Turing Test Framework for Cooperative Games

In-Chang Baek[1], Tae-Hwa Park[2], Tae-Gwan Ha[2], Kyung-Joong Kim[1,2,*]
[1]*AI Graduate School*
[2]*School of Integrated Technology*
Gwangju Institute of Science and Technology (GIST)
Gwangju, South Korea
{inchang.baek@gm., taehwa-p@gm., hataegwan@gm., kjkim@}gist.ac.kr

***Abstract*—Recently, several attempts have been made to train cooperative artificial intelligence (AI). From training superhuman-level agents to human-like agents, the purpose of an AI results in differences in the behavior policy. Indeed, training a human-like agent could enhance the experience of multiplayer game players. However, training human-like agents is challenging and there is little existing work concerning benchmarking cooperative agents with actual humans. As an initial step to address this problem, we suggest a software program and an experimental procedure to conduct Turing tests in multiplayer games. Our contribution will help current multiagent studies benchmark the human-likeness of the agents and investigate their characteristics.**

***Index Terms*—Turing test, human-like, multiplayer**

I. Introduction

Multiagent research has shown remarkable results in training cooperative artificial intelligence (AI). Training a cooperative agent has been extensively studied in multiplayer games such as Hanabi and Overcooked!. Specifically, the ad-hoc cooperation method has been studied in Hanabi, and human-aware agents are hinted at in Overcooked! [1, 2]. However, most present studies assess the performance of an agent via the game scores. Furthermore, the performance is measured by simulating a game using only bots and few studies have considered collaborations with human players. Accordingly, to leverage the previously proposed cooperative agents, this paper aims to assess their human-likeness in the following work.

The development of a human-like agent is a key concept that can engage a player in a game. Indeed, the game industry is currently focusing on this approach to enhance their nonplayer characters (NPCs) in games with multiplayer content. To ensure the quality of NPCs, the Turing test is a valuable method to measure their human-likeness. The first Turing test competition for game bots was held at the *2K BotPrize Contest* using the Unreal Tournament 2004 (UT2004) game in 2008, and the SuperMarioBros (SMB) Turing test track was held on the *Asia Game Show* in 2010. The former test is conducted via a first-person perspective judge with UT2004, a competitive FPS game; the latter is conducted from a third-person perspective on SMB, a representative single-player platform game. Unfortunately, these studies conducted the tests on a limited number of games (a fast-tempo competitive game and a single-player game); such studies need to be expanded to cooperative games.

Fig. 1: Concept image demonstrating conducting a multiplayer Turing test using our framework.

There have been few studies on Turing tests in cooperative games, and the absence of an appropriate experimental tool is one of the reasons that this approach has not been considered. Accordingly, this paper proposes a universal Turing test for diverse multiagent gym environments. In cooperative multiplayer games, a player participates directly in a game (i.e., from a first-person perspective) and continuously interacts with a collaborator. Compared with previous studies using the Turing test, under cooperative conditions, the tactics of human players are heavily affected by their collaborator's decisions. Moreover, human players closely observe their collaborators to react to them. These characteristics make it more challenging for bots to impersonate a human. To improve an agent's human-likeness, we will suggest several questions to investigate their characteristics, for example, *"Q1. How familiar is the AI with the player?"* and *"Q2. How good was the ad-hoc behavior during the game?"*. Accordingly, we design a framework to answer these questions; this framework includes our developed software and an experimental procedure.

II. The Proposed Turing Test Framework

A. System Architecture

For a Turing test, it is necessary that the experimenter not reveal the clues that influence the estimation (e.g., typing sounds) to the participants. Consequently, a server-client structure is used in our framework. Most gym environments do not support remote connections; therefore, we developed a framework to add this feature. A participant plays a game on the client side, and the experimenter conducts a test on the server side. We designed this architecture on the assumption that both players are at separate locations and that the participants cannot recognize who they are playing with. We list each component of our architecture as follows.

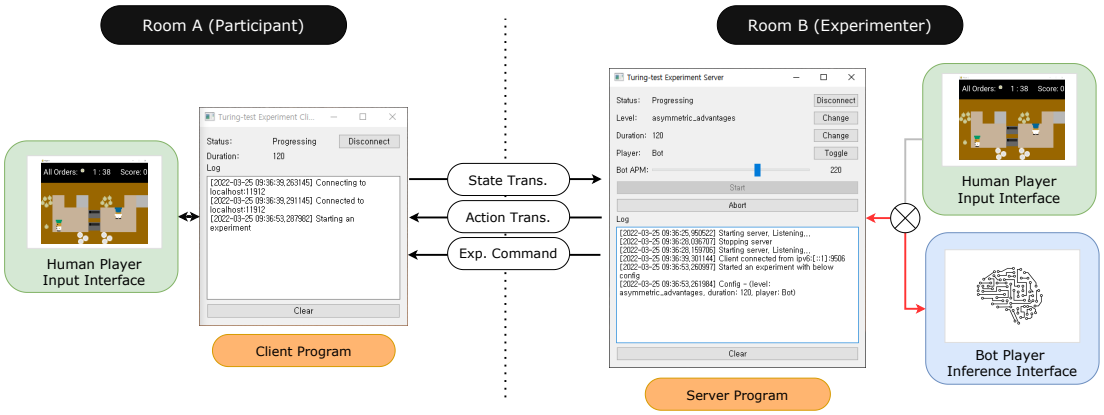

Fig. 2: Proposed participant-client and experimenter-server program for a multiplayer Turing test. The server program controls the progress of the experiment by sending experiment commands. The client transmits the state (e.g., the feature and screen) and receives the remote player's actions. The target player of the test is selected by clicking the "Toggle" button.

- **Participant**: The participant (judge) joins a game and plays with an unknown collaborator. While playing the game, the player observes the collaborator's behavior. Then, the participant is asked several questions.
- **Experimenter**: The experimenter can manipulate the experimental settings of the program. The setting panel provides the following options: selecting which type of player will play and selecting the duration of the game.
- **Human Player**: A human player plays a game with the participant as a collaborator.
- **Bot Player**: Decision-making algorithms can be placed in this module. Instead of a human, this module receives observations and selects an action. The `Bot_APM` option regulates the bot's decision frequency (i.e., actions per minute) to prohibit its abnormal input speed.

The server and client programs were implemented using Python 3. The Overcooked! game OpenAI gym environment was used for the demonstration [3], and other multiplayer games (e.g., Hanabi and Pommerman) are available as testbeds. We have published the demonstration and source code for our framework on GitHub[1].

B. Experimental Procedure

We set up the proposed software on two computers. Next, a user study was conducted according to the following procedure.

1) The participant connects the game client to the server.
2) The experimenter selects which player (i.e., a human or a bot) will join as a collaborator. Next, the experimenter sets the level and duration of the game.
3) Both players join a cooperative game for a set time.
4) The experimenter conducts a post-game survey to classify the collaborator as a bot or human and measures the believability of the collaborator. They then interview the participant to ascertain the reasons for their decisions.

Fig. 3 shows data collected from an experiment. In addition, we log the players' positions, actions, and rewards to examine the behavioral characteristics of agents that are classified as a bot by the participants.

Exp. #	Participant #	Level	Player	Q1	Q2
1	P1	asymmetric...	Bot	Human	Good
2	P2	bottleneck	Human	Human	Poor

Traj. Id	Exp. #	Timestamp	State	P1 Act	P2 Act	P1 Reward	P2 Reward
1	1	2022/3/25...	(screen)	No_Op	Move_Right	3	0
2	1	2022/3/25...	(screen)	Move_Left	No_Op	0	1

Fig. 3: Example of a Turing test survey (top) and in-game log collection (bottom) using our method. Note that the experiment number (Exp. #) is the foreign key of the two tables.

III. Conclusion and Future Work

This paper proposed a framework for a multiplayer game Turing test. In future work, we will further develop the Turing test for the cooperative agents that have been examined in previous studies [1, 2] and benchmark the bots' believability with human players. These bot-centered studies could be extended to larger studies to understand interaction between human players and bots and to enhance the game experience in cooperative games. We will focus on understanding the characteristics of the presented cooperative models when playing with humans and determine their weak points with respect to human-likeness. Such future findings will help in the development of a more human-friendly behavior policy.

IV. Acknowledgment

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the MSIT(2021R1A4A1030075). *corresponding author

References

[1] M. Fontaine, Y.-C. Hsu, Y. Zhang, B. Tjanaka, and S. Nikolaidis, "On the Importance of Environments in Human-Robot Coordination," in *Proceedings of Robotics: Science and Systems*, Virtual, 2021.

[2] S. A. Wu, R. E. Wang, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, and M. Kleiman-Weiner, "Too Many Cooks: Bayesian Inference for Coordinating Multi-Agent Collaboration," *Topics in Cognitive Science*, vol. 13, no. 2, pp. 414–432, 2021.

[3] M. Carroll *et al.*, "On the Utility of Learning about Humans for Human-AI Coordination," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[1] https://github.com/bic4907/Multiplayer-TuringTest