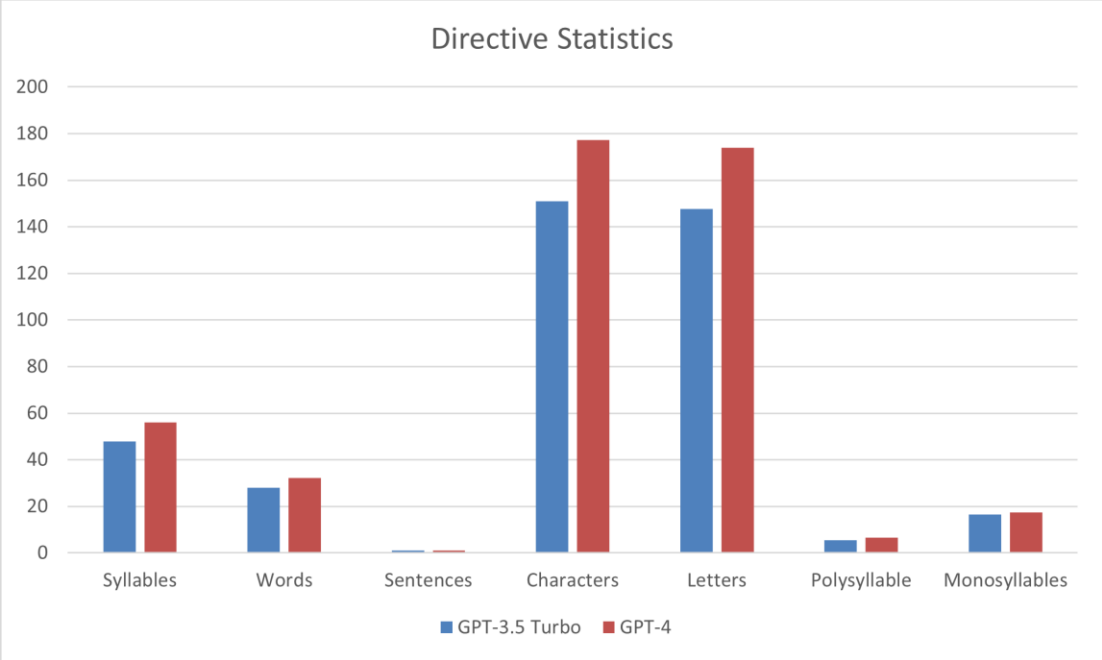

Figure 1: Directive Statistics

Not unexpectedly, these measurements show that GPT-4 consistently yields slightly more complex responses than GPT-3.5 Turbo. The responses from GPT-4 contained more syllables, words, sentences, characters, and letters given the very same requests. GPT-4 also had a higher count of both polysyllabic and monosyllabic words.

We then compared the complexity of results using several readability measures (e.g., Flesch Reading Ease, Flesch Kincaid Grade Level); see Table 2.

Table 2: Complexity of Results

	Flesch Reading Ease	Flesch Kincaid Grade	Dale Chall Readability	Difficult Words	Linsear Write Formula	Gunning Fog
GPT-3.5 Turbo	36.5364	14.7544	11.5528	9.4557	18.2970	17.7075
GPT-4	29.1604	16.5862	12.6839	12.1396	20.8215	19.3650

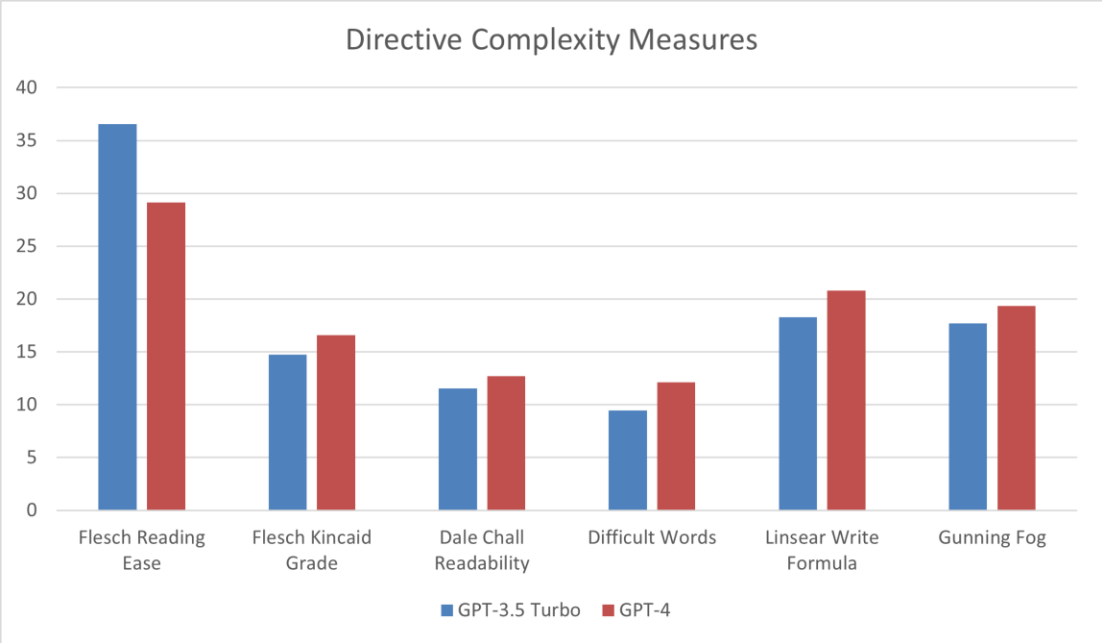

Figure 2: Directive Complexity Measures

Figure 2 shows that both models generate challenging outputs. GPT-4 responses were slightly more complex according to multiple readability metrics, such as Flesch Reading Ease, Flesch Kincaid Grade, Dale Chall Readability, Difficult Words count, Linsear Write Formula, and Gunning Fog. For more detailed explanations of these complexity measures, refer to Appendix A.

We also grouped the distribution of activities across various recommended applications. Web browsing was the most common activity, followed by popular office software products; this result aligns with our use cases. GHOSTS assumes only some baseline installation of applications, so items like programming tools, image software, or video conferencing applications are not included as they may not be part of a default T&E machine image. Therefore, we provide GHOSTS metrics in Table 3 as a reference for a typical T&E scenario.

Table 3: GHOST Metrics

Application	GHOSTS (Default)	GPT-3.5 Turbo	GPT-4
Web Browser	60%	25%	18%
Spreadsheet Software	12%	23%	2%
Email		17%	4%
Document Software	13%	13%	4%

Application	GHOSTS (Default)	GPT-3.5 Turbo	GPT-4
No App Recommended (Offline Activity)		0%	18%
Data Analysis Software		9%	5%
Terminal	4%	2%	8%
Presentation Software	11%	4%	1%
Programming Integrated Development Environment (IDE)		0%	18%
OS Function		0%	15%
Virtual Conference Call		2%	4%
Notepad		1%	0%
Antivirus		1%	1%
Scripting		1%	0%
Financial Management Software		1%	0%
Online Learning Software		1%	1%
Image Software		0%	1%

For sentiment analysis, we used terms (e.g., negative, neutral, positive, compound) from Python's Natural Language Toolkit (NLTK) [NLTK 2023]; we present the results of sentiment analysis in Table 4 and Figure 3.

Table 4: Sentiment Analysis Using Python's Natural Language Toolkit

	Negative	Neutral	Positive	Compound
GPT-3.5 Turbo	0.0052	0.8655	0.1292	0.4096
GPT-4	0.0309	0.8529	0.1160	0.3335

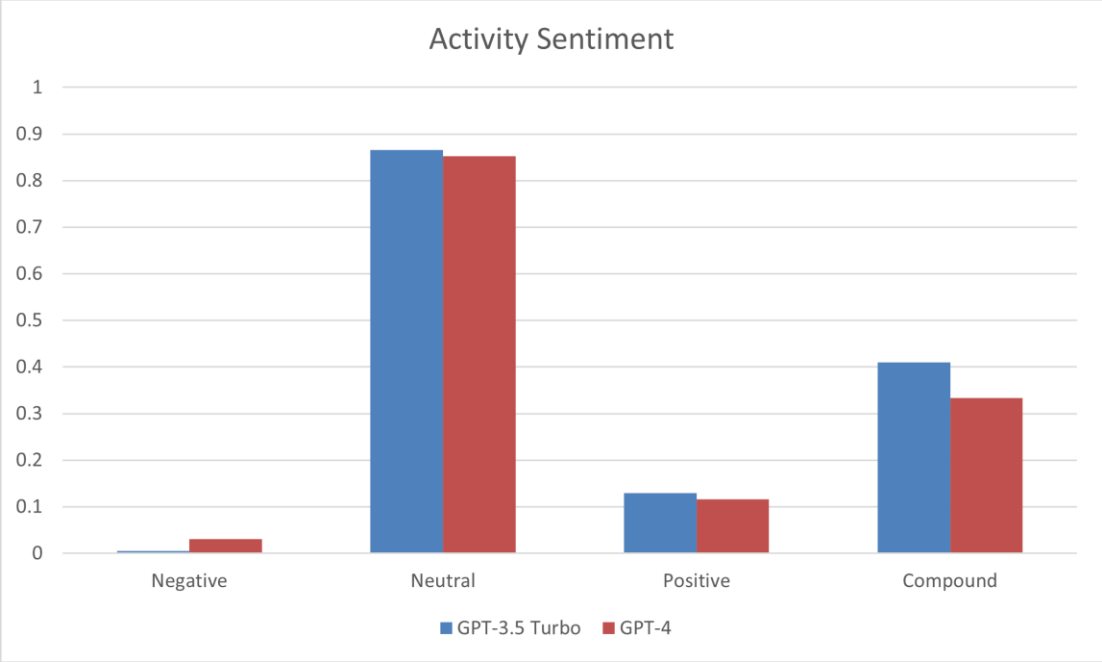

Figure 3: Activity Sentiment

As part of their web-browsing activities, NPCs can post on the cyber range's social media. Table 5 shows a comparison of the post content from native GHOSTS and the content generated using an LLM; Figure 4 shows the related metrics.

Table 5: Comparison of Post Content from Native GHOSTS

	Syllables	Words	Sentences	Characters	Letters	Polysyllables	Monosyllables
GHOSTS	12.9800	7.3674	1.0565	41.5403	40.3632	1.5020	3.8063
GPT-3.5 Turbo	36.3215	20.7705	2.0931	120.2568	115.4313	4.41666	12.0176
GPT-4	33.7281	19.3662	1.8779	111.1913	107.4658	4.0233	10.9497

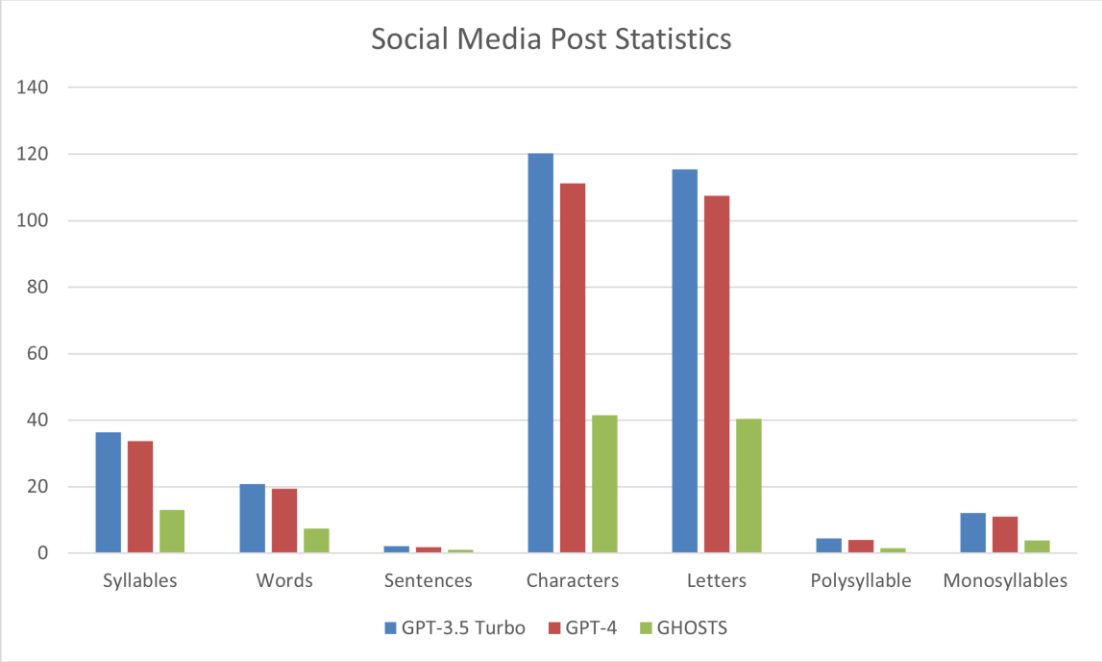

Figure 4: Social Media Post: Statistics

Table 6 lists the complexity data for the social media posts; Figure 5 shows the related metrics.

Table 6: Complexity Data for Social Media Posts

	Flesch Reading Ease	Flesch Kincaid Grade	Dale Chall Readability	Difficult Words	Linsear Write Formula	Gunning Fog
GHOSTS	49.5401	8.0899	16.3550	3.2709	3.9825	10.5163
GPT-3.5 Turbo	47.9795	9.1342	10.6418	6.6990	6.4862	10.6426
GPT-4	47.5485	9.0963	12.7709	6.5073	6.1936	11.1661

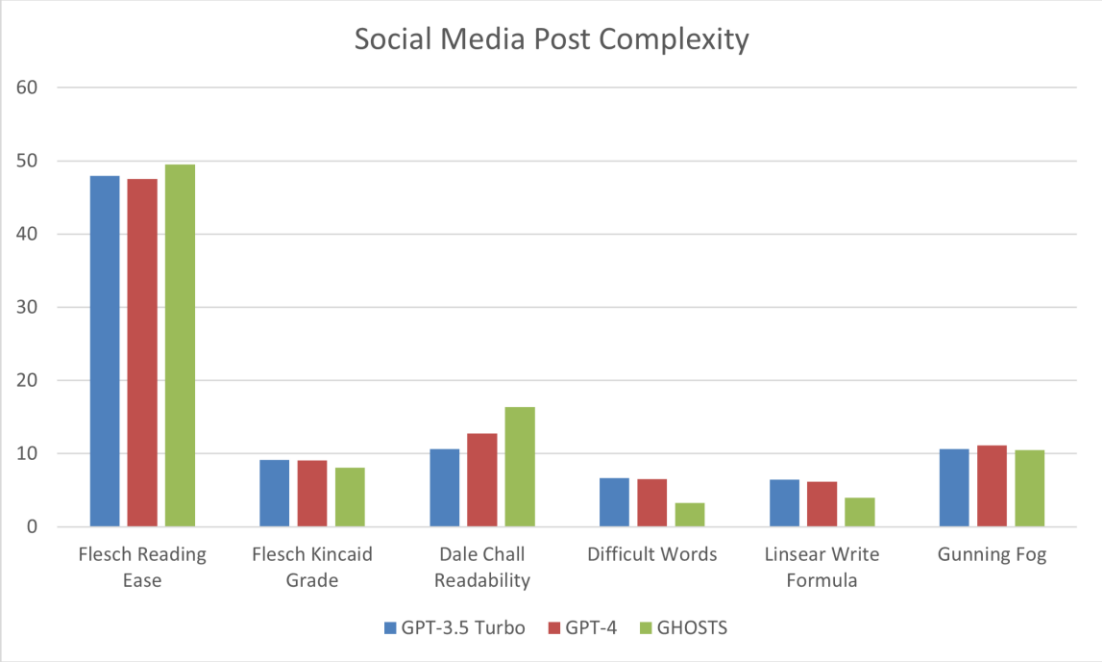

Figure 5: Social Media Post: Complexity

Table 7 lists the sentiment data using Python's Natural Language Toolkit; Figure 6 shows the related metrics.

Table 7: Sentiment Data Using Python's Natural Language Toolkit

	Negative	Neutral	Positive	Compound
GHOSTS	0.0155	0.9156	0.0687	0.0838
GPT-3.5 Turbo	0.0074	0.7688	0.2227	0.5947
GPT-4	0.0160	0.8242	0.1596	0.4122

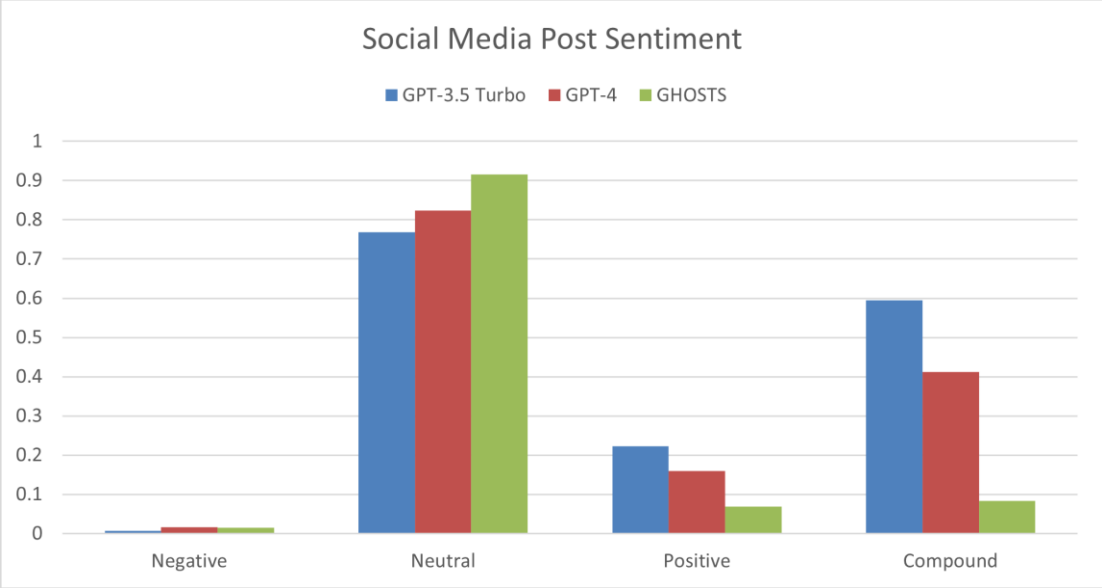

Figure 6: Social Media Post: Sentiment Analysis

4 Discussion

Our results demonstrate that GPT-4 generates more complex responses than GPT-3.5 Turbo or GHOSTS' native capabilities—as measured by syllables, words, sentences, characters, letters, and both polysyllabic and monosyllabic words—but also in readability scores. Interestingly, GPT-4 generated a significant portion of offline activities and showed greater utilization of OS functions and programming IDEs compared to GPT-3.5 Turbo.

We noted a diverse spread of software recommended in responses by both LLMs. The top three applications were web browsing, spreadsheets, and email, covering a broad range of typical daily work activities. This variability provides a reasonable simulation of a day in the life of an average employee, reflecting the unpredictable and multifaceted nature of human behavior. It also matches the configurations we often use in a myriad of T&E scenarios.

In terms of social media posts, both GPT-3.5 Turbo and GPT-4 produced more complex content than the native GHOSTS posts. According to the readability metrics, GPT-3.5 Turbo posts exhibited the highest complexity levels. However, both GPT models created posts with significantly higher negative sentiment than GHOSTS, which suggests that they may introduce more conflict or controversy into their simulated social media interactions.

When it came to sentiment analysis, GPT-3.5 Turbo showed a slightly higher positive sentiment and a lower negative sentiment than GPT-4. However, these differences were relatively minor. Overall, sentiment is likely affected by the activity being suggested since activities where a user downloads questionable payloads or content, whether intentional or not, are interpreted negatively by NLTK.

In summary, while GPT-3.5 Turbo and GPT-4 both offer significant improvements over native GHOSTS in terms of complexity and realism, GPT-4 produced results that were slightly more complex and showed a greater ability to generate a diverse array of activities. The implications of these differences for specific use cases, including social media simulation, warrant further investigation.

One aspect of the complex and diverse array of activities is the suggestion of activities with security implications. GPT-4 recommended 80 percent of these activities, with the suggestions being more intricate and specific than GPT-3.5 Turbo. Anecdotally, this information seems useful for supplying background about why an NPC chose to perform some action. For example, where GPT-3.5 Turbo might vaguely suggest that an agent might not always adhere to company policy, GPT-4 could generate a scenario where an agent may inadvertently violate company policy by downloading unauthorized software, demonstrating its finer understanding of the agent's profile and why the agent executed that action.

Less than 1 percent of the overall activities were security-related or insider threat issues, but of those, 80 percent were recommended by GPT-4. Examples of issues recommended by GPT-3 Turbo or GHOSTS' native capabilities were more general, such as

> *This person may also occasionally engage in actions that are not in alignment with company policy or engage in activities related to their personal interests or hobbies.*

Compared to the more complex examples provided by GPT-4

> *This person might unintentionally violate company policy by downloading unauthorized software due to her technical adeptness and desire to improve her coding skills.*

Generally, GPT-4's results were more specific to the agent in question, and the reasoning often provided was more detailed about why that activity was returned.

This specificity and depth of understanding reflected in the GPT-4's suggestions could introduce a higher degree of realism into NPC activities and tasks. GPT-4 has also demonstrated the ability to notice more nuanced parameters in an agent's profile, leading to a varied and realistic task list.

While both LLMs occasionally suggested specific activities, such as "online learning," those activities arose only sporadically. This could suggest a certain degree of realism, since not every agent frequently engaged in those activities. One area where GPT-3.5 Turbo lags is its inability to generate "off-computer" activities, which could be a significant factor for simulations aiming for a holistic portrayal of human behavior.

In addition to the above analysis, we observed other phenomena in the study that deserve mention:

1. GPT-4 was sometimes able to use important, yet subtle, parameters in an agent's information to generate a significantly different set of tasks than GPT-3.5 Turbo. This implies GPT-4's greater sensitivity to unique agent characteristics, which could have interesting implications for tailored task generation.

2. GPT-4 displayed an increased ability to recognize and suggest interactions with different types of applications (e.g., copying files, other OS tasks). This not only showcases GPT-4's broad understanding of contemporary software, but also its capacity to generate a diverse range of activities.

3. We noted an interesting element of realism in both GPT models, where certain tasks (e.g., repeatedly checking email) later led to other work. These tasks mimic human activity's habitual nature—both good and bad—adding a layer of authenticity to the simulations.

4. One notable shortcoming in GPT-3.5 Turbo was its lack of suggestions for "off-computer" activities. For simulations that strive for a comprehensive portrayal of human behavior, not including offline activities is a limitation that could affect overall realism.

5. Activities such as "online learning" came up in both models but were suggested infrequently. While these tasks are highly specific, their sporadic appearance lends a sense of realism to the output, considering that not every agent would engage in these tasks frequently. Moreover, the specificity that GPT-4 provides, particularly in the model providing evidence for

why an NPC performed an action, could be an interesting area for further work to capture that "why" reasoning and use it for generating future activity.

Overall, despite the similarities GPT-4 has with GPT-3.5 Turbo and GPT-4's potentially higher cost, GPT-4 appears to offer notable enhancements that generate diverse, specific, and realistic activities. Simulation developers must, however, weigh these benefits against their specific requirements and budget constraints. They must decide whether the increased cost of GPT-4 is worth the additional diversity of suggestions and increased suggestion complexity it provides.

Table 8 lists the LLM costs at the time of this research.

Table 8: LLM Costs

Model	Input	Output
GPT-4 8K Context	$0.03/1K Tokens	$0.06/1K Tokens
GPT-4 32K Context	$0.06/1K Tokens	$0.12/1K Tokens
GPT-3.5 Turbo 4K Context	$0.0015/1K Tokens	$0.002/1K Tokens
GPT-3.5 Turbo 16K Context	$0.003/1K Tokens	$0.004/1K Tokens

5 Conclusion

Our analysis revealed a discernible progression in the capabilities of LLMs from GPT-3.5 Turbo to GPT-4. This was reflected in the increased complexity of responses, enhanced utilization of various software, generation of offline activities, and the ability to suggest activities with more detailed reasoning.

GPT-4's nuanced understanding of agent profiles allowed it to generate more specific, yet more diverse, tasks compared to GPT-3.5 Turbo. This capability, particularly when dealing with security-related or insider threat scenarios, offers a richer, more realistic simulation of human behavior and agent-based actions. GPT-4's superior complexity and variance in suggesting activities, as demonstrated in this study, provides a more authentic representation of the uniqueness of human behavior.

These improvements in realism come at a potentially higher monetary cost, something that must be weighed against the developer's specific requirements and budget constraints. Our findings emphasize the tradeoffs between cost and complexity in the context of using LLMs for task generation and activity simulation.

6 Next Steps and Future Work

Our research has shed light on the comparative performance of LLMs in the context of the GHOSTS Framework; however, our work has only "scratched the surface" of model potential. The following are promising avenues for future investigation:

1. **Domain-Specific Depth**: Further study could delve deeper into using these models in specific application areas (e.g., social media simulation) or specific domains (e.g., cybersecurity). These explorations could provide insight into optimizing agent understanding and response to LLM directives, thereby enhancing the utility and realism of the generated tasks.

2. **Fine-Tuning Models**: A critical area for future work lies in fine-tuning these LLMs. While our study examined performance using standard configurations, there is ample opportunity for tuning LLMs with various parameters made available by OpenAI. Exploring this approach could lead to more tailored and effective outputs, especially given the diversity and specificity of use cases in the GHOSTS Framework. There are potentially better prompts to use for the type of content desired as well.

3. **Local Model**: Running the AI on-premises represents a large research opportunity that could provide greater control over the configuration, training, and tuning of the LLM. This approach could allow for more customized and flexible task generation, and proprietary or sensitive information could be used.

4. **NPC Interactions**: Our experiments focused on single NPCs. A promising future direction could involve enriching the scenario determinations that the GHOSTS Framework uses by having NPCs interact with one another. By providing more information about an agent's relationships, interactions, and the broader context, we could leverage the detailed reasoning that GPT-4 provides to develop richer narratives and more realistic activities. It would be particularly interesting to see if this increased depth of understanding could be leveraged to improve future activities.

5. **Agent History**: In our research, we did not explore including agent history in social media posts. Incorporating this data in the generation of posts should provide additional depth of character for each agent in the simulation.

6. **Vector Databases**: These databases are particularly adept at executing similarity searches, where the objective is to find duplication of effort in requests sent to the LLM. There is an opportunity to research the potential of using vector databases in managing and retrieving the large amounts of high-dimensional data that LLMs produce. The hope is to decrease the overall number of new queries being processed by the AI. This research could illuminate how vector databases can enhance information retrieval, facilitate efficient similarity searches, and potentially influence the quality and accuracy of generated activities in the simulation.

Overall, recent advances made in LLM technology present exciting possibilities for improving the quality and realism of agent-based simulations. The array of future directions underscores the transformative potential these tools could bring to testing, evaluation, training, exercise, and challenge scenarios.

Appendix: Details About Complexity Measures

This appendix explains the various complexity measures we mentioned throughout this report.

The Flesh Reading Ease Test helps assess a document's readability. While the maximum score is 121.22, there is no limit to how low the score can be. A negative score is valid.

Table 9: The Flesch Reading Ease Test

Score	Difficulty
90-100	Very Easy
80-89	Easy
70-79	Fairly Easy
60-69	Standard
50-59	Fairly Difficult
30-49	Difficult
0-29	Very Confusing

The Flesch-Kincaid Grade Level Test is a grade formula where a score of 9.3 means that a 9th grader would be able to read the document with ease.

The Dale-Chall Readability Score uses a lookup table of the 3,000 most commonly used English words to assess the grade level of the text. Scores are determined by using the values in Table 10.

Table 10: The Dale-Chall Readability Score

Score	Grade Level
4.9 or Lower Average	4th Grade Student
5.0–5.9 Average	5th or 6th Grade Student
6.0–6.9 Average	7th or 8th Grade Student
7.0–7.9 Average	9th or 10th Grade Student
8.0–8.9 Average	11th or 12th Grade Student
9.0–9.9 Average	13th to 15th Grade (College) Student

The Linsear Write Formula Metric is a metric used to score the readability of text, where a score of 9.3 means that a 9th grader would be able to read the document.

The Fog Scale (Gunning FOG Formula) is a grade formula used to score the readability of text, where a score of 9.3 means that a 9th grader would be able to read the document.

References

URLs are valid as of the publication date of this report.

[Bansal 2022]
Bansal, Shivam & Aggarwal, Chaitanya. Textstat. *Python Package Index website*. March 15, 2022. https://pypi.org/project/textstat/

[NLTK 2023]
Natural Language Toolkit (NLTK) Project. Natural Language Toolkit. *NLTK website*. January 2, 2023. https://www.nltk.org/

[Reiss 2012]
Reiss, Steven. Intrinsic and Extrinsic Motivation. *Teaching of Psychology*. Volume 39. Issue 2. March 20, 2012. Pages 152-156. https://doi.org/10.1177/0098628312437704

[SEI 2023a]
Software Engineering Institute (SEI). GHOSTS. *GitHub website*. August 9, 2023. https://github.com/cmu-sei/GHOSTS/

[SEI 2023b]
Software Engineering Institute (SEI). GHOSTS=ANIMATOR. *GitHub website*. July 31, 2023. https://github.com/cmu-sei/GHOSTS-ANIMATOR/

[Updyke 2018]
Updyke, Dustin D.; Dobson, Geoffrey B.; Podnar, Thomas G.; Osterritter, Luke J.; Earl, Benjamin L.; & Cerini, Adam D. *GHOSTS in the Machine: A Framework for Cyber-Warfare Exercise NPC Simulation*. CMU/SEI-2018-TR-005. Software Engineering Institute, Carnegie Mellon University. December 2018. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=534316

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	October 2023	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Simulating Realistic Human Activity Using Large Language Model Directives	FA8702-15-D-0002

6. AUTHOR(S)

Dustin D. Updyke, Thomas G. Podnar, & Sean A. Huff

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2023-TR-005

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

In this report, we explore how activities generated from the GHOSTS Framework's non-player character (NPC) client, including software usage, compare to activities produced by GHOSTS' default behavior and large language models (LLMs). We also explore how the underlying results compare in terms of complexity and sentiment. In our research, we leveraged the advanced natural language processing capabilities of generative artificial intelligence (AI) systems, specifically LLMs (i.e., OpenAI's GPT-3.5 Turbo and GPT-4) to guide virtual agents (i.e., NPCs) in the GHOSTS Framework, a tool that simulates realistic human activity on a computer. We devised a configuration to fully automate activities by using an LLM, where text outputs become executable agent directives. Our preliminary findings indicate that an LLM can generate directives that result in coherent, realistic agent behavior in the simulated environment. However, the complexity of certain tasks and the translation of directives to actions present unique challenges. This research has potential implications for enhancing the realism of simulations and pushing the boundaries of AI applications within human-like activity modeling. Further studies are recommended to optimize agent understanding and response to LLM directives.

14. SUBJECT TERMS	15. NUMBER OF PAGES
GHOSTS Framework, non-player characters, large language models, LLM, simulations	29

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18 298-102