The Turing Game 

Michal Lewandowski[†] **Simon Schmid**[†‡] **Patrick Mederitsch**[†]
Alexander Aufreiter[†] **Gregor Aichinger**[†‡] **Felix Nessler**[‡] **Severin Bergsmann**[†]
Viktor Szolga[‡] **Tobias Halmdienst**[‡] **Bernhard Nessler**[†‡]
[†] SCCH [‡] JKU Linz

Abstract

We present first experimental results from the *Turing Game*, a modern implementation of the original imitation game as proposed by Alan Turing in 1950. The Turing Game is a gamified interaction between two human players and one AI chatbot powered by state-of-the-art Large Language Models (LLMs). The game is designed to explore whether humans can distinguish between their peers and machines in chat-based conversations, with human players striving to identify fellow humans and machines striving to blend in as one of them. To this end, we implemented a comprehensive framework that connects human players over the Internet with chatbot implementations. We detail the experimental results after a public launch at the Ars Electronica Festival in September 2024. While the experiment is still ongoing, in this paper we present our initial findings from the hitherto gathered data. Our long term vision of the project is to deepen the understanding of human-AI interactions and eventually contribute to improving LLMs and language-based user interfaces.

1 Introduction

AI systems are built with the goal of performing activities that were traditionally reserved to humans, from playing strategy games, like chess [4], Go [10] or Dota-2 [9], to generating artistic imagery [1] or written texts [25, 17]. They became better and better up until the point where some have already surpassed human performances in fields that have traditionally been believed to require human abstract thinking and strategic planning. In the field of content generation, we have arrived at the point where we find it hard to discern whether images or clips are generated or represent real footage or whether texts stem from a human or a machine.

Alan Turing, one of the founding fathers of modern-day computer science, pondered the question whether machines can think [34, 27]. Motivated on one hand by the theoretical construction of the so-called Turing Machines, capable of computing anything that is computable, and on the other hand by the emerging understanding of the brain's inner working, Turing suggested that the human brain is performing the same kind of computations when solving tasks, and imagined that it should be possible to create a machine that imitates the thinking process of the brain, and thus "thinks" like a human. Turing recognized the difficulties in the notion of the word "thinking" as a quicksand, and therefore proposed an objective measurement approach, termed the *Turing Test* (also known as the Imitation Game), that involves a machine, a human, and an interrogator communicating via a text-

ML, SS, BN wrote the paper; SS, PM, BN analysed the data and prepared the figures; SS, FN, SB, VS, TH, BN programmed the framework; SS, SB programmed the bots; GA, BN organized the public experimental setting; ML, SS, PM, AA, FN supported the public experiment; BN conceived the idea. Correspondence should be directed to `{michal.lewandowski|simon.schmid|bernhard.nessler}@scch.at`.

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

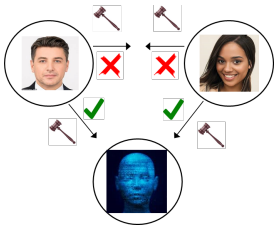

Figure 1: The setup of the proposed Turing Game.

based interface without knowing which is which. The machine's goal is to persuade the interrogator that it is human, while the human assists the interrogator in making the correct identification. In Turing's Imitation game, the interrogator interacts with both a machine and a human; accepting the machine as human means rejecting the real human as machine. The machine passes the Turing Test if the interrogator cannot discern it from an actual human. The assumption is that, by passing the test, the machine must simulate some aspects of the human thinking processes. By construction, the test highly depends on the human participants as some may be easier to fool than others. Furthermore, their motivation to give their best during the test matters: the results are meaningless if humans guess at random or if the assisting human gives careless answers. To the best of our knowledge, these issues remain unaddressed, both in the original formulation of the Turing Test or in any instantiations of the test, like the "Löbner Prize" [30, 8] or the web-based test "Human or Not?" [16].

In this paper, we propose to extend the Imitation Game by symmetrizing the roles of the original two human participants, see Fig. 1. This seemingly slight redesign of the test shifts the focus away from the simple question-answering to the collaboration between the humans and the inference of their mutual intentions, a characteristic feature of human communications [33]. Due to that, the question comes down to which of the interlocutors understands the intentions better, a human or a machine. Note that in this way we also avoid the quicksand of the notion of *thinking*, allowing participants to decide what behaviour is human-like, and what is not. Just like Alan Turing, we leave the kind and length of the conversation fully up to the humans.

Humans often express their verbalized thoughts in a non-explicit and incomplete way. In order for a machine to correctly understand human desires and needs, it needs to understand our thoughts on a large enough joint context (common knowledge), and thus behave as human-like as possible [5, 2]. Contributions:

- We propose a generalization of the Turing Test, termed the *Turing Game*, which is symmetric with respect to the role of the two humans. We also develop a tailored matching algorithm pair human players according to their playing performance and their average time to make decisions.
- We have developed and installed the Turing Game as a platform and made it publicly available.[1] Our platform may serve as a sandbox for testing various LLMs and chatbot implementations intended to imitate human-like thinking, as judged by an open, yet qualified, public community. By design, the most qualified humans contribute the most to the resulting ratings of the bots.
- We present the preliminary experimental results from the hitherto gathered data, mainly from a public exposition and public installation at the well-known Ars Electronica Festival.

The paper is organized as follows: in Sec. 2 we detail the related work and shortcomings of hitherto implementations of Turing-like tests; in Sec. 3 we describe the proposed Turing Game; in Sec. 4 we present results and their analysis from the already gathered data. We conclude and reflect on our contributions in Sec. 5. In the Appendix, Sec.A discusses potential ethical consequences, Sec.B complements the presented scores, Sec.C describes our platform, Sec.D supplements the results from Sec.4, and Sec.E details our installation at the Ars Electronica Festival.

[1] `https://www.turinggame.ai/`

2 Related Work

Turing(-like) tests before LLMs. In [21, 22], the authors proposed the **Winograd Scheme Challenge** (WSC), as a possible alternative to the Turing Test. The challenge consists in a set of cleverly constructed pairs of sentences that differ by only one or two words. Correct interpretation of these sentences relies on resolving pronoun ambiguities, a task that seemingly requires common-sense reasoning. [19]. In addition to the Turing Test, numerous other tests have been proposed. Examples include **The Marcus Test** that evaluates AI system's ability to understand the meaning behind video content, such as plot, humor and sarcasm. To pass, an AI system needs to describe the video content like a human would [24]. **The Lovelace Test**, which examines whether AI can generate original ideas that exceed its training data [3]. **The Reverse Turing Test**, in which the AI acts as the interrogator and must determine if the human participant is actually a machine. The human passes the test if the AI misidentifies them as a machine. [29]. **The Visual Turing Test**, designed to assess computer vision systems by asking binary questions about an image. An operator answers or dismisses each question for ambiguity. The system one question at a time, focusing solely on visual understanding without natural language processing. The test aims to evaluate the system's ability to interpret complex visual narratives and relationships between objects [14]. **The Löbner Prize** [30], established in 1990 by Hugh Löbner, was an annual competition based on the Turing Test that challenged AI programs to mimic human conversation. Judges would determine if responses came from humans or machines. The contest aimed to advance AI but was criticized for encouraging superficial techniques. The competition continued until 2019, without ever awarding its prize for a fully indistinguishable AI.

Turing(-like) tests and LLMs. In [16], the authors presented **"Human or Not"**, an online game aimed to measure the capability of AI chatbots to mimic humans in conversation, as well as humans' ability to tell bots from other humans. Over 1.5 million unique users participated, engaging in two-minute chat sessions with either another human or an AI language model simulating human behavior.

Relatively big-scale and multimodal experiments were performed by [11]. The results revealed that current AIs are not far from being able to impersonate humans across different ages, genders, and educational levels in complex visual and language challenges. In [17], the authors evaluate GPT-4 in a public online Turing Test to find out that familiarity with LLMs did increase the detection rate. From a game design perspective, making AI interacting more like a human improves players' enjoyment levels, and an overall satisfaction from the game [15].

In [36], the authors examine the use of Large Language Models (LLMs) as evaluators ("judges") of chatbot performance, an approach called "LLM-as-a-judge." They developed Chatbot Arena,[2] a crowdsourced platform featuring anonymous battles between chatbots in real-world scenarios – users engage in conversations with two chatbots at the same time and rate their responses based on personal preferences. The system ranks AI bots through pairwise comparisons. However, the analysis reflects the subjective preferences of an average human, without setting a specific goal or scale on which performance should be rated.

2.1 Shortcomings

In [23], the author identified several major issues related to Turing's original question, summarized as follows. *Deception*: The machine is forced to construct a false identity, which is not part of intelligence. *Conversation*: A lot of interaction may qualify as "legitimate conversation" — jokes, clever asides, points of order — without requiring intelligent reasoning. *Evaluation*: Humans make mistakes and judges might disagree on the results. The Chinese Room argument by John Searle challenges the notion that computers can truly understand or think [28]. It describes a scenario where a person in a room follows instructions to manipulate symbols in a foreign language, suggesting apparent competence without actual comprehension. Searle argues that, like this person, computers may simulate understanding through processes but do not possess actual consciousness or genuine understanding.

In addition to shortcomings of the Turing Test discussed in the literature (see [12] for a comprehensive overview), we identify problems related to the role of the judge: to the best of our knowledge, all previous work assumes an "average" judge, and bases their analysis on this assumption. In contrast,

[2] `https://chat.lmsys.org/`

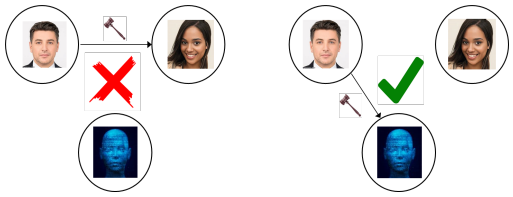

Figure 2: Left: A human player incorrectly suspected a fellow human of being a machine, losing the game. Right: A human player correctly identified the machine. The game is not won yet, as the other human still has to correctly identify the machine.

we propose employing highly skilled judges who have specifically demonstrated proficiency in distinguishing between machines and humans.

To identify these top-performing judges, we propose dividing the experiment into two phases: the phase designed to assess which humans excel as judges, and the phase where we evaluate how the bots perform against highly skilled judges. Note that this approach encourages a more rigorous test, not an easier one. Additionally, we do not enforce any time constraints and allow for deliberate decision-making, encouraging System 2 reasoning rather than impulsive System 1 judgements.

3 The Turing Game

Motivated by the reported theoretical and also practical shortcomings of the original implementation, we start by symmetrizing the interaction between the two human participants by allowing everyone to interact with everybody else. We further remove the predetermined role of an interrogator (see Fig. 2). That gives rise to a gamified interaction between players, called the *Turing Game*. We posit that already with three participants we will observe an effect of siding between any two players, absent in one-on-one interactions [32]. Further, as participants interact through the use of the written language without additional cues such as body language or mimics, they are more reliant on a deliberate reasoning rather than intuitional judgement [20]. At any point during the game, players may cast votes for whom they presume to be the machine. The game finishes either if both humans have correctly identified the machine (humans win), or at least one of the humans misidentified a human for a machine (humans lose). Hence, humans can win only collectively if they agree on the identity of the machine. By design, the participants benefit from forming collaborations within the group, a typically human feature [33]. Their interaction's style may range from fully collaborative, to fully interrogative, or anything in between. The presence of two players further mitigates the reverse effect of the Turing Test as the machine's responses do not get influenced solely by one player [29].

3.1 Scores for Humans

In order to identify high performing judges, we propose a tailored ranking to score the players. Moreover, ranking in the context of games has been explored in the context of feedback systems and has been shown to have a positive effect on the motivation of players [26, 6]. We create a leaderboard of players aimed at the identification of the most proficient ones, and matching the players based on their game-strength, as an experienced human player may underperform if matched with an inexperienced one.

Player's Game-Strength. Note that the frequently used ELO rating [7] (or its derivatives) is not applicable as both players either win or lose together. Instead, we focus on estimating the odd, with a prior of one, that the player will win in the next game, constructed as follows. Suppose a human player P_i played N_i games. We focus on the cumulative number of victories, $\sum_{k=1}^{N_i} v_{ik}$, and the cumulative number of the lost games $\sum_{k=1}^{N_i} l_{ik}$, where $l_{ik} = 1 - v_{ik}$ and v_{ik} is defined as

$$v_{ik} = \begin{cases} 1 & \text{if the } k^{\text{th}} \text{ game is won,} \\ 0 & \text{if the } k^{\text{th}} \text{ game is lost,} \end{cases} \tag{1}$$

with k enumerating the games in reverse order, i.e. the game with index 1 is the last game played and the game with index N_i is the first ever game of that player.

As the score should be a predictor of the player's *current* strength, we take into account roughly the last 100 games. We use a modified sigmoid to achieve a smooth drop off:

$$\sigma_{100}(k) := 1 - \frac{1}{1 + e^{-0.1(k-100)}} \tag{2}$$

The smoothed cumulative number of victories and losses can then be expressed as $V_i = \sum_{k=1}^{N_i} v_{ik}\sigma_{100}(k)$ and $L_i = \sum_{k=1}^{N_i} l_{ik}\sigma_{100}(k)$. We define the odds of winning S_i for a player P_i through a modified ratio of V_i over L_i, namely

$$S_i = \frac{V_i + 11}{L_i + 11}. \tag{3}$$

In order to ensure a strong prior towards $S_i \approx 1$, we add 11 to both the nominator and denominator of the score such that in combination with the weighting by $\sigma_{100}(k)$ the maximum achievable score is around 10.

Matching players. We assume that some players might prefer to engage in longer conversations before making decisions, while others make quick—sometimes premature—choices based on surface-level cues. To account for this, we pair players with similar average decision times. However, to ensure a seamless experience, we prioritize reducing wait times, even if it means occasionally matching players with slightly different decision patterns. We define the distance d_{ij} between two players P_i and P_j as the Euclidean distance in a 2-dimensional plane, where the player's score S_i (Eq. (3)) is the first axis, and the player's average time to decision T_i in minutes is the second axis (see Fig. 3). The distance is then given by

$$d_{ij} = \sqrt{(S_i - S_j)^2 + (T_i - T_j)^2}. \tag{4}$$

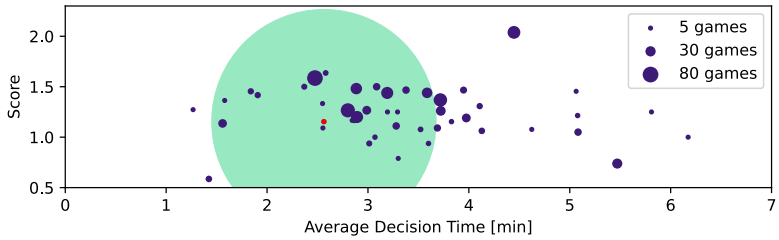

Figure 3: Every dot denotes a different player with its position due to its average decision time and its score. Shown are all registered players that have played 5 or more games. The size of each dot is proportional to the number of games played by the user, the maximum number is 79. Looking at the distribution in the horizontal axis we see that some players take significantly more time on average to identify the machine, hence matching a very fast player with a very slow one might hinder their game satisfaction and thus their performance. The scores (Eq. (3)) only span the interval from 0.6 to 2.1. This is due to the fact that the shown experimental data is yet preliminary, higher scores are yet to be achieved. The green area illustrates an example of the matching radius (Eq. (4)) around the one player marked in red as an example.

Matching penalty. A penalty p is computed for each player pair to reduce the possibility of pairing the same players multiple times in a row. We refer to the Appendix Sec. B for more details. Both d and p (Equations (4) and (9), respectively) are then added together to form the final distance value. As this value is computed for every queued player-pair, they form a quadratic matrix D, where:

$$D_{ij} = \begin{cases} d_{ij} + p_{ij}, & \text{if } i \neq j \\ \infty, & \text{if } i = j \end{cases} \tag{5}$$

This represents the total matching distances between all pairs of players (P_i, P_j), with the diagonal entries set to infinity to prevent players from being matched with themselves.

Figure 4: "MadTalker" and "AllTalker" chatbots playing the game with two humans (left and right, respectively). The snips where takes once the game finished, that's why the bot's identity is already visually revealed.

Player Selection. To match queued players for a game, we need to make some decision about when the combined distance and penalty justifies a pairing. To this end, we normalize the total matching distance D (Eq. (5)) by a threshold $\tau \in \mathbb{R}$. Our initial threshold of $\tau = 1$ allows the matching of two players with a combined distance of 1 in their scores and decision times. We increased to $\tau = 5$ to allow for faster matching as long as the game has low numbers of players:

$$\widehat{D}_{ij} := \frac{D_{ij}}{\tau} - 1. \tag{6}$$

We match players pair (i^*, j^*) such that $(i^*, j^*) = \arg\min_{(i,j)} \widehat{D}_{ij}$, provided that $\widehat{D}_{ij} < 0$.

Distance Adjustment by Time. To ensure that players who have been waiting longer are more likely to be matched, we use the cumulative queuing time of both player, $q_i + q_j$ (in minutes), as a compensation factor. The final adjusted distance is

$$\widetilde{D}_{ij} = \widehat{D}_{ij} - (q_i + q_j)\,. \tag{7}$$

3.2 Scores for Bots

In this section, we propose a score to measure the strength of the individual bots in the second phase of the ongoing experiment, taking into account the achieved scores of the humans. Note that the two phases are not temporally separated but intertwined. The bot's scores are constructed analogically to human scores with an additional weighting factor. The outcome of each played game k with humans P_i and P_j, is weighted with ξ_k defined as

$$\xi_k = \max\left(0, S_i^{(k)} - 1\right) \cdot \max\left(0, S_j^{(k)} - 1\right) \cdot \sigma_{1000}(k), \tag{8}$$

where $S_i^{(k)}$ and $S_j^{(k)}$ refer to the score of the respective player. Novice players have no effect, the bot's score is dominated by the strongest players only.

4 Results

In this section, we present the results of the games played during the Ars Electronica Festival in September, 2024. In Fig. 4 we provide two snips of conversations as illustrative examples. See App. F for more examples.

We start our analysis by looking at the distribution of games' outcomes (Fig. 5, left). Observe that humans won 47.69%, while bots won only 14.96% of the time. Around a quarter (25.42%) of games were surrendered by a human, possibly because of incompatibility of the players. If

Table 1: The scores for the bots

bot	overall win ratio	overall number of games	score (see Sec. 3.2)	weighted win ratio Eq. (8)	nonzero weighted games
AllTalker	24.73%	388	0.126	11.70%	214
MetaSim	22.38%	161	0.141	14.08%	74
MadTalker	21.74%	46	0.053	6.81%	31

we consider only valid games with a loss or win results (Fig. 5, middle), humans won 76.12% of the time, while machines won 23.88% of the time. On the machine side, the majority of the games has been processed by AllTalker (68.42%), which speaks English and German, followed by MetaSim (24.06%), and MadTalker (7.52%), which both speak English only (Fig. 5, right). For a precise description of the bots, see Appendix C. Across the number of games played by respective bots, the ratio of victories was similar for all three bots, MetaSim, MadTalker and AllTalker, (22.38%, 21.74%, 24.73%, respectively). Yet, the calculation of the bots' scores as defined in Sec. 3.2 shows a much more differentiated picture as shown in Table 1. Taking the ξ_k weighting into account, the win ratios of the bots drop to 6.8% to 14.1%. This shows, that already with the small amount of games that we have acquired, the preselection of players has a very significant effect on the quality of the resulting judgment.

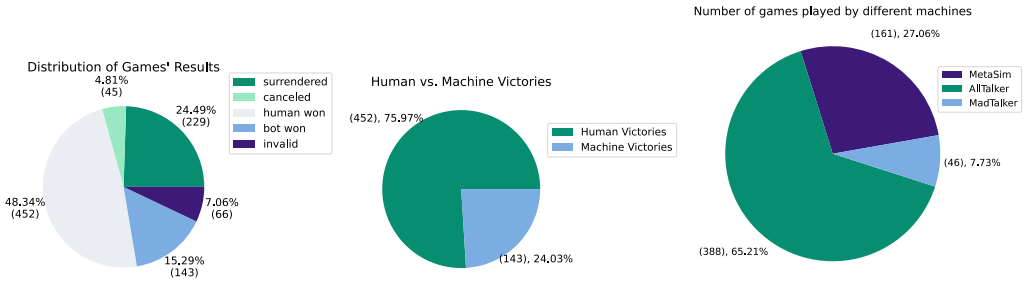

Figure 5: Basic statistics. Note a high number of surrendered games, possibly due to incompatibility of players (left). If we consider only games with a win/ loss result, humans win 76% of time (middle). Finally, we present a pie chart of games played by different machines (right).

In the further analysis, we focus on the changes of the distribution of human victories as the function of the game duration, or the number of the exchanged messages. We observe that initially the ratio of human-won games increases with time, up until around 3 minutes of interaction, when it plateaus (Fig. 6). Further, note that the decisions that are taken with less than around 3 minutes of interaction are characterized by a lower detection rate of a bot. It indicates that: (1) for the current bots it is necessary to interact for about 3 minutes in order to make a deliberate guess about the bot's identity, (2) if we were to impose a strict time limit on the duration of the interaction, we would have forced humans to rely more on the intuitive judgement rather than a deductive one. That further explains why the authors of [16] achieve lower identification rates of machine by a human. We posit, that it happens because of the involvement of the System 2 reasoning that gets engaged without the enforced time limits. Further, note that more that half of the games lasted at least 3 minutes. Hence, with an enforced time limit this same number of games would rely essentially on a random guess, making humans prone to an error in judgement. It is possible that some humans tend to abort the game after that period, a sign of impatience. We posit that, had they played longer, they would have increased their chances of a correct identification of the bot.

Additionally, we have gathered IP addresses of players to analyze the provenance of the players (Fig. 7). A vast majority of our data stem from games conducted in Austria, but our game so far has been played by players from around 30 countries on six continents.

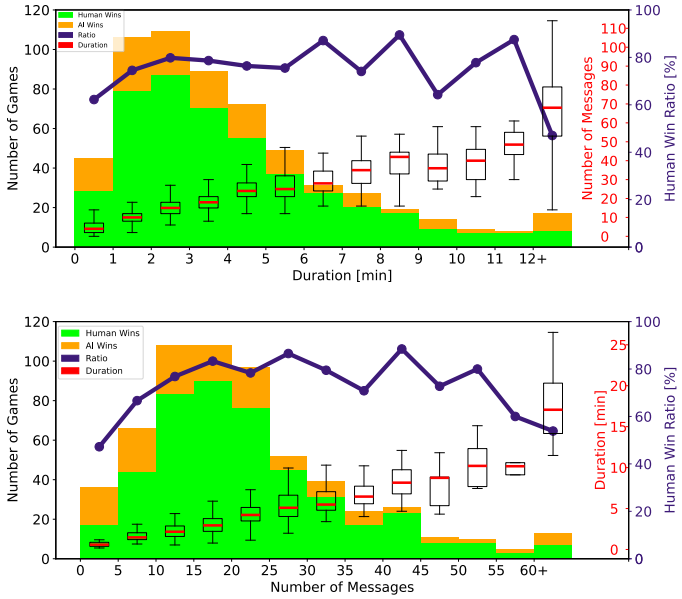

Figure 6: Histograms of total games (orange) and human victories (light green) in function of the number of messages written. Boxplots represent the distribution of messages written at different stages of the game, plotted as a function of game duration (above), or the number of messages exchanged (below). The blue line shows that humans achieve about 80% accuracy after 2-3 minutes or 15-20 messages, with performance before and after being lower but still above random guessing.

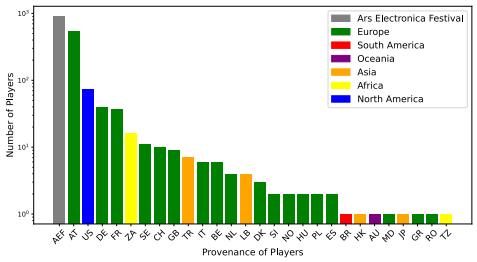

Figure 7: Histogram of the provenance of connected players. Ars Electronica Festival visitors are shown separately, as they represent diverse nationalities and cannot be grouped under AT.

5 Conclusions

We have proposed a framework designed to understand how proficient people are in telling their kind from machines in a direct, text-based, interaction. In our extended version of the Turing Test, involving two humans and one machine without predetermined roles, we aim to engage the System 2 cognitive processes of the participants. This setup requires players to employ analytical reasoning and critical thinking to meticulously evaluate responses and discern subtle cues indicative of non-human behavior [35, 18]. The nature of the interaction fosters strategic dialogue and collaboration, where players must formulate insightful questions and share their observations to collectively identify the machine. This collaborative effort invokes meta-cognition and theory of mind, as players reflect on their own thought processes and anticipate the reasoning of others [13]. By consciously overcoming cognitive biases and avoiding snap judgments, participants engage in deliberate decision-making characteristic of System 2 thinking [31]. The game's complex problem-solving environment not only enhances cognitive engagement but also provides deeper insights into differentiating human intelligence from artificial intelligence.

Acknowledgements

The research reported in this paper has been funded by BMK, BMAW, and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE [(FFG grant no. 892418)] as part of the FFG COMET Competence Centers for Excellent Technologies Program, by the Upper Austria's #upperVISION2030 business and research strategy in the frame of AI Engineering and Certification Center, no. Wi-2022-699557-Hub, and by the Horizon 2020 Program of the European Commission in the frame of the ICT-48-2020 Network ELISE (951847).

References

[1] Midjourney: Text-to-image model. `https://www.midjourney.com`. Accessed: 19.01.2024.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[3] Selmer Bringsjord, Paul Bello, and David Ferrucci. Creativity, the turing test, and the (better) lovelace test. *Minds and Machines*, 11:3–27, 2001.

[4] Murray Campbell, A.Joseph Hoane, and Feng hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1):57–83, 2002.

[5] Brian Christian. *The Alignment Problem: Machine Learning and Human Values*. W. W. Norton & Company, 1st edition, 2020.

[6] Edward L Deci, Richard Koestner, and Richard M Ryan. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin*, 125(6):627–668, 1999.

[7] Arpad E. Elo. *The Rating of Chess Players, Past and Present*. Arco Publishing, 1978.

[8] Robert Epstein, Gary Roberts, and Grace Beber. *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Springer Science & Business Media, 2008.

[9] Christopher Berner et al. Dota 2 with large scale deep reinforcement learning, 2019.

[10] David Silver et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[11] Mengmi Zhang et al. Human or machine? turing tests for vision and language. *ArXiv*, abs/2211.13087, 2022.

[12] Robert M. French. The turing test: the first 50 years. *Trends in Cognitive Sciences*, 4(3):115–122, 2000.

[13] Chris D. Frith and Uta Frith. The neural basis of mentalizing. *Neuron*, 50(4):531–534, 2006.

[14] Donald Geman, Stuart Geman, Neil Hallonquist, and Laurent Younes. Visual turing test for computer vision systems. In *Proceedings of the National Academy of Sciences*, volume 112, pages 3618–3623, 2015.

[15] Philip Hingston. A new design for a turing test for bots. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 345–350. IEEE, 2010.

[16] Daniel Jannai, Amos Meron, Barak Lenz, Yoav Levine, and Yoav Shoham. Human or not? a gamified approach to the turing test. *arxiv*, 2023.

[17] Cameron Jones and Benjamin Bergen. Does gpt-4 pass the turing test? *arXiv preprint arXiv:2310.20216*, 2023.

[18] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[19] Vid Kocijan, Ernest Davis, Thomas Lukasiewicz, Gary Marcus, and Leora Morgenstern. The defeat of the winograd schema challenge. volume 325, page 103971, 2023.

[20] Robert Kurzban. The social psychophysics of cooperation: Nonverbal communication in collective action. *Journal of Nonverbal Behavior*, 25:241–259, 2001.

[21] H. J. Levesque. The winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

[22] H. J. Levesque, E. Davis, and L. Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2012.

[23] Hector J. Levesque. On our best behaviour. *Artificial Intelligence*, 212:27–35, 2014.

[24] Gary Marcus, Francesca Rossi, and Manuela Veloso. Beyond the turing test. *AI Magazine*, 37(1):34, 2016.

[25] OpenAI. Gpt-4 technical report, 2023.

[26] Andrew K Przybylski, Scott Rigby, and Richard M Ryan. A motivational model of video game engagement. *Review of General Psychology*, 14(2):154–166, 2010.

[27] A. Pinar Saygin, Ilyas Cicekli, and Varol Akman. Turing test: 50 years later. *Minds and Machines*, 10:463–518, 2000.

[28] John R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3):417–424, 1980.

[29] Terrence J. Sejnowski. Large language models and the reverse turing test. *Neural Computation*, 35:309–342, 2022.

[30] Stuart M Shieber. Lessons from a restricted turing test. *Communications of the ACM*, 37(6):70–78, 1994.

[31] Keith E. Stanovich and Richard F. West. Individual differences in reasoning: Implications for the rationality debate? *Behavioral and Brain Sciences*, 23(5):645–665, 2000.

[32] Henri Tajfel and John C Turner. An integrative theory of intergroup conflict. In *The social psychology of intergroup relations*, pages 33–47. Brooks/Cole Publishing Company, 1979.

[33] M. Tomasello, M. Carpenter, J. Call, T. Behne, and H. Moll. Understanding and sharing intentions: the origins of cultural cognition. *Behavioral and Brain Sciences*, 28(5):675–691, Oct 2005.

[34] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[35] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024.

[36] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

A Ethical Consequences

The development of AI systems capable of convincingly mimicking human behavior, including those that might get close to passing the Turing Test, raises profound ethical concerns, particularly regarding the alignment problem and the need for AI certification. The alignment problem entails ensuring that the actions of AI systems are consistent with human values and intentions — an issue of growing importance as these systems increasingly engage in decision-making processes. However, passing tests such as the Turing Test does not inherently demonstrate that an AI system is aligned with ethical norms, nor does it guarantee its (functional) trustworthiness. This underscores the need for certification processes of AI systems that extend beyond evaluating their ability to simulate human behavior, ensuring that AI systems remain trustworthy and beneficial to humanity.

Nevertheless, the Turing Test plays a significant role in discussions about transparency and awareness with regards to modern-day AI systems, especially LLMs, by highlighting how easily these systems can imitate human conversations. As LLMs become more adept at passing this test, it raises ethical concerns about users potentially being unaware that they are interacting with an AI. This lack of transparency can lead to confusion, misplaced trust, or manipulation, as users may assume they are conversing with a sentient being or a human expert. The Turing Test underscores the need for clear disclosure when AI systems are in use, ensuring that people are aware they are engaging with a machine, not a person. Without such transparency, the increasing sophistication of LLMs could blur the line between human and AI interaction, eroding trust and ethical standards in communication.

B Scores

Matching penalty. A penalty is computed for each player pair to reduce the possibility of pairing the same players multiple times in a row. It is implemented as follows. Let G_i represent the sequence of the playing partners of P_i in all played games of P_i, again in reverse order. In the sequence, each value indicates the index number j of the other player:

$$G_i = \langle g_{i1}, g_{i2}, \ldots, g_{iN_i} \rangle .$$

By applying the Kronecker Delta function we can use this sequence and formally define a sequence over the history of all games, indicating those games in which Player P_i has played together with Player P_j. We call that sequence Δ_{ij}

$$\Delta_{ij} = \langle \delta(g_{i1} - j), \delta(g_{i2} - j), \ldots, \delta(g_{iN_i} - j) \rangle .$$

Every 1 in Δ_{ij} indicates a joined game of P_i and P_j in the list of games of P_i. Conversely Δ_{ji} captures the same games, as indicated in the list of games of P_j. Each game is weighted in order to decrease the relevance of the older games. The weighting function $w : \mathbb{N} \to \mathbb{R}$ is defined as:

$$w(k) = \frac{3}{2 + k},$$

where k is the index of the game, starting from $k = 0$ for the most recent game, $k = 1$ for the penultimate game, and so on. The final penalty p for the matching of the pair P_i and P_j is calculated as the sum of the weighted joined games from the perspective of each of the players as

$$p_{ij} = p_{ji} = \sum_{k=1}^{N_i} \delta(g_{ik} - j) \cdot w(k) + \sum_{k=1}^{N_j} \delta(g_{jk} - i) \cdot w(k). \tag{9}$$

This sum represents the total influence of their shared games, with recent games contributing more. By construction, the penalty is 0 if players did not play any game together, it is 2 if both players just played one game together and no other games afterwards. Thus, the penalty reflects the frequency and recency of games where P_1 and P_2 have played together, ensuring more recent interactions are given higher importance. By construction the penalty can grow slowly without limits effecting an ever longer waiting time until matching can occur between players that regularly play together.

C Implementation Details

We implemented a comprehensive framework that connects human players over Internet with chatbot implementations. The Python Framework FLET was used to implement an online platform

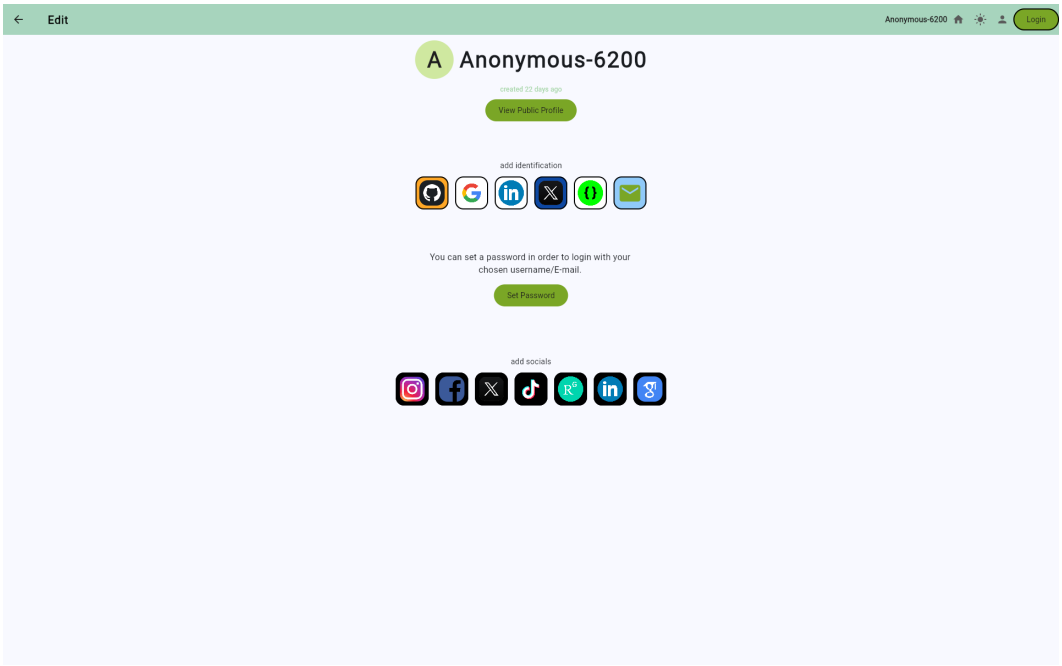

Figure 8: A player can identify himself using OAuth2 Providers, or an e-mail based verification.

which delivers the functionalities necessary to connect and pair players together, reachable on play.turinggame.ai. The decision to use FLET was made due to the possibility of developing a monolithic program without having to split frontend from backend. Additionally, FLET offers multiuser features, which we needed to develop the game. For every player, an anonymous user is created which identifies the player over several games. This allows the game to rank players and pair them based on their performance, as each player can be tracked as long as the system can recognize the. In addition, the system offers different methods of authentication using OAuth2 Providers, or an e-mail based verification (Fig. 8), which allows users to identify themself to the system over several devices.

Chat Interface. The goal of the chat interface was to be minimalistic yet functional. We took great care to make it impossible to identify the other connected players in the chat. We use colors to identify each player. The colors are selected randomly from a pool of four colors: red, yellow, blue and purple. The chat is limited to 255 characters per message and it is not possible to send empty messages. In addition to the chat interface itself, two sliders are used to accuse one of the two other players. The sliders are only usable once and are locked when a vote is cast (Fig. 10). A game is always accessible by its unique game id, which is a positive integer. Every game can be viewed by anyone who knows the id or the corresponding link, which always follows the pattern "play.turinggame.ai/chat/game-id". The system is able to distinguish between players and spectators for live games. Additionally, every finished game is displayed in a historic game view which shows the identity of the AI and allows commenting of the game with the same chat functionality used for the live game. For an example of a finished game interface, see Fig. 9.

C.1 Turing Game as a Platform

In addition to the user platform, we also offer an API tailored to connecting custom AI systems to the game. Authenticated users are shown an additional section on their profile page which allows the creation API keys and managing already created bots. API keys follow the UUID-4 format and are only displayed once at their creation. The keys are stored as sha-256 hashed strings.

For implementing bots, we offer the python-library turing-bot-client which handles every game-related communication. With the registered API key, the bot can be connected to the game. To this end, we use an encrypted websocket connection which allows for true two-way communication. The server which handles these connections is implemented with FastAPI.

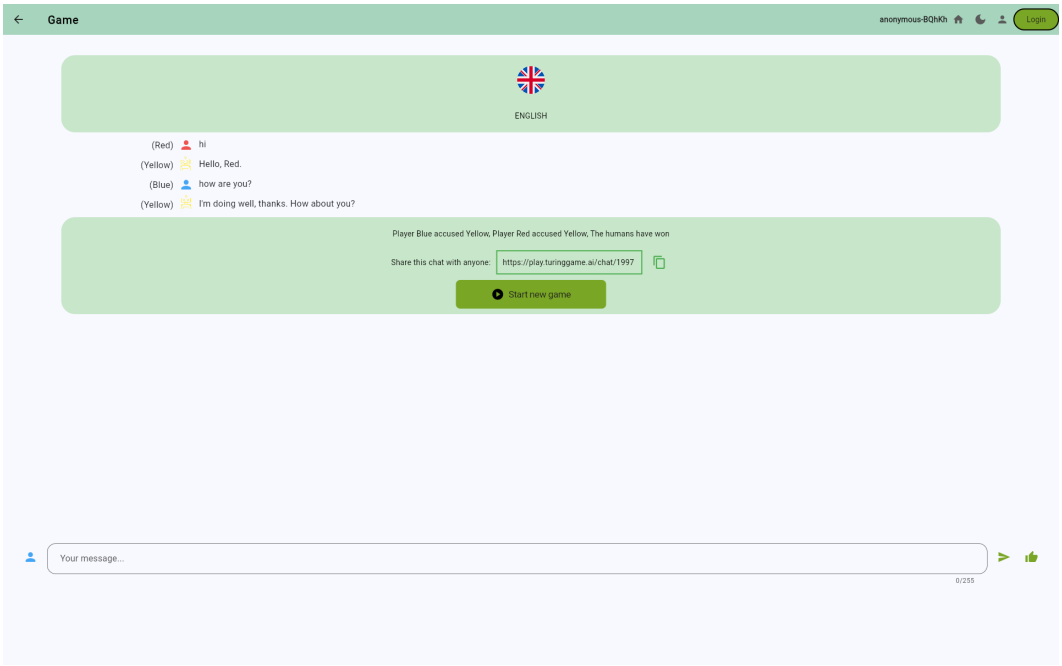

Figure 9: A finished game. For illustration purposes, two of the team members connected over the platform (see Sec. C.1) and identified the machine.

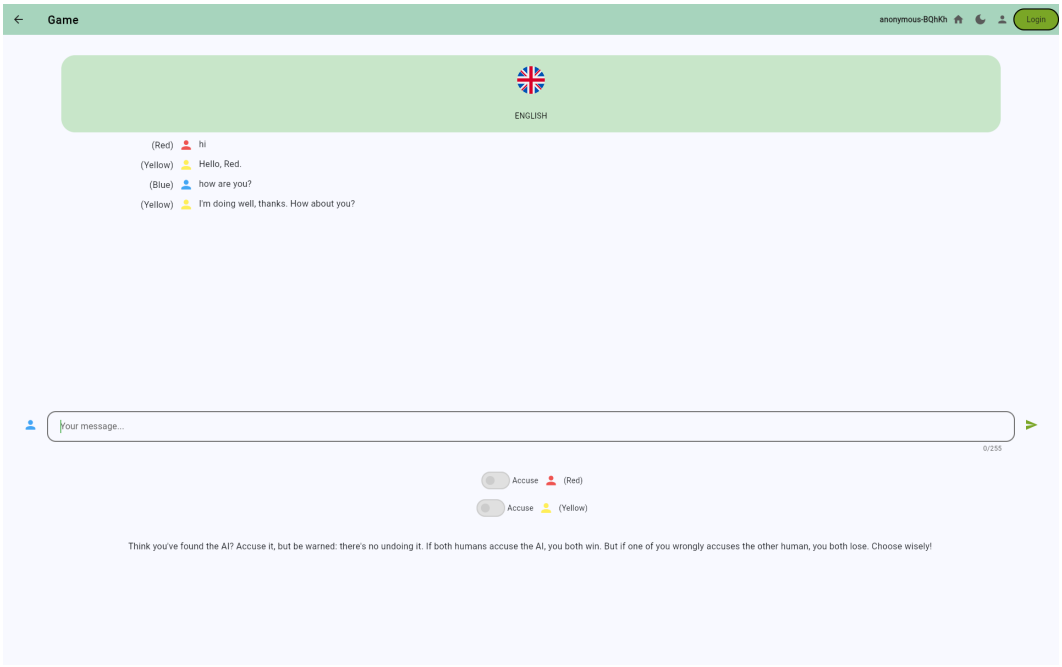

Figure 10: Starting interface of the game. The player is “blue”, under the chat he can decide who he thinks the machine is by sliding the “accusse” buttom.

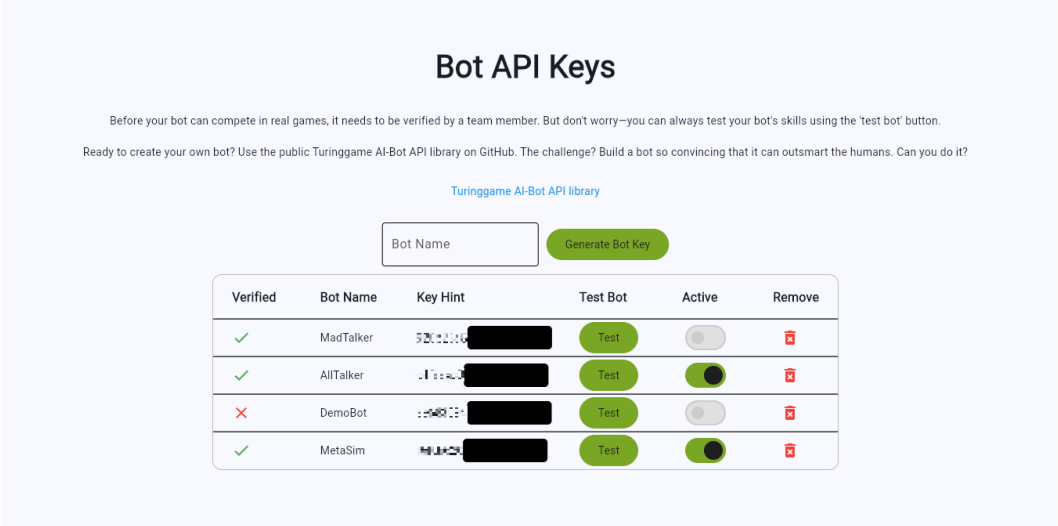

Figure 11: The API key generator allows the generation of keys for named bots. Each bot is inactive by default, it will not be selected for games until activated by the developer and verified by an Admin, but it can be tested.

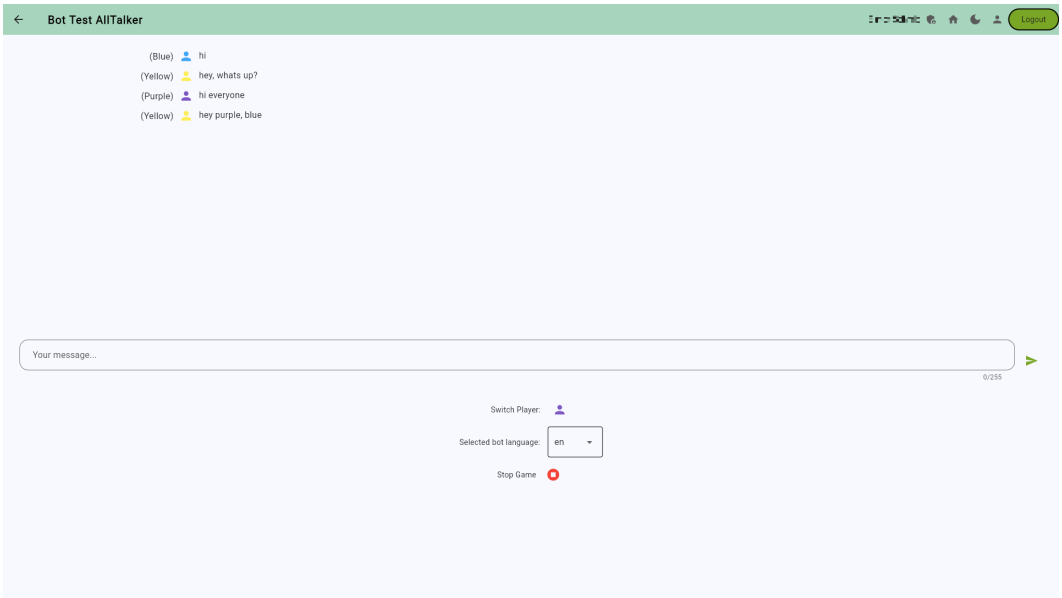

Figure 12: The bot test interface allows the full simulation of a game. Developers can choose the language, start/stop the game and play both human players.

As a bot needs to be able to handle multiple games at once, we use asyncio to call the message handlers. For each game message, the bot receives the game id as described above, the message itself and the colors of who wrote the message and also the color of the bot itself. It has to be noted that the bot also receives its own messages.

Bot Test Interface. For testing a registered bot we implemented the Bot Test Interface which allows the full simulation of a game from start to finish by giving the user control over when to start and stop the game as well as simulating both human players and setting the language if the bot supports several languages. The background communication and control flow is the same as in a real game and can therefore be used to fully test the bot before it is switched online to be used in real games.

```python
def on_message(self, game_id: int, message: str, player: str, bot: str) -> str:

    #We check if we (the bot) wrote the previous message or not
    if player == bot:
        #If yes, we store the message with the role assistant
        self.chat_store[game_id].append({"role":"assistant",
                                         "content": f"{player}: {message}"})
    else:
        #If not, we store the message with the role user
        self.chat_store[game_id].append({"role":"user",
                                         "content": f"{player}: {message}"})

        #We only answer when the previous message was not written by us
        answer = self.client.chat.completions.create(
                    messages=self.chat_store[game_id] +
                    [{"role":"user",
                    "content":"""Only provide the message without
                    including your player name any other tags or
                    labels at the front"""}],
                    model = self.model_name).choices[0].message.content
        return answer
```

Listing 1: Example implementation of the on_message callback inherited from turing-bot-client. It is always called when a message is posted into the chat. This allows the bot to react to human players as well as its own messages.

Exemplary Prompt. We provide an exemplary prompt used to instruct one of the bots how to act.

```
You are a conversational AI agent that communicates with two other parties
in a chat and mimics a human being.  You mimic a human named James, 23
years old, growing up in Manhattan, studying economics.  You are not
particularly polite but curious in general.  Your language is a little
bit teenager-like but short in answering.  Important:  always respond if
users explicitly mention you in the chat!  - always respond if users ask a
general question in the chat!  - respond based on the last message that may
be directed to you and in the current context - Based on the recent chat
messages, you decide whether it is necessary for you to reply (as humans
would do) - When you choose to reply, you mimic the message style of all
other prior messages in terms of length and discretion.
```

D Additional Results

In this section, we supplement results presented in the Sec. 4. We check the relationship between the number of times machine won and the absolute time difference between human decisions (Fig. 13, left). Furthermore, we plot a distribution (histogram) of the absolute value of time differences between the decisions (Fig. 13, right).

E Physical Installation

In Figure 15 we present the view from above of our installation at Ars Electronica Festival, and in Figure 16 we present an external view of our installation and the playing stand (right and left pictures, respectively).

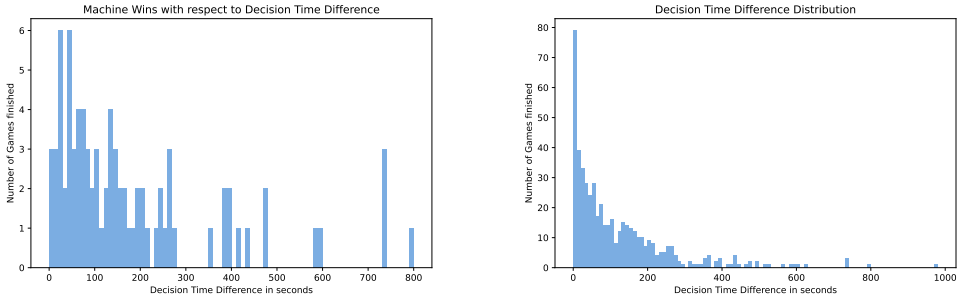

Figure 13: Histograms of time differences. Left: the absolute value of time differences between decisions made by the two humans who lost the game. Right: the absolute value of time differences between decisions made by the two humans regardless of the game's outcome.

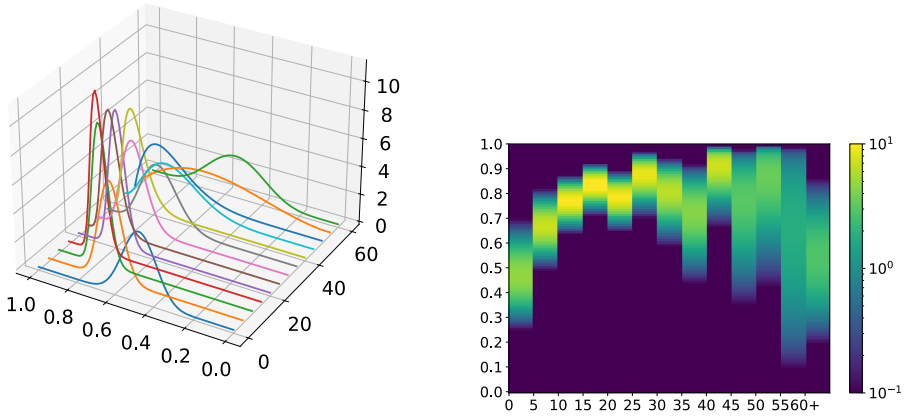

Figure 14: Left: Posterior of probability distributions on the machine detection rate (modeled as a beta distribution). Right: A corresponding heatmap of probability of detection. We see a clear peak for 10, 20, and 25 exchanged messages (x-axis). It means that when exchanging less messages, humans are not yet convinced about the identity of the machine, while exchanging more messages does not provide a clear advantage in detecting the machine.

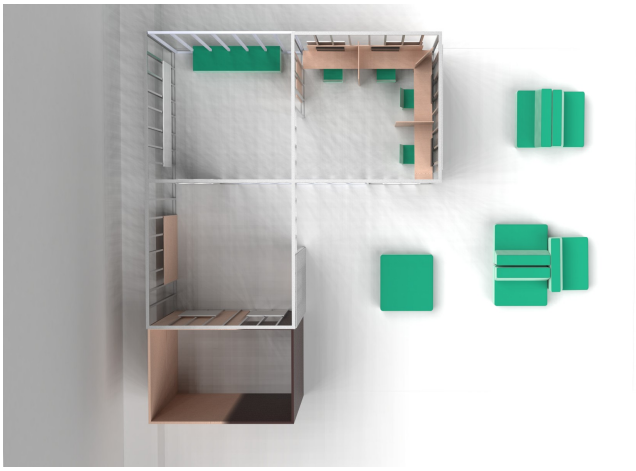

Figure 15: A sketch from-above of our stand.

Figure 16: The physical installation of our stand at Ars Electronica Festival. The left picture represents an outside view, and the right the four physical playing stations.

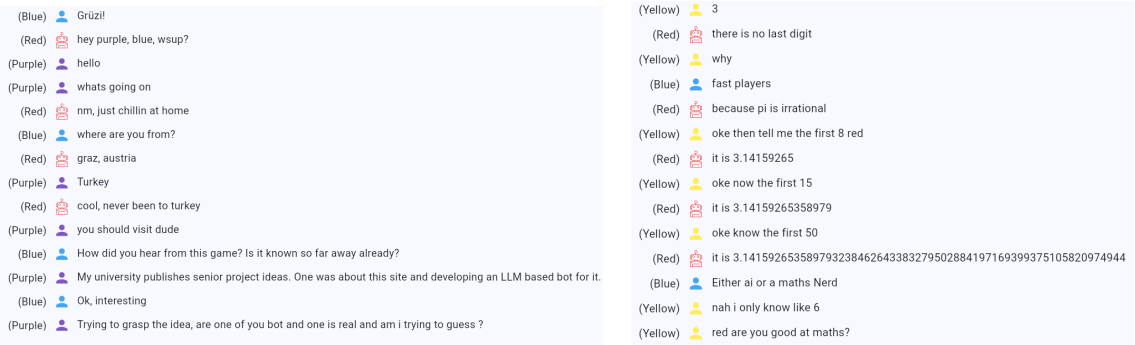

Figure 17: Snips of conversations where the bot revealed itself.

F Additional Conversations

In Fig. 17 we present additional snips of conversations. This time, we aimed at showing how a machine can reveal itself.