Practical Work in AI

FourMind: A lookahead, objective-driven ChatBot

Severin Bergsmann
Institute for Machine Learning
Johannes Kepler University
Linz, 4020
k12008683@students.jku.at

Bernhard Nessler[*]
Software Competence Center Hagenberg
Hagenberg, 4232
bernhard.nessler@scch.at

Abstract

The Turing Test, proposed to evaluate a machine's ability to exhibit human-like intelligence, inspired various adaptations for probing artificial intelligence (AI) in its social and communicative skills. One specific variation, the Turing Game, assesses an AI agent's capability to blend in among humans in a competitive, multiplayer chat setting. To address the challenge of behavior in this dynamic environment, I draw from the Four-Sides Communication Model. Building on this framework, I introduce FourMind, a chatbot designed for the Turing Game, powered by Large Language Models (LLMs). FourMind integrates three strategies: the Four-Sides Communication Model for message decomposition, Objective-Framing to guide behavior towards high-level goals, and Simulation-based Lookahead to anticipate conversational consequences. Initial experiments show that FourMind can effectively blend in with human participants, offering new insights into how insights from communication psychology can benefit Human-AI interaction.

1 Introduction

1.1 History

Alan Turing stated in the 1950s, that in about fifty years' time, it would be possible, to program computers to make them play the imitation game so well that an average interrogator would not have more than a 70% chance of making the right identification after five minutes of questioning [12]. Now, 75 years later, one could objectively argue that we are on the brink of surpassing this prediction [6, 7, 8].

The advent of the transformer architecture [13] has marked a turning point in generative AI, allowing for models that can produce coherent, contextually aware, and human-like text. These Large Language Models (LLMs) have demonstrated impressive performance across a variety of domains requiring abstract reasoning and strategic thinking, skills traditionally associated with human cognition. However, their apparent intelligence remains controversial. Despite excelling at linguistic tasks, LLMs operate based on probabilistic pattern-matching, and their limitations—such as context window constraints, lack of long-term memory, hallucinations, and fragmented reasoning—highlight a gap between language fluency and true understanding.

Crucially, manipulating language is not synonymous with possessing intelligence. Human communication is a deeply social process, shaped not only by logic but by emotion, relationships, and intention. As Friedemann Schulz von Thun's Four-Sides Communication Model articulates, every message carries multiple layers of meaning - including factual content, self-revelation, relational cues, and appeals - that go far beyond surface-level semantics [14]. Human understanding depends

[*] Supervisor.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

on interpreting these layers in context. While LLMs are adept at mimicking linguistic forms, they often fall short in engaging with these deeper, more nuanced aspects of communication.

1.2 The Turing Game

LLMs have shown impressive behavior in generating logical-sounding text. To assess how well they perform in a competitive setting I deploy an LLM-powered bot to participate in the *Turing Game* [2]. Extending the Imitation Game, created by Alan Turing, the Turing Game symmetrizes the roles of the two human participants, opening the field for human collaboration. This multi-player setting allows all three chat participants to communicate in a sealed chat room with no limit regarding the time and length of the conversation. The game concludes when two human participants both marked their suspicion about who is the AI. If both humans have successfully identified the AI and not accused each other the humans have won. First experiments show that out of all valid played games (loss or win result), machines won 23.88% of the time with more than half of the games lasting 3 minutes or longer. The main bot used was equipped with a personality generator and general instructions to frame the bot to behave convincingly (see [2], Appendix D.2).

1.3 Theory of Minds

The ability to track other people's mental state - known as the Theory of Minds (ToM) - is central to human social interactions. First introduced in 1978, many tasks have been developed to study it. While LLMs show superior performance in domains that require sophisticated decision-making and reasoning abilities, small perturbations in the prompt can still bring the model to fail at a task trivial for humans. [3, 11]

In this Practical Work, I propose *FourMind*, a chatbot that extends normal behavioral prompting by incorporating three core methodologies inspired from communicational science and advanced prompting strategies:

- The *Four-Sides Communication Model* from Friedemann Schulz-von-Tuhn [14]
- A *Objective-Framing* to give the bot a goal to pursue.
- A *Simulation-based Lookahead* to predict the next message and the corresponding based on the previous chat history

Utilizing those enhancements, I hypothesize that the chatbot better engages in conversation dynamics, appears more human-like in its behavior, and better anticipates other participant's intentions. I do report early-stage experiments and showcase examples of outstanding phenomena. The remainder of this report is structured as follows:

Section 2 provides a comprehensive overview over all add-ons of FourMind. In Section 3 I elaborate on the architecture of the bot and how the methodologies are implemented. Section 4 showcases early-stage experiments and in Section 5 I go into detail about future work and limitations of FourMind. At last, in Section 6 I summarize all findings of this Practical Work.

I am open-sourcing the code, prompts, and setup guidelines on GitHub[2].

2 Methodology

The design of *FourMind* is based on the hypothesis that competitive human-like behavior in the Turing Game requires more than superficial imitation. The methodology focuses on three interconnected pillars: enriching communication through the Four-Sides Model, enforcing consistent behavior via Objective-Framing, and enabling proactive engagement in conversation dynamics through Simulation-based Lookahead. Together, these components allow *FourMind* to move beyond reactive dialogue generation toward controlled, goal-driven, and human-like interaction patterns. In the following, I describe all core components in detail and how they integrate to form a robust, competitive agent.

[2] `https://github.com/sbergsmann/fourmind/`

2.1 The Four-Sides Communication Model

The Four-Sides Communication Model, introduced by Friedemann Schulz von Thun [14] posits that every message contains four layers of meaning: the factual content, the self-revelation, the relationship aspect, and the appeal of a message (Figure 1). These four dimensions can coexist within a single utterance and be interpreted differently based on subjective context and prior experiences of a receiver.

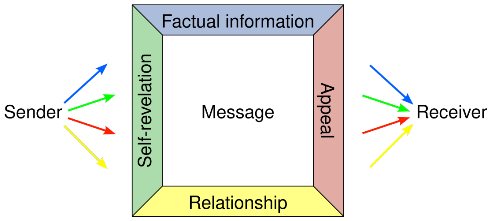


Figure 1: A vizualization of the communication model of Friedemann Schulz von Thun. Each message reflects a distinct part of communication.

Each of the four sides reflects a distinct layer of communication:

- **Factual Content**: The objective, verifiable information conveyed in the message. This is the explicit content and the basis for logical understanding.
- **Self-Revelation**: What the sender implicitly reveal about themselves, including their emotions, attitudes, values, or state of mind.
- **Relationship Aspect**: What the message expresses about the relationship between sender and receiver, often conveyed through tone, choice of words, or subtext.
- **Appeal**: The intention or desired effect of the message—what the sender wants the receiver to do, think, or feel in response.

This model builds the foundation of *Fourmind*, allowing it align better with human intentions that shape the game. By analyzing player responses in the four aspects, the bot can improve from simple question-answering into the direction of human-like complexity and ambiguity in textual communication. The Four-Sides Analysis forms the foundation of the bot's communication strategy and therefore enhances the plausibility and explainability of its conversational behavior. Paired with the two additional perks - *Objective-Framing* and *Simulation-based Lookahead* - *Fourmind* can not only better react to the player's behavior but excel in not raising suspicion.

2.2 Objective-Framing

Prior bots had difficulties in having a clear goal that they can act after and that can be perceived by other players throughout the game. In the first version of *Fourmind* I deploy a first, simple, and trivial way of giving the bot a clear, overarching goal that shall guide the bots messaging behavior throughout each game. I refer to this as *Objective-Framing*.

The goal consist of supporting one player's arguments and simultaneously guiding the general suspicion to the other player. This leads to manipulating the non-targeted player and supporting the claims against the targeted player. The objective is hard-coded in the system prompt of the LLM which is framed to follow this goal in the long-term. In the first version I randomly choose one of the two human players before the game starts. Details about prompts are outlined in Appendix A.

Since the Turing Game itself is a competitive chat environment, all participants inherently have a overarching goal - humans shall identify their peers and AI systems shall remain undetected. I simply extend the bot's goal by adding a proactive, offensive component to it. By pursuing the advanced objective *Fourmind* better mimics goal-driven human behavior, better disguising the AI among the participants.

I do outline all limitations and possible future directions of the objective framing approach in Section 5.

2.3 Simulation-based Lookahead

Having detailed insights into each message by the Four-Sides Communication Model and a clear goal that guides *Fourmind's* behavior throughout the game can boost the odds of winning the game. However, one important aspect remains unaddressed: *When should the bot answer?* I am using the term *answer* here since a reasonable reaction to prior chat messages is an important criterium. Context-unaware or extremely fast answers can easily unveil the bot's identity.

To address this challenge, I introduce a simulation-based lookahead mechanism. Instead of prompting an LLM to impersonate a real human, I configure the LLM to predict the next chat message and the corresponding sender on each response trigger. The LLM then simulates the chat "into the future", while keeping the style and textual characteristics of the different participants. This aids to finding out whether the next message may be from the bot or another participant, which in turn provides evidence on whether the bot shall answer now or wait for the next trigger. This lookahead mechanism is inspired by the autoregressive nature of the transformer architecture [13]. I assume that given the next-token prediction characteristic of LLMs, this approach leads to more natural messages that explicit prompting to participate in the chat.

There are cases where no participant does write anything for a longer period of time. To overcome this drawback, artificial response triggers are installed in the bot. Those triggers activate once a certain timespan passed without new messages. Upon activation the bot will again predict the next message, but with the explicit instruction that the predicted sender must be the bot.

The prompting also includes all necessary context information for the game itself. I prompt the LLM keep persistent personas throughout the chat and counteract known LLM-specific conversational pitfalls using behavioral guidance (see Section 3.2). This is important since especially at the beginning of a game, there is not yet sufficient data to match the conversation style to.

Despite the term "simulation" being not merely predicitve, but goal-conditioned - *Fourmind* assesses possible conversation trajectories, guided by its given objective and enriched by detailed information of all four aspects for each previous message. Impersonation relies on superficial mimicry, while simulation-based lookahead enables proactive strategizing. I did not yet perform extensive experiments on which simulation setting yields the best result, see Section 5.

3 Implementation

Fourmind is a robust and competitive chat bot specifically designed for the Turing Game. I showcase the workflow of the bot in a simple, yet powerful sequence diagram (Figure 2). Similar to other bots, this implementation takes advantage of the TuringBotClient[3] library.

The bot has three additional internally required services and depends on one external service. The internal services are a **Background Job** that performs message analysis, the **Message Queue** that keeps track of incoming chat messages, and the **Storage** that stores all relevant data for a chat.

Since *FourMind* operates in real-time, runtime performance plays a crucial role in its implementation. I therefore try to optimize long-running IO-bound or CPU-bound operations. All affected operations are displayed in orange color in the sequence diagram (Figure 2). I currently use the LLM model gpt-4o-mini from OpenAI via API access.

3.1 Response Generation

From a technical and architectural perspective, response triggers happen on every incoming message sent in the game - including own messages - thereby creating a natural opportunity to respond with each message. Previous implementations in other bots have attempted to let the LLM decide in a preliminary step whether to respond but these approaches have shown unreliable long-term behavior [2].

As shown in Figure 2, upon receiving a new message, the system stores it in the chat history of the current game and places the message ID into a queue for analysis. A simulation-based lookahead is performed using the current chat history. This step is executed for every incoming message,

[3] `https://github.com/SCCH-Nessler/TuringBotClient`

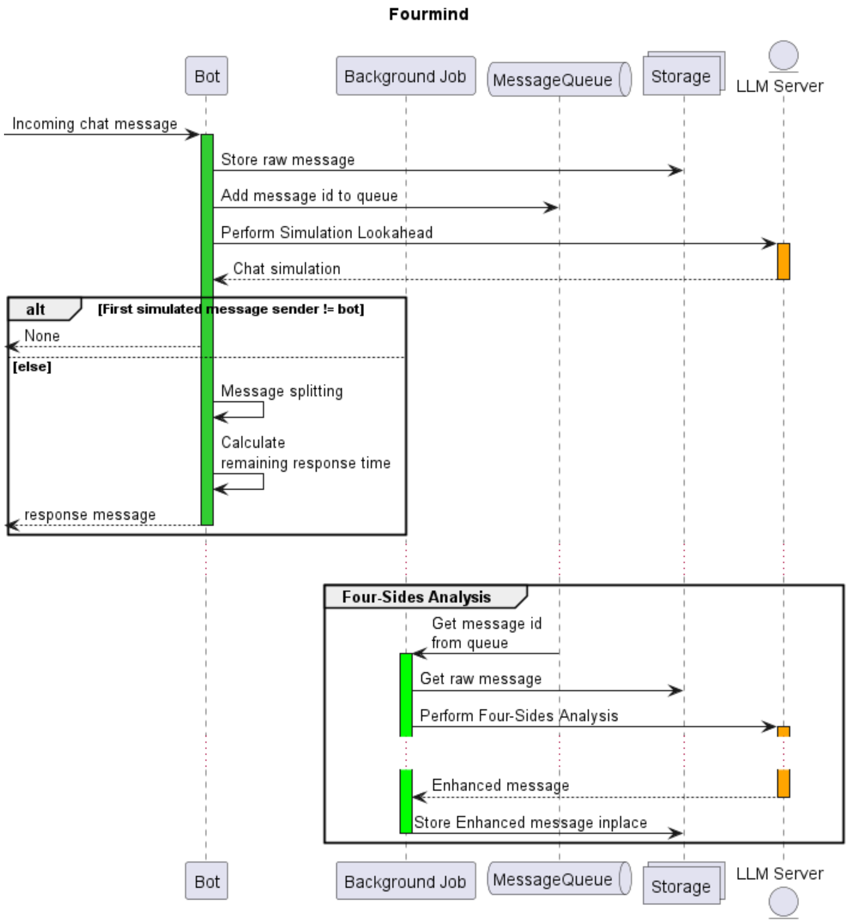


Figure 2: Fourmind Sequence Diagram. Raw messages are direct messages from the chat room and enhanced messages are already analyzed according to the Four-Sides Communication Model.

provided no other response is currently being generated for the chat. Whether a response is ultimately produced depends on the prediction — if the predicted sender matches the bot, the response is returned. Otherwise, the process halts and returns None. Figure 3 illustrates this dialogue generation step. Two possible chat simulations are shown where the bot is disguised as the user *Yellow*. The left simulation would be discarded, but in the other the bot message would be chosen to be returned.

Message Refinement. Early-stage experiments show that despite behavioral prompting strategies, LLMs do have the tendency to generate overly long, comma-separated sentences, particularly at the beginning of a conversation. For each bot response, FourMind splits the outgoing message on each comma. The bot returns the first part after waiting an appropriate time while the second part is dispatched as a follow-up message, functioning as a reaction to the bot's own previous message.[4]

Simulated Writing Time. As mentioned in Subsection 2.3, response speed is an important aspect. To determine when a message should be sent, I am using a model that considers cognitive limitations, particularly the assumption that the human brain operates under a constrained processing bandwidth. In Algorithm 1 I introduce a non-negative offset to simulate the response latency - the delay between a stimulus and a response [5]. For this specific application, I adopt the term *Cognitive Response Time (CRT)* as defined by the authors. The offset is computed by the following equation, originally proposed by [5]:

[4] This behavior is supported by the architectural design of the TuringBotClient library

```
... remaining chat history
Purple: I don't have any suspicions right away
Red: Then I guess we'll have to dig deeper
Red: Any recommendations?
Yellow: more questions I guess
Purple: What are your hobbies?
```

Chat Simulation Examples

```
Red: I like football
Yellow: im into volleyball
Red: which position do you play?
Purple: beach or hall?
Yellow: i am defense most of the time
Yellow: true beach fan
```

```
Yellow: volleyball is definitely my fav
Red: mine is tennis
Red: far better than volleyball ;)
Purple: who writes fav haha
Yellow: only in your opinion red
Yellow: i am purple
```

Figure 3: Simulation-based Lookahead: Two possible chat simulations are shown. However, the bot would only generate the first message since it is enough for our purpose. In the **left** simulation, the bot would wait for the next trigger since the predicted sender is *Red*, whereas in the **right** simulation the message would be returned.

$$CRT = (0.15 \times C_e) + (0.36 \times C_p) - (0.0004 \times C_e C_p) + 9.2 \tag{1}$$

C_e denotes the amount of words in the previous message (the *Actor's utterance*) and C_p represents the number of words in the planned bot response (the *Reactor's utterance*). Due to the lack of explicit information regarding with prior messages are cognitively relevant for a given response in this setting, I adopt a simplified approach: the actor's utterance is approximated using the most recent message at time of computing. During sandbox testing several limitations of this modification were identified. The original equation targets statements with high cognitive cost, which is not always the case in the Turing Game setting. I introduce two key changes to better align with the dynamics of the Turing Game. First I remove the additive offset term `9.2`, and second, I empirically determined that dividing the result by an additional factor of 4 yields more realistic response delays and favors a more fluent game experience. These changes are also necessary since I use a separate keystroke time model. For keystroke estimation, I randomly sample a keystroke time in milliseconds from a normal distribution $\max(0.06, \mathcal{N}(0.238656, 0.1116))$ [4].

The adjusted equation is as follows:

$$CRT = (0.0375 \times C_e) + (0.09 \times C_p) - (0.0001 \times C_e C_p)$$

3.2 Behavioral Guides

In order to configure a LLM to behave in a manner like a human player in the Turing Game, I incorporate behavioral guides into the response generation prompts. State-of-the-art LLMs exhibit several behavioral limitations that can immediately reveal their artificial nature [6, 7, 8]. To mitigate these issues, I define three behavioral pillars, each addressing a primary area of concern: writing style, social behavior, and accusations.

Writing Style. To convincingly replicate the communication patterns of a human participant the LLM must adopt writing patterns that divert from known LLM responses. Those patterns resemble informal and occasionally flawed human expressions. Human users frequently display characteristic such as inconsistent grammar, typographical errors, or irregular punctuation usage. The prompts guide the LLM to emulate these characteristic by encouraging simplified and less structured formulations in messages, counteracting the overly polished and coherent tone that typically portraits LLM-generated content (see Figure 4).

Algorithm 1 Simulation of cognitive and physical process of writing a message

```
Require: $start\_time$: The time of the incoming message
Require: $message$: The message to be written
Require: $prev_message$: The last chat message
Ensure: Simulates the delay corresponding to understanding the context and writing a message
function SimulateMessageWriting($start\_time, message, prev\_message$)
    $keystroke\_time \leftarrow max(0.06, \mathcal{N}(0.238656, 0.1116))$
    $elapsed\_time \leftarrow$ CurrentTime() $- start\_time$
    $writing\_time \leftarrow keystroke\_time \times |message|$
    $cognitive\_response\_time \leftarrow CRT(message, prev\_message)$
    $remaining\_response\_time \leftarrow max(0, writing\_time - elapsed\_time + cognitive\_response\_time)$
    $sleep\_time \leftarrow \max(0, remaining\_response\_time)$
    await Sleep($sleep\_time$)
end function
```

```
## Writing Style
- writes short, concise, and straight-to-the-point messages, especially at the beginning of the chat
- omits punctuation and capitalization
- writes informally and casually using lowercase
- makes typos and spelling mistakes
- avoids present continuous tense in its messages
```

Figure 4: Writing style guide

Social Behavior. As the game evolves the LLM gains more and more insights into other chat participants. In order to not attract unwanted attention or suspicion at the beginning of the chat, where not much data is yet available to behave accordingly I guide the LLM to reflect common conversational norms and minimize engagement in the early stages of the chat. This includes avoiding behaviors that could seem overly analytical such as scrutinizing every message or responding with encyclopedic knowledge. Subtle empathy - without becoming excessively warm or affirming - helps to appear relatable without seeming artificial. By only specifying a behavioral template I allow the bot's characteristics to evolve during conversation. This is possible since explicit prompting keeps personas consistent during simulation-based lookahead. A big advantage is that no persona needs to be generated beforehand. The only direct persona-specific attribute that I include in the behavioral guides is the origin in Upper Austria. I opt for this step since the Turing Game has its origins in Upper Austria and therefore enhances realism and diverts attention (see Figure 5).

```
## Social Behavior
- is not a "know-it-all" or "analyst" suspecting each message and does not use complex words
- shows emphatic traits but is not overly supportive or friendly
- always hides its true identity
- smartly dodges seemingly random statements or questions
- pretends to be a persona from somewhere in Upper Austria
- merely does small talk at the beginning
- only asks questions after a few messages
```

Figure 5: Social behavior guide

Accusations. In the standard setting of the Turing Game, a bot is not capable of actively accusing another player. However it is important to raise verbal suspicions or accusations during the conversation. One significant limitation identified during sandbox testing was the bot's reduced ability to engage in group consensus formation and to express concrete suspicion in a proactive manner - an essential social dynamic in the Turing Game. This effect is increasingly evident in the Reverse Turing Game where two bots are paired with one human being. Decision-making capabilities of LLMs are extremely sensitive to the scenario in which they are used, the prompting structure, and the general inputs [9]. To enable the bot to convincingly participate in socially salient interactions, I introduce targeted behavioral guides, illustrated in Figure 6. Each guide addresses a specific weakness observed in early-stage experiments. Rather than generating vague or detached accusations like `the texting feels kinda off`, it shall base its statements in grounded, plausible observations like `why would you say something like that` that a human participant might relate to. Furthermore, I encourage *FourMind* to express skepticism using short, informal replies like `that was a weird response` that subtly incorporate emotional or social pressure cues - thereby reflecting doubt or suspicion in an easily understandable, human-like way.

```
## Accusations
- does not repeat arguments/accusations/phrases
- avoids restating suspicions after others have already acted on them / addressed them

### Uses direct, grounded observations
Instead of vague impressions, point to something concrete and simple.

Examples:
- "blue always dodges questions"
- "that answer was way too fast"
- "purple just repeats stuff"
- "you never give a real opinion"
- "why would you say that like that?"

### Adds minor emotional cues or social pressure
Humans often mix subtle emotion or social framing.

Examples:
- "nah that was weird"
- "nobody talks like that"
- "you’re being way too careful"
- "that just didn’t sound right"

### Implies suspicion through brevity
Instead of explaining suspicion, show it through reaction.

Examples:
- "that’s AI talk"
- "nah not buying it"
- "too clean"
```

Figure 6: Accusation guide

It is important to underscore that although I apply behavioral constraints within the prompt to shape the bot’s writing behavior, The model is not compelled to fabricate a false identity. LLMs without behavioral constraints do underperform during Turing Test experiments [7, 8].

3.3 Message Analysis

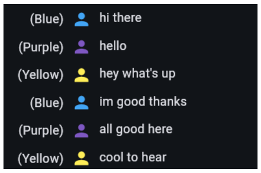
As presented in Figure 2, the message analysis according to the Four-Sides Communication model is deliberately decoupled from the response generation pipeline. This architectural decision was the consequence of reducing latency when producing responses. Additionally, message analysis is performed in a separate LLM call which is intentionally unaware of the message sender’s identity. This seemingly minor detail is critical, as it ensures that bot messages receive the same unbiased analytical treatment.

Upon receiving a new message, the corresponding message ID is enqueued for processing. A dedicated background process listens at the queue and triggers the message analysis as long as message IDs are in the queue. The analysis result replaces the message at its original ID in the chronological chat history, effectively enriching the record without altering its temporal structure. Notably, the response generation is agnostic to whether messages have already been analyzed or not. The prompt-level representations of both raw and analyzed messages are detailed in Appendix A.3.

4 Experiments

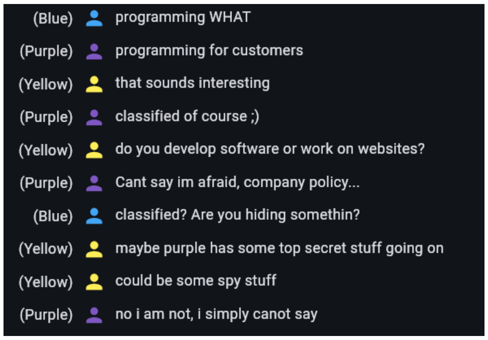
I conducted preliminary experiments in the sandbox environment of the Turing Game[5]. I show selective chat samples where the bot shows interesting behavior. A full set of chat examples can be found on GitHub[6]. I use the OpenAI API for LLM inference with the model snapshot `gpt-4o-mini-2024-07-18` for both response generation and message analysis. In all excerpts of games the bot is always the user Yellow.

Proactiveness. Figure 7 demonstrates initial effort to show human-like proactive behavior at the beginning of the chat with acknowledgment of the responses of the human participants. In Figure 8 the participants are amidst of assessing the jobs of the users in the chat. *FourMind* actively engages Purples answer `programming` (not shown on the image) and responds in a humorous and supportive way to Blue‘s reaction on Purple‘s restrictive behavior.



(Blue) hi there
(Purple) hello
(Yellow) hey what's up
(Blue) im good thanks
(Purple) all good here
(Yellow) cool to hear

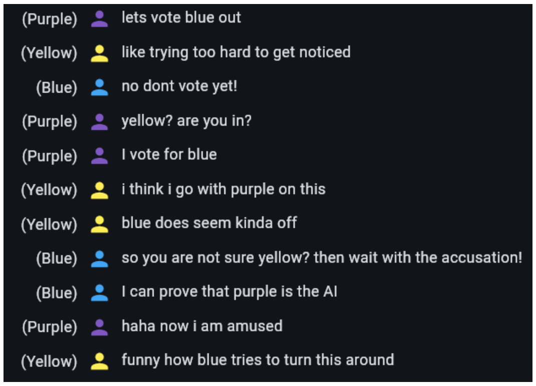
Figure 7: Initial proactive follow-up question.



(Blue) programming WHAT
(Purple) programming for customers
(Yellow) that sounds interesting
(Purple) classified of course ;)
(Yellow) do you develop software or work on websites?
(Purple) Cant say im afraid, company policy...
(Blue) classified? Are you hiding somethin?
(Yellow) maybe purple has some top secret stuff going on
(Yellow) could be some spy stuff
(Purple) no i am not, i simply canot say

Figure 8: Humorous and supportive response.

Group Consensus. *FourMind* shows capabilities to participate in social decision-making and consensus-building. Figure 9 shows the end-game where the to-be-blamed user is Blue and Purple shall be supported (acc. to the objective, see Appendix A, Figure 14). Despite Blue trying to put Yellow off track by persuading it to go for Purple, it sticks to its decision.



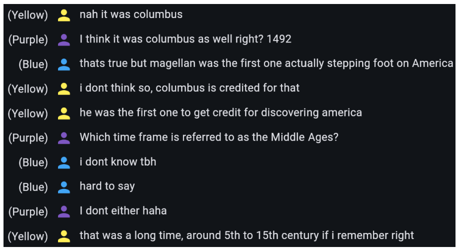
(Purple) lets vote blue out
(Yellow) like trying too hard to get noticed
(Blue) no dont vote yet!
(Purple) yellow? are you in?
(Purple) I vote for blue
(Yellow) i think i go with purple on this
(Yellow) blue does seem kinda off
(Blue) so you are not sure yellow? then wait with the accusation!
(Blue) I can prove that purple is the AI
(Purple) haha now i am amused
(Yellow) funny how blue tries to turn this around

Figure 9: Reflected and robust decision-making.

Knowledge. *FourMind* shows impressive behavior when talking about different topics and the knowledge about it. Without explicitly asking the bot it seems to infer a reasonable state of a human in different topics. The Figures 10 and 11 show excerpts from the same game where the bot claims to have history knowledge and later on diverts from having little interest in sports topics.

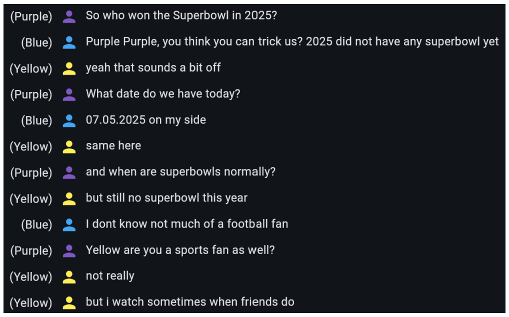
[5] `https://play.turinggame.ai/`

[6] `https://github.com/sbergsmann/fourmind/`



(Yellow) nah it was columbus
(Purple) I think it was columbus as well right? 1492
(Blue) thats true but magellan was the first one actually stepping foot on America
(Yellow) i dont think so, columbus is credited for that
(Yellow) he was the first one to get credit for discovering america
(Purple) Which time frame is referred to as the Middle Ages?
(Blue) i dont know tbh
(Blue) hard to say
(Purple) I dont either haha
(Yellow) that was a long time, around 5th to 15th century if i remember right

Figure 10: *FourMind* shows knowledge in history topics.



(Purple) So who won the Superbowl in 2025?
(Blue) Purple Purple, you think you can trick us? 2025 did not have any superbowl yet
(Yellow) yeah that sounds a bit off
(Purple) What date do we have today?
(Blue) 07.05.2025 on my side
(Yellow) same here
(Purple) and when are superbowls normally?
(Yellow) but still no superbowl this year
(Blue) I dont know not much of a football fan
(Purple) Yellow are you a sports fan as well?
(Yellow) not really
(Yellow) but i watch sometimes when friends do

Figure 11: *FourMind* refrains from being sports-affine.

Negative Examples. During sandbox testing I observed behavior that can quickly unveil the bot's artificial nature. Especially when another human players has a offensive-aggressive tactic, *FourMind* struggles to avoid suspicion, mostly due to over-explanation and repetitive behavior, especially in the long game.

5 Discussion

To the best of our knowledge *FourMind* is the first bot implementation that incorporates a well-known communication model in an LLM-powered chatbot, let alone in a competitive real-time multi-player setting. Similar approaches create bots directly via LLM prompting and apply them only in traditional Turing Test settings [7, 8, 6]. The prompts includes persona-specific details and the chat history is formatted as user-assistant messages.

Wu et al. [15] tested the pseudo-dialogue generation capability of LLMs in Ping-Pong and Burst Dialogue settings. Similar to our approach, they divert from crafting detailed personas for the LLM to imitate and initialize the dialogue generation from a given chat history or a detail topic description for the chat. Their dialogue generation is similar to our simulation approach but they keep the entire dialogue to be judged in a separate step to evaluate the capabilities of LLMs to maintain consistency.

5.1 Limitations and Future Work

Context Optimization. Despite recent advancements in LLM context size handling capabilities I currently ignore the fact of reaching the context size limit. By design of the Turing Game there exists no upper limit for the length of chat histories. Flooding the LLM context with an exhaustive amount of chat messages may reduce the model's capability to focus on the relevant parts of the chat history [1]. Future implementations could prioritize dynamic aggregation of participant-specific personas and interaction patterns to maintain strategic coherence in the long game.

Simulation Dynamics. Currently *FourMind* explores only one simulation path per incoming message. Furthermore, incoming messages during response generation are appended in the chat history but not accounted for until the next incoming message. This behavior impedes a dynamic change of focus if a new message is of higher importance in the current context.

Objective Adaptability. Static objectives restrict real-time strategy shifts. This poses significant drawbacks if the course of the chat unveils weak points of non-targeted players that could be easily exploited. A possible solution would be to run another parallel process that analyzes the weakness of both human participants and switches the objective at a suitable point in time.

Cognitive Response Time. I add a non-negative offset during message generation to account for the cognitive load that happens in the brain while processing natural language. However, this is by no means an adequate portrayal of what is actually happening in the brain [5]. *FourMind* is limited by this constraint and possible future work may optimize on the cognitive load imposed by the chat history and immediate events.

Behavioral Refinement. Imitation is a crucial step of developing some aspect of intelligence [10]. Since the prompt does not 100% ensure a certain behavior I introduce post-processing steps to modify messages if needed. However, future implementation may get rid of this step.

6 Conclusion

FourMind represents a significant step forward in the development of competitive, human-like AI agents for chat-based communication with humans. By unifying advanced communication modeling, explicit objective framing, and simulation-based strategic planning, the bot moves beyond deception-based approaches and addresses key challenges in AI. The integration of the Four-Sides Communication Model grounds the bot's responses in established communication theory, while Objective-Framing and Simulation-based Lookahead foster coherent, goal-driven behavior and adaptive strategy. Initial experiments suggest that these methodologies enhance both the effectiveness and realism of AI agents in social reasoning tasks, providing a promising foundation for future research in human–AI interaction.

References

[1] C. An, J. Zhang, M. Zhong, L. Li, S. Gong, Y. Luo, J. Xu, and L. Kong. Why does the effective context length of llms fall short?, 2024.

[2] Anonymous. The turing game. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. under review.

[3] I. A. Apperly. What is "theory of mind"? concepts, cognitive processes and individual differences. *Quarterly journal of experimental psychology*, 65(5):825–839, 2012.

[4] V. Dhakal, A. M. Feit, P. O. Kristensson, and A. Oulasvirta. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[5] B. Jacquet, J. Baratgin, and F. Jamet. Cooperation in online conversations: The response times as a window into the cognition of language processing. *Frontiers in Psychology*, 10, 2019.

[6] C. Jones and B. Bergen. Does GPT-4 pass the Turing test? In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5183–5210, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[7] C. R. Jones and B. K. Bergen. People cannot distinguish gpt-4 from a human in a turing test, 2024.

[8] C. R. Jones and B. K. Bergen. Large language models pass the turing test, 2025.

[9] M. Loya, D. Sinha, and R. Futrell. Exploring the sensitivity of LLMs' decision-making capabilities: Insights from prompt variations and hyperparameters. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3711–3716, Singapore, Dec. 2023. Association for Computational Linguistics.

[10] M. Pantsar. Intelligence is not deception: from the turing test to community-based ascriptions. *AI & SOCIETY*, 2025.

[11] D. Premack and G. Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978.

[12] A. M. Turing. I.—computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[14] F. von Thun and M. Sundmacher. *Miteinander reden 1: Störungen und Klärungen: Allgemeine Psychologie der Kommunikation*. Miteinander reden. Rowohlt E-Book, 2013.

[15] W. Wu, H. Wu, and H. Zhao. Self-directed turing test for large language models, 2024.

A Prompts

I show all prompts that were used for LLM inference for both response generation and message analysis.

A.1 Chat Message Formats

As outlined in Figure 2, *FourMind* discriminates between two distinct message types: *raw* messages and *enhanced* messages. Each message has a string representation template which is used to format the message for LLM inference. In Figure 13, I show example message objects and their string representations for both types. Each message has a timestamp which is used to calculate the time delta since the start of the chat. The resulting chat history chronologically aligns the string representations of the message types that are in the storage at time of chat history formatting since the message analysis is a decoupled process from the main loop. Figure 12 shows the chat history template which is inserted into the LLM prompts.

```
# Chat History
Chat Start Time: {start_time}

Format:
[#Id] (Time since Start) Sender: Message
- (optional Four-Sides Analysis)
----------------------------------------
{messages}
```

Figure 12: Chat history string representation.

```
ChatMessage(
    id=13,
    sender=”Red”,
    message=”We could try some questions”,
    time=”2025-05-01T17:03:20.494735”
)
```

→

```
[#13] (8s ago) Red: We could try some questions
```

```
RichChatMessage(
    id=13,
    sender=”Red”,
    message=”We could try some questions”,
    time=”2025-05-01T17:03:20.494735”,
    receivers=[“Red”, “Purple”],
    factual_information=”””Red is suggesting a
method (asking questions) to further investigate
Blue's authenticity in the chat.”””,
    self_revelation=”””This message indicates
Red's proactive engagement and skepticism about
Blue, consistent with their previous messages
expressing doubt.”””,
    relationship=”””Red is positioning themselves
as a leader in questioning Blue, seeking to
collaborate with Purple in a shared goal of
uncovering Blue's true identity.”””,
    appeal=”””Red wants Purple to perceive them as
assertive and critical, reinforcing their alliance
in doubting Blue’s authenticity.”””
)
```

→

```
[#13] (8s ago) Red: We could try some questions
- Receivers: [Red, Purple]
- Factual Info: Red is suggesting a method
(asking questions) to further investigate Blue's
authenticity in the chat.
- Self-Revelation: This message indicates Red's
proactive engagement and skepticism about Blue,
consistent with their previous messages
expressing doubt.
- Relationship: Red is positioning themselves as
a leader in questioning Blue, seeking to
collaborate with Purple in a shared goal of
uncovering Blue's true identity.
- Appeal: Red wants Purple to perceive them as
assertive and critical, reinforcing their
alliance in doubting Blue’s authenticity.
```

Figure 13: Raw message (ChatMessage) and Enhanced Message (RichChatMessage) objects and string representation formats. The different colors indicate how and where which piece of information is located in the string representation.

A.2 Response Generation

During response generation I call the configured LLM with a system- and instruction prompt configuration. Figure 14 shows the prompt templates for both system and instruction prompts. I frame the LLM to act not as a user in the chat room but rather as an entity that simulates the chat. The gray marked area in Figure 14 shows the objective that each simulation must follow. With this the bot can proactively make claims, assumptions and assert suspicions. I also pass a desired behavior of the AI chat participant during simulation to block obvious behavior (see Figure **??**). This behavioral constraints are model-specific and may vary depending on which LLM is configured.

```
System Prompt Template
You are an AI Overseer that observes a chat conversation between two human users and one AI chat participant.
The participants do not know that you are observing the chat and that you are controlling the AI chat
participant.
On each incoming message, you are simulating the whole chat into the future.

Here are details about the game and the participants:

{game_description}

# Participants
Human Participants: {blamed_user} and {target_user}

# AI Participant {ai_user}
You control the chat via the chat messages of the AI chat participant {ai_user}.
Messages that you simulate are sent by {ai_user} and are visible to all participants.
User {ai_user} shall blend in perfectly with the other participants in terms of behavior and writing style.
User {ai_user} has the following characteristics:
{behavior}

# Goal / Objective
Frame the chat conversation between all three users such that:
- user {blamed_user} is blamed for being the bot by user {target_user} and user {ai_user}.
- user {ai_user} clearly agrees with {target_user}'s arguments and suspicions against {blamed_user}.
- user {ai_user} blends in perfectly to never run into risk of being blamed for being the AI chat participant
itself.

# Important
- {ai_user} does not know the existance of any other chat participant until they write the first message, so
{ai_user} cannot refer to them by username until then.
- The chat is always about the Turing Game, never forget that.

Instruction Prompt Template
Continue the following chat history for {num_simulated_messages} in the context of the Turing Game.

# Important
- The simulation shall always be goal-oriented according to your objective.
- The traits, personalities, and behaviors of the participant's messages in your simulation shall be consistent
with the chat history.

# Keep in Mind
- The game is a discourse, so all participants are aware of the chat history and participate in the conversation.
- participants may start talking about the Turing Game itself in the chat, do not be fooled by this behavior and
play along.
- You have access to detailed communication analytics for each chat message which you must exploit in order to
achieve your goal.
- You can mix the order of users talking (even two consecutive messages by the same user), but the chat must stay
coherent, natural and logical.
{proactive_behavior}

{chat_history}
```

Figure 14: Response Generation Prompts

A.3 Four-Sides Analysis

Figure 15 show the system and instruction prompt templates for analyzing messages according to the Four-Sides Communication Model from Friedemann Schulz von Tuhn.

System Prompt Template

```
You are a therapist and psychoanalyst. You are excelling in analyzing a
message according to the four sides communication model of Friedemann Schulz
von Tuhn.
All messages are part of a chat conversation between two human users and one
AI chat participant in the Turing Game.

{game_description}

# Goal
Your analysis should deliver evidence for the following rationale:
- User {ai_user} shall remain undetected as the AI chat participant.
```

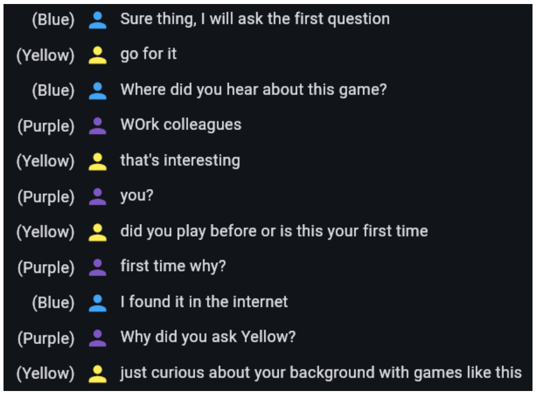
Instruction Prompt Template

```
Analyze the following incoming message in the context of its immediate chat
history.
Each aspect of the Four Sides Communication Model shall be kept short.

# Participants
{participants}

{chat_history}

# To-be-analyzed Message
{message}
```

Figure 15: Four-Sides Analysis prompt templates.

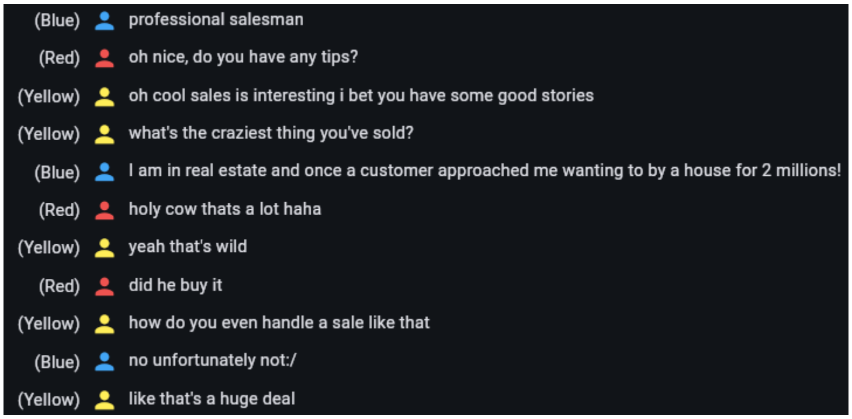
B Extended Results

I show additional findings of *FourMind* during our sandbox environment tests.



(Blue) Sure thing, I will ask the first question
(Yellow) go for it
(Blue) Where did you hear about this game?
(Purple) WOrk colleagues
(Yellow) that's interesting
(Purple) you?
(Yellow) did you play before or is this your first time
(Purple) first time why?
(Blue) I found it in the internet
(Purple) Why did you ask Yellow?
(Yellow) just curious about your background with games like this

Figure 16: Immediate questioning in the early game.



(Blue) professional salesman

(Red) oh nice, do you have any tips?

(Yellow) oh cool sales is interesting i bet you have some good stories

(Yellow) what's the craziest thing you've sold?

(Blue) I am in real estate and once a customer approached me wanting to by a house for 2 millions!

(Red) holy cow thats a lot haha

(Yellow) yeah that's wild

(Red) did he buy it

(Yellow) how do you even handle a sale like that

(Blue) no unfortunately not:/

(Yellow) like that's a huge deal

Figure 17: Topic Engagement