GPT-NPC: Enhancing NPC Human-Likeness and Autonomy in Video Games

David Ogunlesi[a,*] **and Xiaoyang Wang**[a,**]

[a]Department of Computer Science
University of Exeter, UK

Abstract. Non-player Characters (NPCs) in video games have been ubiquitous for many years. However, the interactions players have with these NPCs have typically been limited to pre-defined dialogue options, limiting the player's immersion within the game world. Recent advances in AI and Natural Language Processing (NLP) have opened up new possibilities for creating NPCs that are capable of engaging in more naturalistic conversational interactions with players. We explore the use of GPT-powered NPCs in video games in the context of virtual reality and develop NPCs that possess long-term memory, autonomous behaviour, and believable interactions. We propose a novel **GPT-NPC** framework that utilises GPT models, in a modularised approach, to generate coherent and engaging dialogue for NPCs, while also incorporating thought and emotion modelling to create more immersive characters. To assess the effectiveness of our framework, human participants are involved, to rate the human-likeness of NPCs. We demonstrate a mixed level of general believability, showing a rating score of 60%. From this, we further discuss ways to better improve and utilise the framework. This paper aims to contribute to the growing field of AI-powered game development and create more engaging gaming experiences for players.

Accepted for presentation at the First Workshop on "eXtended Reality & Intelligent Agents", XRIA24@ECAI24, Santiago De Compostela (Spain), 20 October 2024.

1 Introduction

"How may we make machines think?" is a question that humans have been pondering since the very genesis of machines; and only an evolution of the much older question *"How may we make agents in the likeness of mankind?"*. The field of Artificial Intelligence (AI) intends to solve this very question. 70 years after the first formalisation of AI [14], the debut of attention-based generative transformers [29] paved a new way forward, advancing the development of Artificial General Intelligence (AGI), which would directly answer this question. Researchers and enthusiasts have envisioned computational agents that can serve as believable proxies for human behaviour for many decades evidently, and this has extended to many industries. With the rise of the generative era of AI, and the release of *ChatGPT* [24], interest has skyrocketed. This interest has further exemplified the use cases of such agents, ones which could populate virtual spaces and communities, exhibit realistic social phenomena [21], provide guidance and counselling, help people with social anxiety [30, 9], deal with loneliness, etc. This paper focuses on infusing digital non-player characters (NPCs) that can navigate the complexities of human interaction and digital worlds in a believable way, contributing significantly to immersion in interactive entertainment. Generative agents, and by extensions Large Language Models (LLMs) have sparse research into embodying these models into physical/digital spaces. Video game worlds act as a good medium to explore the extent to which these models can observe, understand, interact with and affect their environment.

In this paper, we introduce **GPT-NPC**s, agents that draw on generative models, specifically *OpenAI's* GPT series of models, to simulate believable human behaviour. Through experiments with human participants, we aim to demonstrate the factors that contribute to the perception of human likeness. GPT-NPCs draw on a wide range of inferences about themselves, the world, and their environment. They could generate thoughts, act on those thoughts, display emotions, maintain active memories and engage in conversation with the user in video game environments.

The contributions of this paper are three-fold. First, identifying the most significant contributing factors towards human likeness; second, presenting an architecture that achieves a reasonable level of human likeness; and third exploring the impact of thinking and emotive agents.

2 Related Work

In this section, we reflect on prior literature on computer-human interaction, specifically to build accurate proxies of human likeness. This objective has remained a challenge due to the complexity of human nature [15]. However Large Language Models, though insufficient alone, open a door for creating more believable agents when a suitable prompting architecture is used.

2.1 Pre-Transformer Approaches to Human-like AI

In the pursuit of creating human-like AI, various pre-transformer approaches have been explored. These include rule-based systems such as Finite State Machines (FSM) [28], Behavior Trees, Goal-Oriented Action Planning (GOAP)[5], and models based on fulfilling simulated needs. These systems have been utilised in video games to create AI that can react to the player realistically. While these systems offer structured ways to emulate human behaviour, they often fail to capture the complexity and nuances of human interactions. These methods have limited ability in addressing possible behaviour a human would make in the same situation. There has been a shift towards

---

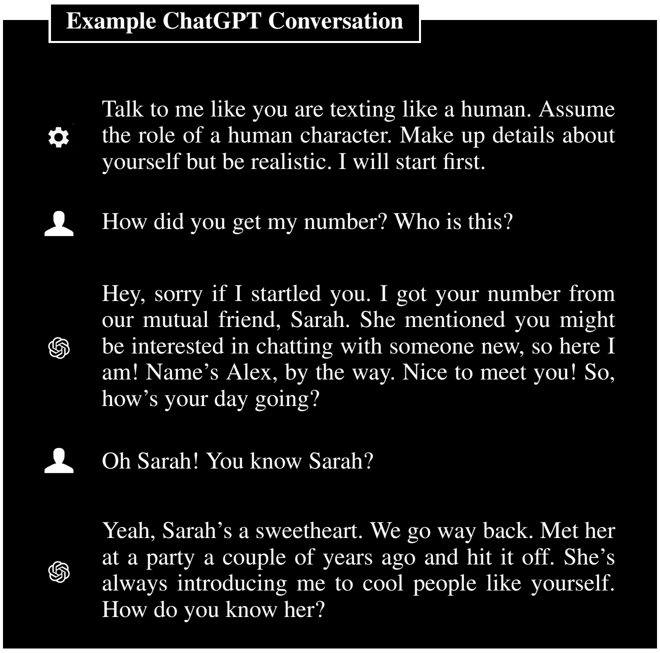
[*] Email: do312@exeter.ac.uk

[**] Corresponding Author. Email: x.wang7@exeter.ac.uk.

growing demand for larger, more interconnected virtual worlds to accommodate increasingly complex behaviours [8]. However, much of the content in these worlds is manually crafted, posing scalability issues for truly open-ended experiences. Machine learning techniques have shown promise in addressing this challenge through pattern recognition, but does not extend to the generation of dialogue. Nevertheless, the development of believable AI agents remains an open problem, with LLMs emerging as a leading approach.

2.2 *Large Language Models for Human-like Behaviours*

LLMs present a compelling solution for simulating human-like behaviour due to their ability to encode a wide array of human actions and responses from vast training datasets. These models can be prompted to generate behaviour that appears more believable comparing with non-LLM models.



Example ChatGPT Conversation

Talk to me like you are texting like a human. Assume the role of a human character. Make up details about yourself but be realistic. I will start first.

How did you get my number? Who is this?

Hey, sorry if I startled you. I got your number from our mutual friend, Sarah. She mentioned you might be interested in chatting with someone new, so here I am! Name's Alex, by the way. Nice to meet you! So, how's your day going?

Oh Sarah! You know Sarah?

Yeah, Sarah's a sweetheart. We go way back. Met her at a party a couple of years ago and hit it off. She's always introducing me to cool people like yourself. How do you know her?

Figure 1. Example of a prompted conversation with ChatGPT demonstrating human-likeness. Notice ChatGPT's propensity to hallucinate.

By chaining prompts or simulating a chain of thoughts [33, 18], researchers have demonstrated the potential of LLMs in diffirent tasks, such as solving abstract inductive reasoning problems [32], simulating villages populated by multiple LLMs [22], and creating AI chatbots resembling specific characters [17]. This posits that LLMs could be one of the key ingredients in the quest to create believable agents, as they possess the breadth of encoded knowledge of human, the world, culture and human experience to draw upon. However, a major hindrance in leveraging LLMs for this purpose is the limitation imposed by the context window, which restricts the model's ability to remember prompts. This issue has been solved by the use of Vector Databases [10]. Vector Databases allow GPT models to draw upon a repository of knowledge based on relevance, which offloads the overhead of context memory thus keeping the prompt length to a minimum. However, the utility of vector databases is shadowed by recent advancements in context length. Google's *Gemini* boasts an incredible 1.5 million token context window [25], reaching a theoretical limit of 10 million tokens. To illustrate the power, Gemini would be able to learn an entire language it has never seen before, purely through its context window. It might be able to store dozens of research papers and give reasoned and justified ideas for novel research ideas. However, with increased context comes an increased operation cost. It is unclear how significant the context window is to the intelligence of GPT models, their ability to perform tasks, and how well they work with other models. There is likely to be remarkable applicability to create proxies of human interaction. It is important to first see what can be done with less, as it can only improve the operation at scale.

2.3 *GPT Models for Dialogue Generation*

GPT series models are naturally good at generating dialogue with similarity to human behaviour to some extent. GPT-4 has passed a Turning test [13]. However, it would be misleading to say GPT-4 reaches a human level of believability alone. GPT-4 was only able to pass the turning test 43% of the time according to [12]. This is significantly under the pass rate baseline for humans of 63%. However, being able to pass the turning test at all is remarkable. Humans are social creatures. Our intelligence and the emergence of consciousness itself can be attributed to an evolutionary need to socialise [16]. To this end, the power of GPT models to generate dialogue is one facet that assists in simulating conscious behaviour.

GPT models work mostly with textual data. Using *OpenAI's* "Chat" models allows us to prompt the GPT model conversationally. These prompts are given within the context window of the API, analogue to *working memory* for GPT models, the major limitation on memory for models. Current small context windows, apart from those in enterprise GPT models, cannot accommodate the extensive specific knowledge and detailed prompts required to consistently align GPT models with a character. *Vector Databases* are an instrumental tool in creating more consistent conversational GPT models.

3 GPT-NPC: Core Architecture

3.1 *Modular Asynchronous Approach*

We present the architecture with modular units, see Fig. 2. Each module runs its processing asynchronously. This prevents modules from having to wait for each other, enabling the "mind" of the agent to think all at once, potentially akin to how human minds process information. A modular approach also allows the architecture to be extended trivially.

Module Interface The abstract module has an interface which provides functions that can be derived from concrete module implementations. These interfaces form the lifecycle of the module.

These interface functions are split into 2 groups, *Initialisation* and *Update*, as shown in Fig. 3. Different granularities of updates were added to accommodate varying degrees of information priority. Not all information requires frame-by-frame freshness, for example, idle thoughts can be generated every 4 seconds on long updates, but perceiving world events would need to occur every frame. *AnnouceOnUpdated* allows modules to announce when they have updated their information, allowing other modules to respond and update their information correspondingly, leading to a wave propagation of updates whilst the agent operates.

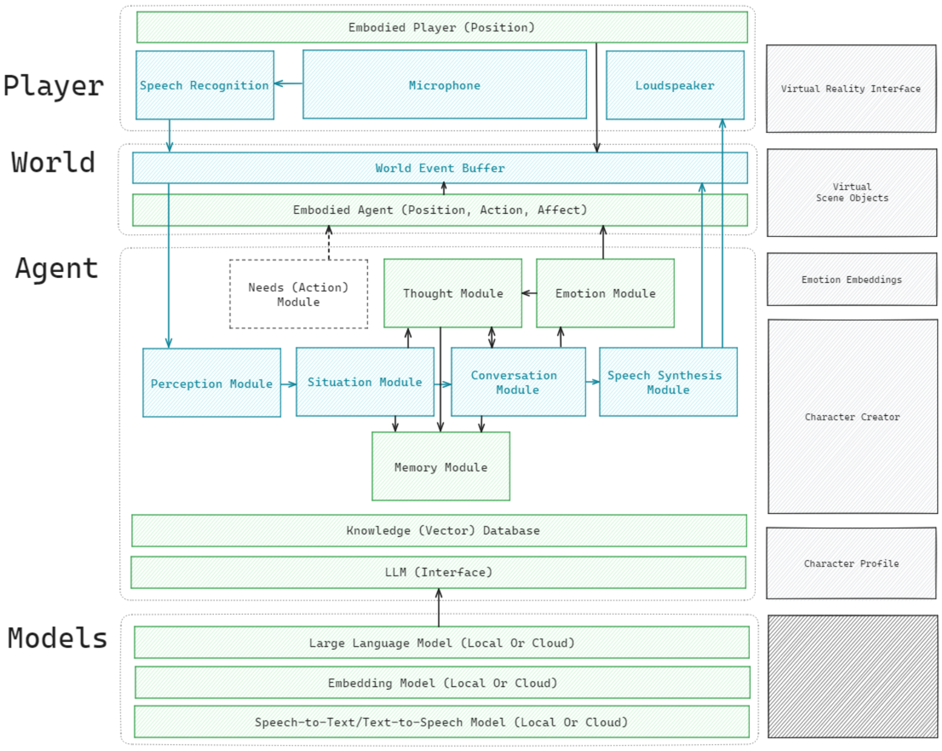


Figure 2. The proposed GPT-NPC architecture. Arrows denote a *work flow* from A → B. Agents perceive their environment, and all relevant perceptions are transformed into high-level comprehension, i.e., a *Situation*. Situations create a wave of updates, generating internal reflections (thoughts) and generating conversational replies. This makes up the thinking process, with all internally processed thoughts being saved into a memory buffer. Relevant memories are retrieved to determine actions. On the right side, the grey boxes show the abstracted components of the GPT-NPC, from the implementation perspective.

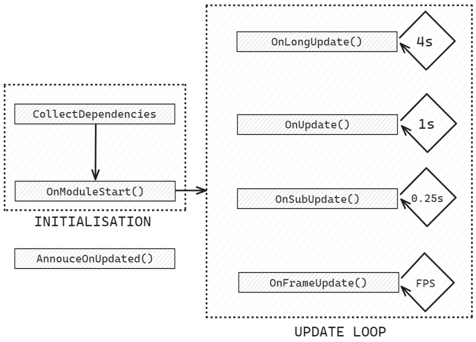


Figure 3. Architecture for a single module. Interface functions are split into 2 groups, *Initialisation* and *Update*. Different granularities of updates were added to accommodate varying degrees of information priority. For example, long updates can be generated every 4 seconds; some updates would need to occur every frame. *AnnouceOnUpdated* allows modules to announce when they have updated their information, allowing other modules to respond and update their information correspondingly.

3.2 Main Information Pathway

Each module can access information from the other module, thus forming dependency chains. However, these are soft dependencies and no module requires another module to function. Some modules could be described as "Core" modules, forming the main information processing pathway that facilitates conversation. This pathway is highlighted in blue in Fig. 2. Similar to the neural pathways in the brain, many different information pathways can be derived, each pathway a string of individual processing units, and each processing unit potentially harbouring its own Generative Transformer. We could design pathways for ideas & creativity, a pathway for automatic behaviours & a pathway for habits, and so on. However, we artificially attempt to mimic human behaviours from a top-down approach, whereas the human mind is naturally bottom-up. To our knowledge, no bottom-up approaches for creating such in all areas of human likeness exist. The full potential of our "forced" approaches must be carefully explored. This conclusion may very well bring us to a suitable simulacra of human likeness such that its utility is tenable.

Overall the modular approach builds a team of GPTs, all working together with individual tasks to create a whole which is greater than the sum of its parts. This model, *Mixture of Experts* [31], is a core tenet of the GPT-NPC architecture we propose. However, we use LLMs instead of smaller precisely trained models. We will discuss the main pathway in more details, as that constitutes the core modules of the architecture.

3.3 Player Interaction

The *Player Interaction* icomponent is the least technically constrained part of the architecture and can be replaced with any interface capable of producing the required world representation, as discussed in the following subsection. The ***Speech Recognition*** module is the most important module for converting player speech into a textual representation. In our implementation, OpenAI's *Whisper* [23] Speech-To-Text model was used for its world-class speech recognition capabilities.

3.4 World

The world embodies the AI in a tangible space and allows it to build an awareness of its surroundings, which grounds it and contributes to behavioural consistency. However, GPT models, and all LLMs predominantly operate with text as the main language of communication [29]. This could be thought to be their internal mode of thought in the context of embodying models within agents. For agents to perceive the world, the world needs to be translated into such representation, forming the *Internal World Representation* of the model. A distinction should be made between our use of the term “internal world model” and its other meaning as knowledge about the human world embedded into the model as a side-effect of its training process. We explicitly define how the world is represented and manipulate it through careful prompt engineering to help the model understand through its context window, which is the state of the world.

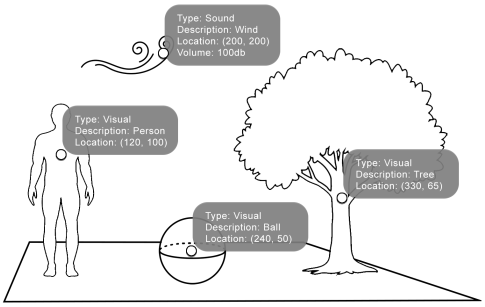


Figure 4. Illustration of *World Events*. Each event has attributes that encode intrinsic properties about the world relevant to the agents.

World Representation (World Space) In the GPT-NPC architecture, the world is represented internally through strings. These strings are known as *World Events*. This layer of representation sits underneath the digital world that that user can see, and thus can be considered an abstraction. It follows that the agent’s world model is also abstracted away. GPT-4 as of recently obtained the capability to parse video and images[2], which could be used to instead provide a local view of the world. In the current architecture, the world model is global. The agents’ perception of this global world model will be discussed below.

World events have attributes that describe the location, and source additionally. The type of world event adds additional attributes. In the case of sound events, their volume and duration are also considered. These are all encoded into the final string representation, for example:

> `“Heard Person saying ‘Hello’ at 5 metres away, volume normal.”`

Fig. 4 illustrates how the environment and underlying world representation relate.

World Event Buffer World Events need to be stored, so they can be referenced by the agents that inhabit the world. We employ a “World Buffer” which coalesces all the world events into a pool of events to draw from. This buffer acts as an internal truth model of the world which is then interpreted by agents.

The reasoning behind pooling world events is motivated by the need to enable the architecture to handle multiple agents. In the case of multiple agents, having a solid world representation assists in agents being able to act realistically, such as only interacting with agents they can see.

3.5 Agent

The most important facet of the GPT-NPC architecture is the agent. The agent can be split into 4 core modules, being *Perception*, *Situation*, *Conversation* and *Speech Synthesis*, and 4 complementary modules, being *Memory*, *Thoughts*, *Emotions* and *Needs*. We start with the main pathway as shown in Fig. 2.

Perception Module Agents parse the world representation, filtering it through their perception module, transforming it into a subjective internal representation, as shown in Fig. 5. Subjectivity is an interesting emergent feature of restricting agents’ global view of the world. Agents do not have perfect knowledge and, thus are susceptible to flawed or misinformed behaviour, hallmarks of human genuineness.

All
World
Events
Perception Module
Percieved
World
Events

Figure 5. The perception module.

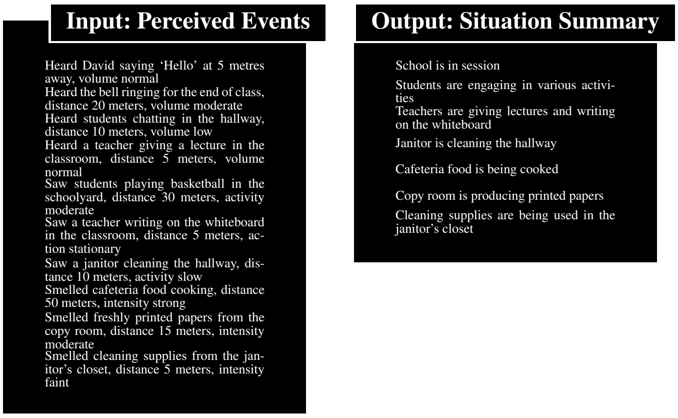
Representations are grouped into modalities from the most important to the least important: Sight, Hearing, Smell, Touch and Taste. Smell, Touch and Taste have been excluded in this paper, but are possible within the architecture’s design.

In this respect, the perception module is synonymous with the sensory system in humans. For each modality, a sensory limit is put in place to prevent overstimulation, limiting the number of attributes in events (see Fig. 4). Sight has a limit of 8, auditory events have a limit of 4. This continues for the other types. This restriction is important due to the limited context window of GPT models. If there were a thousand visible people in front of the agent, parsing them all into the model’s context would be impossible. The sensory restriction also mimics another facet of human behaviour, *selective attention* [7], one that is best demonstrated with the *Monkey Business Illusion*. When prompted to watch an unassuming video of people doing something attention-grabbing, viewers rarely notice a person in a monkey costume entering and exiting the video.

Situation Module (GPT API) The situation module is GPT-powered, see Fig. 6. Its role is to aggregate perceived world events into a summary to provide a comprehensive overview of what is happening. It can be seen as a compression function, that reduces the complexity of the agents the world perceives into an easier-to-understand form. The purpose of this module is to increase comprehension. Fig. 7 shows an example of its input and output.

Perceived
World
Events
Situation Module
Situation
Summary

Figure 6. The situation module.



Input: Perceived Events

Heard David saying ‘Hello’ at 5 metres away, volume normal
Heard the bell ringing for the end of class, distance 20 meters, volume moderate
Heard students chatting in the hallway, distance 10 meters, volume low
Heard a teacher giving a lecture in the classroom, distance 5 meters, volume normal
Saw students playing basketball in the schoolyard, distance 30 meters, activity moderate
Saw a teacher writing on the whiteboard in the classroom, distance 5 meters, action stationary
Saw a janitor cleaning the hallway, distance 10 meters, activity slow
Smelled cafeteria food cooking, distance 50 meters, intensity strong
Smelled freshly printed papers from the copy room, distance 15 meters, intensity moderate
Smelled cleaning supplies from the janitor’s closet, distance 5 meters, intensity faint

Output: Situation Summary

School is in session
Students are engaging in various activities
Teachers are giving lectures and writing on the whiteboard
Janitor is cleaning the hallway
Cafeteria food is being cooked
Copy room is producing printed papers
Cleaning supplies are being used in the janitor’s closet

Figure 7. Example input and outputs for the situation module. The situation module summarises what the agent perceives, transmuting it into a more concise and high-level form.

Conversation Module (GPT API) The conversation module is one of the key modules in the main information pathway. Its purpose is to take the internal state of the agent and decide what to say next (Fig. 8). It is directly responsible for giving agents the ability to communicate, and for this reason, employs the most prompt engineering to achieve its goal.

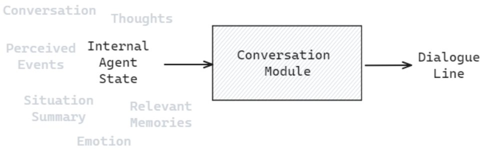


Figure 8. The conversation module takes the internal state, outputting the next line of dialogue if a conversation is active

Fig. 9 illustrates the flow of information, starting with the situation module being updated. *CheckForSpeechDirectedAtMe* is from the *Situation* module, which checks if any relevant auditory world events may be directed at the agent, such as a person saying “Hello” to greet the agent. If the event was directed towards the agent, then *GenerateNextSentence* decides on what to say in response. This forms the core backbone of the module. Alongside the main process, are 2 side processes: *PROACTIVE & ASYNCHRONOUS*.

Proactive Process The proactive process is a vital component of the conversation module which allows the agent to exercise will and be autonomous. This is a core pillar in creating believable proxies of humans. When agents generate thoughts, *CheckForSpeechDesireFromThoughts* determines whether the agent wishes to say something. If there is this desire, and they are free to speak, then they begin to generate a new line of dialogue based on the prior thought. This allows agents to initiate conversation with the user, or other agents in the world, *proactively*. Fig. 10 shows an example conversation utilising this feature. If the agent desires to speak (proactive), or there is an event was directed towards the agent (main process), the *GenerateNextSentence* will work to produce the next line.

Asynchronous Process The asynchronous process occurs alongside the main process. Whilst the agent is generating the next sentence, thoughts and emotions are generated. These contribute to the overall *Internal Agent State* which then affects the processing of other modules. It is this feedback loop which benefits the proactive process. The generation of thoughts is important as it gives the agent the ability to self-reflect on what is said to it, and also what itself says.

Memory Module (Embedding API) The memory module (Fig. 11) handles the structure and retrieval mechanisms of memory, delineating three primary types: core memories, keyword-indexed memories, and geo-temporal-indexed memories. Core memories serve as foundational elements, consistently accessible. In this work, the only core memories we gave to agents were speech mannerisms. Keyword-indexed memories are catalogued using embeddings (OpenAI’s `text-embedding-ada-002 model`) and retrieved via relevant strings. Geotemporal-indexed memories, tied to specific locations and times, are accessed when within their context. Geotemporal memories were omitted from this paper due to the limited scope of the world.

The retrieval process is guided by a *cosine similarity* function. The main advantage is its low complexity, especially for sparse vectors.

$$\text{similarity}_{\cos}(\mathbf{A},\mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n}\mathbf{A}_i\mathbf{B}_i}{\sqrt{\sum_{i=1}^{n}\mathbf{A}_i^2}\sqrt{\sum_{i=1}^{n}\mathbf{B}_i^2}} \quad (1)$$

A useful memory is a) relevant and b) builds upon context. The number of memories retrieved is limited to the best of N, where $N = 5$. 5 was chosen arbitrarily. Memories are retrieved separately for each contextual string and aggregated together to remove duplicates.

The memory module continually takes the internal state of agents *OnLongUpdate* and builds them into memories. These memories hold 2 attributes alongside the memory string: age and strength. Fresh memories are placed in the short-term memory pool, and every time they are retrieved they are strengthened by 1 point. Every *OnLongUpdate* short-term memories are faded. Once the strength reaches an arbitrary number, they become long-term memories and cannot fade away. If memory is faded sufficiently, they are removed from short-term memory and destroyed permanently. This mimics the current general understanding of how memory works in humans [4].

When memories are being retrieved, the memory module draws upon short and long-term memory. It additionally draws upon the *Agents Knowledge*, which can be described as a list of truth strings about the agent, that describe everything the agent should know. The agents’ knowledge is a standalone knowledge base that the agents can use, as described in Section 3.6. This is a significant factor in creating believable human proxies as it creates the illusion of personality and lived experience.

Thought Module (GPT API) We now explore the most significant modules in contributing to the believability of agents. The thought module, like many other modules, takes in the internal agent state and generates new thoughts. The module is very simple, but it is more the emergent capabilities of this module that interest us. One important note is that other modules can append thought strings to the thought module, creating a wealth of thoughts, and not just internal reflections of the agent. Thoughts are generated continuously and aim to mimic a sort of active consciousness. However, one limitation of GPT models appears to be their tendency for repetition. When placed into infinite feedback loops in isolation, thoughts tend to stagnate. With the addition of external stimuli, such as new context or new inputs from users, thoughts remain relevant and interesting. Using more powerful models like GPT-4 additionally helps with generating varied thoughts.

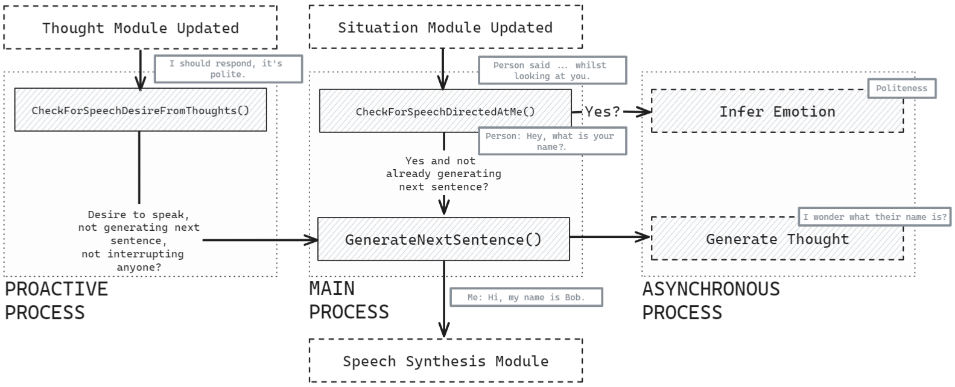
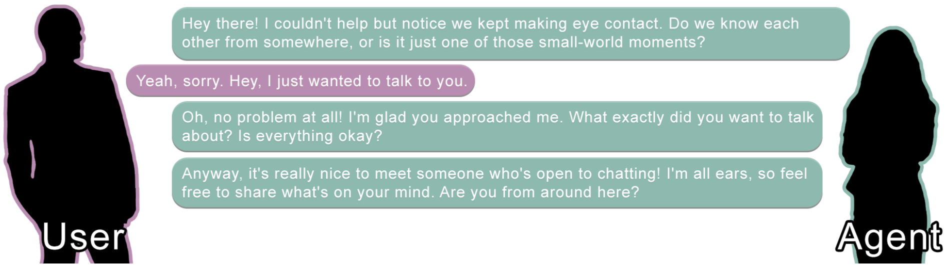

Thought Module Updated
I should respond, it's polite.
CheckForSpeechDesireFromThoughts()
Desire to speak, not generating next sentence, not interrupting anyone?
PROACTIVE PROCESS
Situation Module Updated
Person said ... whilst looking at you.
CheckForSpeechDirectedAtMe()
Person: Hey, what is your name?.
Yes?
Yes and not already generating next sentence?
GenerateNextSentence()
Me: Hi, my name is Bob.
MAIN PROCESS
Speech Synthesis Module
Politeness
Infer Emotion
I wonder what their name is?
Generate Thought
ASYNCHRONOUS PROCESS

Figure 9. The main information flows go through the conversation module. The flows are split into 3 processes, each operating independently.


Hey there! I couldn't help but notice we kept making eye contact. Do we know each other from somewhere, or is it just one of those small-world moments?

Yeah, sorry. Hey, I just wanted to talk to you.

Oh, no problem at all! I'm glad you approached me. What exactly did you want to talk about? Is everything okay?

Anyway, it's really nice to meet someone who's open to chatting! I'm all ears, so feel free to share what's on your mind. Are you from around here?

Figure 10. Example conversation with the thought module enabled. The user is standing opposite the agent, the agent proactively asks the user why they are staring. Information about the state of the user has been obtained from the agents' internal world model.

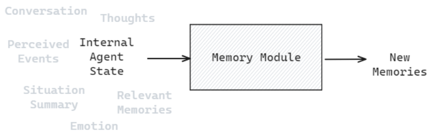


Figure 11. The memory module.

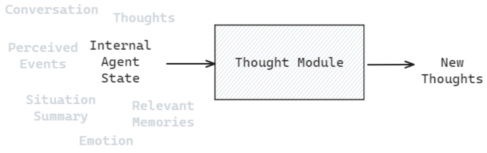


Figure 12. The thought module takes in the internal agent state and generates new thoughts.

Emotion Module Emotion modules have been studied with emotional behaviour trees, and more recently, the enhanced notion of behaviour trees to promote empathetic human-agent interaction [11, 6]. The emotion module is the last major module in the GPT-NPC architecture (Fig. 13). The problem of figuring out how to change the emotional state of an agent given conversational input is not trivial. It first involves inferring the emotion of the conversation line, and then deciding what the resultant emotional effect should be. These 2 problems have been tackled without using GPT models in our structure.

Emotional Inference via Vector Databases Initially the problem of tackling sentiment analysis to infer emotions seemed something best suited to GPT. However, to avoid more calls to the API, which slow the architecture down, another approach was used. Emotion keyword strings can be embedded into a vector database and can be used to infer the emotion of a piece of text. This discovery is not something new [27, 34, 26]. There are directions in the latent space of the embedding model that correspond to certain emotions. In this case, the model used was OpenAI's `text-embedding-ada-002 model`. We can infer emotions from the user's speech using this technique. To achieve this we use a list of emotion keywords from a classification of 48 emotions [1]. It is also important that the input sentence does not contain the emotion keyword, otherwise, the embedding model weighs that emotion incredibly high (even if the emotion isn't present in the sentence's sentiment) as appearance in the text is very strongly encoded in the latent space. Trivial text filtering was used to solve this issue.

External Dialogue
Emotion Module
Emotional Affect

Figure 13. The emotion module infers the emotion of the conversation.

Emotional Response using Game Theory A conversational interaction with emotions involved can be modelled using game theory, such as the *Prisoners Dilemma.* The agent can either collaborate emotionally or defect emotionally. A collaboration is to match the user's emotion (out of sympathy) and a defect is the opposite. Traditionally for the game, we can apply various strategies [3]. The most successful according to a simulated tournament done in the 1980s,

---

[1] Emotion annotation and representation language: https://socialselves.wordpress.com/wp-content/uploads/2013/03/earl.pdf

is the "tit-for-tat" strategy, which boils down to mirroring the opponent. This leads to the behaviour of the agent mirroring the perceived emotion of the user. This is something humans often do when talking to people too [20]. The reason tit-for-tat is successful with the Prisoners Dilemma, especially for repeated games is that it is: a) Nice, b) Forgiving, c) Retaliatory, and d) Clear [19]. These qualities extend well to emotional conversations with the assumption made that the agent is trying to be a benevolent agent who wants to maximise the gain for both individuals. One can imagine applying other more deceptive strategies to make agents that can ultimately deceive, lie and manipulate.

Each of these 48 emotions is grouped into categories. These categories were used to create expressions for the agent. Ideally, each emotion would have its expression with various microexpressions contributing to a more comprehensive expression set. The categories can be seen below with their associated expression. See Figures 14 and 15.

Speech Synthesis Module The speech synthesis module uses OpenAI's Text-to-Speech models to turn the agents' generated dialogue lines into heard speech. One limitation here is that the model does not support emotions, and struggles with non-straightforward formatting. *ElevenLab's* speech synthesis [1] are better suited to the task, but the increased quality comes with considerable cost.

Needs can be viewed as an extension of the existing cognitive framework, where agents derive needs from contextual cues and generate corresponding actions. Although this module is not implemented in the current work, we plan to explore its integration in future research.

3.6 Agent Knowledge & Character Profiles

The last and arguably most important factor when concerning believability is providing agents with a wealth of background knowledge to draw upon. This knowledge is retrieved alongside memories and forms the core of the agent's memories. It imbues the agent with personality and life experience, making it more human than otherwise. Due to the GPT model's inclination to hallucinate, providing concrete knowledge ensures alignment to a target character can be maintained.

Agent Knowledge consists of a series of "truth strings". These are split into several categories for easy comprehension, but ultimately get aggregated and embedded into a single knowledge vector database the agent can pull from. The quality of these truth strings is integral to the agent's believability. The more specific and detailed truth strings are, the more details the agent can draw upon when engaging in conversation. For example: *"I went on holiday to Spain in 2020"* as a truth string alone would not be enough as the GPT model would have to hallucinate to fill in the gaps. Providing 100 truth strings about each hour of the holiday and their thoughts surrounding it would provide a rich experience for the agent to draw from. For this reason, the quality and detail of agent knowledge are directly correlated with believability. Thus creating believable agents requires a serious amount of manual work to sell the illusion of lived experience.

As mentioned earlier agent knowledge is divided into 12 categories:
`Basic Information, Physical Attributes, Personal History, Interests, Likes/Dislikes, Personality Traits, World View, Social Bonds, General Knowledge, Specific Knowledge, Flaws, Inabilities.`

Inabilities are a notable category. They ensure the agent does not try to draw upon knowledge encoded into the GPT model that the character wouldn't know. An example of an Inability is: *"I don't know how to code"*. When the agent is prompted with the question *"How do I reverse a linked list"*, the agent will simply respond with *"Sorry, I don't know how to code..."*. This greatly helps with alignment. This is a difficult problem in the space of AI, especially due to the nature of building a framework on top of a model that already has a large ocean of encoded knowledge that no single human could know at once.

3.7 User interaction & Environment

The user interface into the virtual world is through a VR headset (Fig. 16). The GPT model is integrated into a virtual game world, where the player interacts through a VR headset. The player's position and movements are reflected in the virtual environment, creating an immersive experience. However, the agent can be interacted with through a traditional screen interface and the architecture does not depend on the interaction layer. The distinct layers are seen in Fig. 2. The headset used to facilitate the interaction is also unimportant. Here we used a PICO VR headset.

Environment Implementation The environment chosen to trial the architecture was a school classroom. This was chosen as it is a location familiar to the test subjects: largely students. It also provides an easy-to-understand social context for the agent to inhabit. It naturally follows that the agent would be a school student. A suitable model was chosen, one that allowed for expressions to be changed. A classroom environment was additionally obtained. See Fig. 17.

4 Evaluation

In this section, we present the trials and experiments that demonstrate the architecture's capabilities.

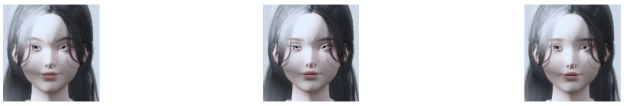
4.1 Local Evaluation

The architecture was first evaluated locally by the designer. The agent's conversational abilities were assessed. Overall the agent can maintain a consistent conversation, remaining aligned to their character. When thoughts were enabled, agents proactively started conversation building the illusion that they have conscious free will. With emotions enabled, the agent's expressions change throughout the conversation in a meaningful way. For example, when a sad topic was brought up, agents would start to frown and vice versa.

4.2 Controlled Evaluation

Results A more numerical approach to validating its performance is needed. Controlled Evaluations were carried out to score the agent based on 6 categories. *Ability to Produce Natural & Coherent Dialogue*, *Ability to Remember Previous Events*, *Awareness of Environment*, *Ability to Display Believable Emotions*, *Ability to Display Autonomous Behaviour*, *Acceptable Latency for Real-Time Conversation*.

Evaluation Procedure The evaluation procedure involved extensive screening of potential participants to ensure certain factors like motion sickness were not a hindrance. Participants were screened in two categories: *Motion Sickness* and *Interactive Media & Gaming Familiarity*. The latter is to ensure subjects have a certain level of



Positive & Lively Positive & Quiet Positive Thoughts

Figure 14. Positive Emotional Affect

Negative & Passive Negative & Forceful Negative & Thoughts Negative & No Control

Figure 15. Negative Emotional Affect

Figure 16. User with PICO 4 VR headset.

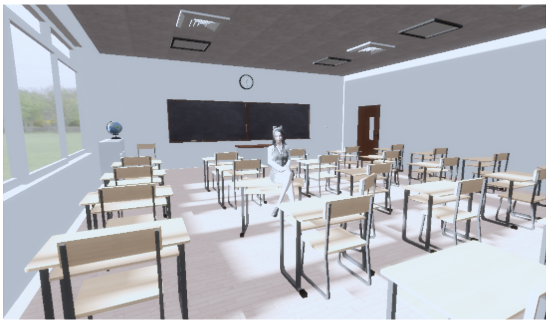
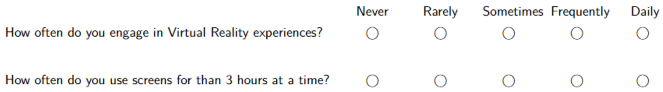


Figure 17. Virtual Classroom Environment with an agent "Morrowii" sitting down in the centre.

technological literacy to ensure a smooth trial. All 15 participants screened were eligible for the trial [2]. The criteria to be eligible were done on a situational basis rather than numerically, due to the nature of the survey. The survey used a Likert scale, see Fig. 18.



	Never	Rarely	Sometimes	Frequently	Daily
How often do you engage in Virtual Reality experiences?	○	○	○	○	○
How often do you use screens for than 3 hours at a time?	○	○	○	○	○

Figure 18. A subset of the questions given in the eligibility survey. The full survey has 32 questions in total.

Eligible participants were then given an additional Tendencies Profiling Survey which obtained tendencies relating to immersion in media and problematic digital media use. This was purely to create profiles for each participant to inform the findings in the final results. After participants went through screening and surveys, trials were conducted. Participants were asked to converse with the agent in 3 configurations for 5 minutes at a time. These configurations were all permutations of the thoughts & emotion modules being enabled/disabled. Using “X” for disabled, and “o” for enabled, configurations are represented as *Thoughts[X] Emotions[X]*, *Thoughts[o] Emotions[X]*, *Thoughts[o] Emotions[o]*. In each configuration, the participant is asked to give scores on six aspects, each has a rank from 0 (absence of the desired functionality) to 5 (optimal performance):

- *coherence* - is the GPT agent being logical and consistent?
- *memory* - how well does the GPT agent remember the context?
- *memory* - how well does the GPT agent remember the context?
- *awareness* - is the agent knowledgable and aware of the situation?
- *emotions* - can the GPT agent understand emotions?
- *autonomy* - can the agent behave autonomously?
- *latency* - how do you score the experienced latency in this interaction?

This was done to gauge the impact these modules had on the believability of agents. Experiments are conducted in April 2024.

Assume W_i is the weight assigned to each success criterion, where i ranges from 1 to 6. S_i represents the category score normalised between 0 and 1, as rated by the participants. The overall evaluation score (E) for an objective was calculated using the weighted average formula:

$$E = \frac{\sum_{i=1}^{6} W_i \cdot S_i}{\sum_{i=1}^{6} W_i} \tag{2}$$

This weighted approach allows for the prioritisation of certain criteria based on their perceived importance. The weights are assigned based on the relative importance of each success criterion. The maximum achievable score is 5, indicating optimal performance, while a score of 0 implies the absence of the desired capability.

From local testing, the weights were determined to be:

---

[2] Participants are recruited from the University of Exeter student community, ranging in age from 18 to 26.

Table 1. Evaluation Results

		No Thoughts, No Emotions	**Thoughts, No Emotions**	**Thoughts, Emotions**
Success Criteron	Coherence	0.8	0.8	0.8
	Memory	0.6	0.6	0.6
	Awareness	0.5	0.65	0.65
	Emotions	0.6	0.7	0.85
	Autonomy	0.71	0.83	0.83
	Latency	0.49	0.37	0.35
Evaluation Score		**0.6**	**0.616**	**0.63**

Table 2. Weights of Different Criterion for Testing

Success criterion	Symbol	Weight
Coherence	S_1	8
Memory	S_2	6
Awareness	S_3	7
Emotions	S_4	5
Autonomy	S_5	2
Latency	S_6	10

These weightings were chosen arbitrarily, but to reflect the relative importance of each success criterion. Latency was the most significant factor in trials and local testing. Aggregating all trial scores together with the weights in place, the scores are shown in Table 1.

4.3 *Analysis*

These results show a clear improvement when agents are given the ability to think and feel emotions. However, latency remains to be a big issue concerning the believability of agent interactions. The latency falls between 2 - 4 seconds which creates awkward pauses in conversation. This delay becomes worse with increased complexity from thoughts and emotions. All participants reported latency being the most significant hindrance in the experience, which is likely to be the most significant factor in creating believable interactions, as opposed to the depth of the interaction. A significant portion of the overhead could be mitigated by using local models, one that runs on special hardware such as a GPU but for processing AI models. API calls are too slow and the reliance on an internet connection severely limits their use.

Some participants were more receptive and forgiving of latency delays due to an empathetic understanding of the underlying complexities present in the architecture. Most trial participants did not engage in any more than a superficial conversation with the agent likely due to time constraints and latency issues. A larger scale of trails is needed to conclude the effectiveness of the architecture as a whole and to have a comprehensive evaluation of the framework.

5 Conclusion

Our GPT-NPC architecture in general showed mixed success in creating believable proxies of human behaviour. Certain important qualities of human interaction that feed into believability only became apparent after extensive trials. These factors are latency and cynicism. Reducing the time to respond to conversations with partners would significantly increase the immersion and suspension of disbelief. Simpler architectures may be useful in this regard as it is unclear whether the tradeoff between communication overhead and conversational ease is worth it. In some contexts, this delay can be natural, or expected even. For example in textual mediums, one is limited by how fast they can type; and many don't expect to receive responses immediately. This increases the believability as it aligns with behaviours observed in humans.

We conclude that the GPT-NPC architecture, although pointing out a direction, has a long way to go before it can get close to becoming a believable proxy for humans. Its modular approach did show some benefits and is a major takeaway: the mix of experts' approaches shows promise in combining lesser models to create a greater whole. Within the realm of video game entertainment, players have come to expect NPCs to be inhuman, which allows these models to operate within achievable bounds. The cost of such models comes into mind, only mitigated again by local models or other low-resource AI models. The abilities of thinking and emotive agents will have to be explored more.

Overall, The GPT-NPC architecture has sparked some interesting conversation and highlighted the important attributes that contribute to the believability of NPCs for video games. One important step towards believably is to apply speech synthesis tools with the ability to interpret emotions, such as *ElevenLab's* speech synthesis [1], to create more believable interactions.

References

[1] URL https://elevenlabs.io/app/speech-synthesis.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] R. Axelrod. Effective choice in the prisoner's dilemma. *Journal of conflict resolution*, 24(1):3–25, 1980.

[4] N. Cowan. What are the differences between long-term, short-term, and working memory? *Progress in brain research*, 169:323–338, 2008.

[5] G. M. de Buy Wenniger and A. Houtkooper. Goap. 2008.

[6] P. Dell'Acqua and S. Costantini. Empathetic human-agent interaction via emotional behavior trees. *Intelligenza Artificiale*, 17(1):89–100, 2023.

[7] J. Duncan. Selective attention and the organization of visual information. *Journal of experimental psychology: General*, 113(4):501, 1984.

[8] U. Götz. On the evolution of narrative mechanics in open-world games. *Media Studies| Volume 82*, page 161, 2021.

[9] C. E. Haupt and M. Marks. Ai-generated medical advice—gpt and beyond. *Jama*, 329(16):1349–1350, 2023.

[10] Z. Jing, Y. Su, Y. Han, B. Yuan, C. Liu, H. Xu, and K. Chen. When large language models meet vector databases: A survey. *arXiv preprint arXiv:2402.01763*, 2024.

[11] A. Johansson and P. Dell'Acqua. Emotional behavior trees. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 355–362. IEEE, 2012.

[12] C. Jones and B. Bergen. Does gpt-4 pass the turing test? *arXiv preprint arXiv:2310.20216*, 2023.

[13] C. R. Jones and B. K. Bergen. People cannot distinguish gpt-4 from a human in a turing test. *arXiv preprint arXiv:2405.08007*, 2024.

[14] A. Kaplan and M. Haenlein. Siri, siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence. *Business horizons*, 62(1):15–25, 2019.

[15] W. G. Kennedy. Modelling human behaviour in agent-based models. In *Agent-based models of geographical systems*, pages 167–179. Springer, 2011.

[16] M. Lewis. The emergence of consciousness and its role in human development. *Annals of the New York Academy of Sciences*, 1001(1):104–133, 2003.

[17] E. d. Luna. Character.ai: What it is and how to use it, May 2023. URL https://mashable.com/article/character-ai-generator-explained.

[18] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, and C. Callison-Burch. Faithful chain-of-thought reasoning. *arXiv preprint arXiv:2301.13379*, 2023.

[19] D. Muller. What game theory reveals about life, the universe, and everything, Dec 2023. URL https://www.youtube.com/watch?v=mScpHTIi-kM.

[20] C. Navarretta. Mirroring facial expressions and emotions in dyadic conversations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 469–474, 2016.

[21] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[22] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[23] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[24] P. P. Ray. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, 2023.

[25] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[26] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, and M. Sedlmair. More than bags of words: Sentiment analysis with word embeddings. *Communication Methods and Measures*, 12(2-3):140–157, 2018.

[27] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou. Sentiment embeddings with applications to sentiment analysis. *IEEE transactions on knowledge and data Engineering*, 28(2):496–509, 2015.

[28] I. Umarov and M. Mozgovoy. Believable and effective ai agents in virtual worlds: Current state and future perspectives. *International Journal of Gaming and Computer-Mediated Simulations (IJGCMS)*, 4(2): 37–59, 2012.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[30] L. Wang, M. I. Mujib, J. Williams, G. Demiris, and J. Huh-Yoo. An evaluation of generative pre-training model-based therapy chatbot for caregivers. *arXiv preprint arXiv:2107.13115*, 2021.

[31] R. Wang, S. An, M. Cheng, T. Zhou, S. J. Hwang, and C.-J. Hsieh. Mixture-of-experts in prompt optimization. 2023.

[32] R. Wang, E. Zelikman, G. Poesia, Y. Pu, N. Haber, and N. D. Goodman. Hypothesis search: Inductive reasoning with language models. *arXiv preprint arXiv:2309.05660*, 2023.

[33] T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, and C. J. Cai. Promptchainer: Chaining large language model prompts through visual programming. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–10, 2022.

[34] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 534–539, 2017.