Beating humans in a penny-matching game by leveraging cognitive hierarchy theory and Bayesian learning

Ran Tian, Nan Li, Ilya Kolmanovsky, and Anouck Girard

***Abstract*— It is a long-standing goal of artificial intelligence (AI) to be superior to human beings in decision making. Games are suitable for testing AI capabilities of making good decisions in non-numerical tasks. In this paper, we develop a new AI algorithm to play the penny-matching game considered in Shannon's "mind-reading machine" (1953) against human players. In particular, we exploit cognitive hierarchy theory and Bayesian learning techniques to continually evolve a model for predicting human player decisions, and let the AI player make decisions according to the model predictions to pursue the best chance of winning. Experimental results show that our AI algorithm beats 27 out of 30 volunteer human players.**

I. Introduction

Developing artificial intelligence (AI) to beat humans in strategic games has been drawing attention/interest of researchers for decades [1]–[10]. Although AI algorithms targeted at specific games may not directly contribute to solving practical engineering problems other than those in the gambling industry, the theories developed alongside can be used to attack many problems of similar natures and of greater significance [1].

Many strategic games, for example, the games of chess, Go, and poker, require the players to do intensive calculations to identify winning strategies. AI algorithms for these games often rely on the super computing power of modern computers to beat human players [2]–[5]. In many other strategic games, computing power may have a less decisive effect on the game result. For instance, this holds for the cases where perfectly rational decision strategies are well-known, including the game of matching pennies (or "odds and evens") [11] and the game of rock-paper-scissors [12]. For such games, recognizing the decision pattern of the opponent human player often plays a vital role in developing winning strategies for the AI player [6]–[10].

In this paper, we focus on the game of matching pennies. We develop an AI to play the game repeatedly against a human player, during which the AI decision strategy is continually evolved to pursue the best chance of winning.

Such a problem has been considered by C. E. Shannon in his seminal paper [6], where he named his AI a "mind-reading machine," followed by D. W. Hagelbarger in [7], where the AI won $5,218$ times out of $9,795$ plays. The principle behind their AI algorithms is the hypothesis that human players are not able to generate *i.i.d.*[1] random numbers but tend to follow certain patterns depending on the previous results to make their new decisions. Their AI algorithms pursue the identification of these patterns and assume that the human player will follow the same patterns the next time the same situation arises. Some later works, including [8]–[10] for the game of rock-paper-scissors, essentially follow the same principle to design their AI algorithms.

In this paper, we propose a new AI algorithm, which leverages cognitive hierarchy theory [13]–[15] and Bayesian learning. We also hypothesize that human players follow certain "patterns" in making decisions, but differently from [6] and [7], we explicitly characterize these "patterns" based on the human player's "level of reasoning," introduced in cognitive hierarchy theory. Furthermore, we assume that humans follow these patterns probabilistically, and use Bayesian learning techniques to identify the associated probabilities.

In summary, the contributions of this paper are:

1) We develop a new AI algorithm to play the penny-matching game, which beats 90% of volunteer human players in our experiments.

2) Our algorithm exploits the level-k framework [13], [14] of cognitive hierarchy theory, which has not been considered in previously developed AI algorithms for matching-penny or rock-paper-scissors games [6]–[10].

3) Although cognitive hierarchy theory has been exploited for modeling human-agent behavior in some other application domains, such as automotive [16], [17], aerospace [18], and cyber-physical security [19] applications, probabilistic reasoning-level transitions in sequential decision-making scenarios, considered in this paper, have not been incorporated in these previous works. And the results of this paper suggest that such reasoning-level transitions exist in human decision making and can be modeled.

4) In the light of 3), we envision that the general approach to modeling human behavior proposed in this paper can find its utility in a broader range of application scenarios involving human-machine interactions.

II. Mathematical formulation of the penny-matching game

The penny-matching game under consideration, denoted by $\mathcal{G}$, is a two-player zero-sum game with the normal-form representation given in Table I. Formally, let u^i denote a decision of player i, $i \in \mathcal{P} = \{1, 2\}$, taking values in the

This research has been supported by the National Science Foundation Award Number CNS 1544844.

Ran Tian, Nan Li, Ilya Kolmanovsky, and Anouck Girard are with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA {tianran, nanli, ilya, anouck}@umich.edu.

[1] Independent and identically distributed

set $\mathcal{U} = \{0, 1\}$. The payoffs of the two players, (r^1, r^2), as functions of (u^1, u^2) are defined as follows:

$$r^1 = 1 - 2\,\mathrm{mod}(u^1 + u^2, 2), \tag{1a}$$

$$r^2 = -r^1. \tag{1b}$$

Player 1 \ Player 2	0	1
0	$(1, -1)$	$(-1, 1)$
1	$(-1, 1)$	$(1, -1)$

TABLE I

GAME IN NORMAL-FORM REPRESENTATION.

We let a human player, as player 1, and an artificial intelligence (AI), as player 2, play the game $\mathcal{G}$ repeatedly. For convenience, we use the subscript $t \in \mathbb{N} \cup \{0\}$ to denote the round of the game. For instance, u_t^1 denotes the human player's decision for game round t, and r_t^1 denotes her obtained payoff in this round.

If $r_t^i > 0$, then we say that player i wins the round t. It is easily seen from (1) that there is always one and only one of the two players winning a round. Our goal is to design a strategy for the AI so that it has a higher winning rate than the human player, where the winning rate is defined as the number of wins divided by the total number of game rounds.

It is clear from (1) that if the human player's decision u_t^1 can be correctly predicted, the AI can win the round t by using the following decision strategy:

$$u_t^2 = 1 - \hat{u}_t^1, \tag{2}$$

where $\hat{u}_t^1$ denotes the predicted value of u_t^1.

Therefore, the goal to win can be achieved through developing a model of the human player that can predict u_t^1 with high accuracy.

It is well-known that the above repeated game has a unique Nash equilibrium, which involves independent repetition of the stage-game equilibrium strategy, i.e., for each player to play according to an *i.i.d* process where at each stage 0 or 1 is chosen with equal probability 0.5. We call this the Nash-equilibrium strategy for $\mathcal{G}$. In particular, as long as one of the two players applies this Nash-equilibrium strategy, the game will end in a draw in expectation.

III. MODELING HUMAN PLAYER BASED ON COGNITIVE HIERARCHY THEORY

A. Level-k models of the human player

Cognitive hierarchy theory (CHT) characterizes human decision-making processes based on assumptions of bounded rationality and iterated reasoning. In the level-k framework of CHT, a human decision-maker is assumed to make decisions based on a finite number of reasoning steps, called "level." In the setting of single-shot games, in particular, a level-k player assumes that the other player(s) are level-$(k-1)$, predicts their decisions based on this assumption, and makes her own decision as the optimal response to the predicted decisions of the other players [13], [14].

In order to formulate the level-k, $k = 0, 1, \ldots$, decision strategies of the two players in our game $\mathcal{G}$, we start from defining the level-0 decision rules of the two players as follows:

$$\hat{u}_t^{1,0} = u_{t-1}^2, \tag{3a}$$

$$\hat{u}_t^{2,0} = 1 - u_{t-1}^1. \tag{3b}$$

The above level-0 decision rules are based on the "naive" thought that the other player will make the same decision as in the previous round, which may represent a player's instinctive response to the game.

On the basis of the level-0 decision rules (3), the level-k decision rules of the two players, with $k \geq 1$, are as follows:

$$\hat{u}_t^{1,k} = \hat{u}_t^{2,k-1}, \tag{4a}$$

$$\hat{u}_t^{2,k} = 1 - \hat{u}_t^{1,k-1}, \tag{4b}$$

i.e., the level-k decision of player i optimally responds to the level-$(k-1)$ decision of player $(3-i)$ in terms of maximizing player i's own payoff.

For a given pair $(u_{t-1}^1, u_{t-1}^2) \in \mathcal{U} \times \mathcal{U}$, the level-$k$ decisions of the two players for $k = 0, 1, 2, \ldots$, are summarized in Table II, where we use $[\kappa]$ to denote the set of non-negative integers κ' that satisfy $\mathrm{mod}(\kappa', 4) = \kappa$.

We let $\sigma_t \in \mathbb{N} \cup \{0\}$ denote the human player's level of reasoning for round t. In principle, if $\sigma_t \in [\kappa]$ for some $\kappa = 0, 1, 2$ or 3, then u_t^1 is determined as $u_t^1 = \hat{u}_t^{1,\sigma_t} = \hat{u}_t^{1,[\kappa]}$, which can be read from Table II. However, to account for the sub-optimality and variability in human decision making, we assume that if $\sigma_t \in [\kappa]$, then the human player makes decisions according to

$$\mathbb{P}(u_t^1 = \hat{u}_t^{1,\sigma_t}) = \frac{e^{\theta}}{e^{\theta} + e^{-\theta}}, \tag{5a}$$

$$\mathbb{P}(u_t^1 = 1 - \hat{u}_t^{1,\sigma_t}) = \frac{e^{-\theta}}{e^{\theta} + e^{-\theta}}, \tag{5b}$$

which is based on the "softmax" decision rule [20] with $\theta > 0$ being a tuning parameter.

k \ Player	1	2
[0]	u_{t-1}^2	$1 - u_{t-1}^1$
[1]	$1 - u_{t-1}^1$	$1 - u_{t-1}^2$
[2]	$1 - u_{t-1}^2$	u_{t-1}^1
[3]	u_{t-1}^1	u_{t-1}^2

TABLE II

LEVEL-k DECISIONS OF PLAYERS 1 AND 2.

B. Transitions of human player's reasoning level

In a single-shot game, a player has only one chance to determine her reasoning level, relying on which to make her decision. In a repeated game, in contrast, a player can adjust her level in each round, for instance, according to whether she is winning or losing.

We assume that the human player in our repeated game $\mathcal{G}$ will probabilistically adjust her reasoning level in each round according to the game result of the previous round, in

particular, based on the following transition model,

$$\mathbb{P}(\sigma_t \in [i] \,|\, \sigma_{t-1} \in [j], r^1_{t-1} = 1) = p^+_{(i+1),(j+1)}, \quad (6a)$$

$$\mathbb{P}(\sigma_t \in [i] \,|\, \sigma_{t-1} \in [j], r^1_{t-1} = -1) = p^-_{(i+1),(j+1)}, \quad (6b)$$

defined for all $i, j \in \{0, 1, 2, 3\}$, where $\mathbb{P}(\cdot|\cdot)$ represents conditional probabilities.

However, due to the fact that different humans may have different transition models, the transition matrices $p^+, p^- \in \{p \in [0,1]^{4\times 4} \mid \sum_{i=1}^4 p_{i,j} = 1, j = 1, 2, 3, 4\}$ are not a priori known, but have to be estimated during the game. Note that there are 12 unknown parameters for each of p^+ and p^-, which poses a requirement of a large set of data for their estimation.

Therefore, we pursue a simplification of the transition model (6) by leveraging the following two observations:

1) If the human player won the previous round, i.e., $u^1_{t-1} = u^2_{t-1}$, then $\hat{u}^{1,[0]}_t = \hat{u}^{1,[3]}_t$ and $\hat{u}^{1,[1]}_t = \hat{u}^{1,[2]}_t$.

2) If the human player lost the previous round, i.e., $u^1_{t-1} = 1 - u^2_{t-1}$, then $\hat{u}^{1,[0]}_t = \hat{u}^{1,[1]}_t$ and $\hat{u}^{1,[2]}_t = \hat{u}^{1,[3]}_t$.

In other words, to predict the human player's action for the next round, there is no need to distinguish her level between [0] and [3], and between [1] and [2] if she won the previous round. And similarly, there is no need to distinguish her level between [0] and [1], and between [2] and [3] if she lost the previous round.

On the basis of the above observations, we consider a simplified transition model as follows:

$$\mathbb{P}(\sigma_t \in \Sigma^+_1 \,|\, \sigma_{t-1} \in \Sigma^+_1, r^1_{t-1} = 1) = q^+_1, \quad (7a)$$

$$\mathbb{P}(\sigma_t \in \Sigma^+_2 \,|\, \sigma_{t-1} \in \Sigma^+_2, r^1_{t-1} = 1) = q^+_2, \quad (7b)$$

$$\mathbb{P}(\sigma_t \in \Sigma^-_1 \,|\, \sigma_{t-1} \in \Sigma^-_1, r^1_{t-1} = -1) = q^-_1, \quad (7c)$$

$$\mathbb{P}(\sigma_t \in \Sigma^-_2 \,|\, \sigma_{t-1} \in \Sigma^-_2, r^1_{t-1} = -1) = q^-_2, \quad (7d)$$

where $\Sigma^+_1 = \{[0], [3]\}$, $\Sigma^+_2 = \{[1], [2]\}$, $\Sigma^-_1 = \{[0], [1]\}$, and $\Sigma^-_2 = \{[2], [3]\}$. Note that the probabilistic transitions from Σ^+_1 to Σ^+_2 and from Σ^+_2 to Σ^+_1 under $r^1_{t-1} = 1$ as well as those from Σ^-_1 to Σ^-_2 and from Σ^-_2 to Σ^-_1 under $r^1_{t-1} = -1$ can be computed based on the law of total probability. For instance, $\mathbb{P}(\sigma_t \in \Sigma^+_2 \,|\, \sigma_{t-1} \in \Sigma^+_1, r^1_{t-1} = 1) = 1 - q^+_1$. Furthermore, we assume that the probability of the event $\sigma_t \in \Sigma^\pm_i$ conditioned on $\sigma_{t-1} \in \Sigma^\pm_j$ and $r^1_{t-1} = \pm 1$ is independent of all other events for every pair of $i, j \in \{0, 1\}$.

Suppose that σ_{t-1}, as well as q^+_1, q^+_2, q^-_1 and q^-_2, is known. Then, depending on $r^1_{t-1} = 1$ or -1, the probabilities of set membership $\sigma_t \in \Sigma^+_1$ and $\sigma_t \in \Sigma^+_2$, or the probabilities of $\sigma_t \in \Sigma^-_1$ and $\sigma_t \in \Sigma^-_2$ can be computed, based on which $\hat{u}^{1,\sigma_t}_t$ can be probabilistically predicted. Indeed, when the exact values of $(\sigma_{t-1}, q^+_1, q^+_2, q^-_1, q^-_2)$ are not known, a distribution on $\{[0], [1], [2], [3]\} \times [0, 1]^4$ characterizing the probability of $(\sigma_{t-1}, q^+_1, q^+_2, q^-_1, q^-_2)$ taking each value of $\{[0], [1], [2], [3]\} \times [0, 1]^4$ is sufficient for the above computation and prediction. Specifically, let π_{t-1} denote such a probability distribution, then

$$\mathbb{P}(\sigma_t \in \Sigma^+_1) = \int_{\Sigma^+_1 \times [0,1]^4} q^+_1 \mathrm{d}\pi_{t-1} + \int_{\Sigma^+_2 \times [0,1]^4} (1 - q^+_2)\, \mathrm{d}\pi_{t-1}, \quad (8)$$

if $r^1_{t-1} = 1$, and

$$\mathbb{P}(\sigma_t \in \Sigma^-_1) = \int_{\Sigma^-_1 \times [0,1]^4} q^-_1 \mathrm{d}\pi_{t-1} + \int_{\Sigma^-_2 \times [0,1]^4} (1 - q^-_2)\, \mathrm{d}\pi_{t-1}, \quad (9)$$

if $r^1_{t-1} = -1$.

To facilitate numerical implementation, we assume that $q^\pm_i$ takes values in a finite set $Q \subset [0, 1]$, where Q can be a grid on $[0, 1]$. In this case, the probability distribution π_{t-1} is discrete, and the formula becomes

$$\begin{aligned}\mathbb{P}(\sigma_t \in \Sigma^+_1) = &\sum_{q^+_1 \in Q} q^+_1 \Big(\sum_{\sigma \in \Sigma^+_1} \sum_{q^+_2 \in Q} \sum_{q^-_1 \in Q} \sum_{q^-_2 \in Q} \\ &\pi_{t-1}(\sigma, q^+_1, q^+_2, q^-_1, q^-_2) \Big) + \sum_{q^+_2 \in Q} (1 - q^+_2) \Big(\\ &\sum_{\sigma \in \Sigma^+_2} \sum_{q^+_1 \in Q} \sum_{q^-_1 \in Q} \sum_{q^-_2 \in Q} \pi_{t-1}(\sigma, q^+_1, q^+_2, q^-_1, q^-_2) \Big), \end{aligned} \quad (10)$$

if $r^1_{t-1} = 1$, and

$$\begin{aligned}\mathbb{P}(\sigma_t \in \Sigma^-_1) = &\sum_{q^-_1 \in Q} q^-_1 \Big(\sum_{\sigma \in \Sigma^-_1} \sum_{q^-_2 \in Q} \sum_{q^+_1 \in Q} \sum_{q^+_2 \in Q} \\ &\pi_{t-1}(\sigma, q^+_1, q^+_2, q^-_1, q^-_2) \Big) + \sum_{q^-_2 \in Q} (1 - q^-_2) \Big(\\ &\sum_{\sigma \in \Sigma^-_2} \sum_{q^-_1 \in Q} \sum_{q^+_1 \in Q} \sum_{q^+_2 \in Q} \pi_{t-1}(\sigma, q^+_1, q^+_2, q^-_1, q^-_2) \Big), \end{aligned} \quad (11)$$

if $r^1_{t-1} = -1$.

C. Bayesian learning of human player's model

On the basis of Sections III-A and III-B, the human player's behavior is modeled based on two parts: her reasoning level σ_t for each round t and the parameters $(q^+_1, q^+_2, q^-_1, q^-_2)$ characterizing her reasoning level transitions. Unfortunately, these variables/parameters are neither a priori known nor directly observable. What can be observed are the human player's decision for each round, u^1_t, and the game result for each round, r^1_t. Note that given (u^1_t, r^1_t), the knowledge of u^2_t and r^2_t is redundant since they can be computed using (u^1_t, r^1_t) and (1).

We will use Bayesian learning techniques to learn $x_t = (\sigma_t, q^+_1, q^+_2, q^-_1, q^-_2)$ from the observable data $\xi_t = \{u^1_0, \ldots, u^1_t, r^1_0, \ldots, r^1_t\}$. Specifically, we pursue a probability distribution on $\{[0], [1], [2], [3]\} \times Q^4$, characterizing our belief about the value of x_t, i.e., the π_t defined at the end of Section III-B, conditioned on the available data ξ_t.

To achieve this, we rely on a hidden Markov chain formulation and its corresponding recursive Bayesian inference formula as follows:

If $r^1_t = 1$, then we have

$$\begin{aligned}&\pi_t(\Sigma^+_1 \times \{q^+_1, q^+_2, q^-_1, q^-_2\}) = \\ &\frac{\mathbb{P}(u^1_t | \sigma_t \in \Sigma^+_1)\, \Pi^+_{t-1}}{\sum_{\hat{\mathbf{q}} \in Q^4} \left(\mathbb{P}(u^1_t | \sigma_t \in \Sigma^+_1)\, \hat{\Pi}^+_{t-1,1} + \mathbb{P}(u^1_t | \sigma_t \in \Sigma^+_2)\, \hat{\Pi}^+_{t-1,2} \right)} \end{aligned} \quad (12)$$

where $\sum_{\hat{\mathbf{q}}\in Q^4}=\sum_{\hat{q}_1^+\in Q}\sum_{\hat{q}_2^+\in Q}\sum_{\hat{q}_1^-\in Q}\sum_{\hat{q}_2^-\in Q}$, and

$$\begin{aligned}\Pi^+_{t-1}&=q_1^+\pi_{t-1}(\Sigma_1^+\times\{q_1^+,q_2^+,q_1^-,q_2^-\})\\&\quad+(1-q_2^+)\pi_{t-1}(\Sigma_2^+\times\{q_1^+,q_2^+,q_1^-,q_2^-\}),\end{aligned}\tag{13a}$$

$$\begin{aligned}\hat{\Pi}^+_{t-1,1}&=\hat{q}_1^+\pi_{t-1}(\Sigma_1^+\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\})\\&\quad+(1-\hat{q}_2^+)\pi_{t-1}(\Sigma_2^+\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\}),\end{aligned}\tag{13b}$$

$$\begin{aligned}\hat{\Pi}^+_{t-1,2}&=(1-\hat{q}_1^+)\pi_{t-1}(\Sigma_1^+\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\})\\&\quad+\hat{q}_2^+\pi_{t-1}(\Sigma_2^+\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\}),\end{aligned}\tag{13c}$$

and based on (5),

$$\begin{aligned}\mathbb{P}(u_t^1|\sigma_t\in\Sigma_1^+)&=\begin{cases}\frac{e^{\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[0]}(=\hat{u}_t^{1,[3]}),\\ \frac{e^{-\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[1]}(=\hat{u}_t^{1,[2]}),\end{cases}\\ \mathbb{P}(u_t^1|\sigma_t\in\Sigma_2^+)&=\begin{cases}\frac{e^{-\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[0]}(=\hat{u}_t^{1,[3]}),\\ \frac{e^{\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[1]}(=\hat{u}_t^{1,[2]}).\end{cases}\end{aligned}\tag{14}$$

Similarly, if $r_t^1=-1$, then we have

$$\pi_t(\Sigma_1^-\times\{q_1^+,q_2^+,q_1^-,q_2^-\})=\frac{\mathbb{P}(u_t^1|\sigma_t\in\Sigma_1^-)\,\Pi^-_{t-1}}{\sum_{\hat{\mathbf{q}}\in Q^4}\mathbb{P}(u_t^1|\sigma_t\in\Sigma_1^-)\,\hat{\Pi}^-_{t-1,1}+\mathbb{P}(u_t^1|\sigma_t\in\Sigma_2^-)\,\hat{\Pi}^-_{t-1,2}}\tag{15}$$

where

$$\begin{aligned}\Pi^-_{t-1}&=q_1^-\pi_{t-1}(\Sigma_1^-\times\{q_1^+,q_2^+,q_1^-,q_2^-\})\\&\quad+(1-q_2^-)\pi_{t-1}(\Sigma_2^-\times\{q_1^+,q_2^+,q_1^-,q_2^-\}),\end{aligned}\tag{16a}$$

$$\begin{aligned}\hat{\Pi}^-_{t-1,1}&=\hat{q}_1^-\pi_{t-1}(\Sigma_1^-\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\})\\&\quad+(1-\hat{q}_2^-)\pi_{t-1}(\Sigma_2^-\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\}),\end{aligned}\tag{16b}$$

$$\begin{aligned}\hat{\Pi}^-_{t-1,2}&=(1-\hat{q}_1^-)\pi_{t-1}(\Sigma_1^-\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\})\\&\quad+\hat{q}_2^-\pi_{t-1}(\Sigma_2^-\times\{\hat{q}_1^+,\hat{q}_2^+,\hat{q}_1^-,\hat{q}_2^-\}),\end{aligned}\tag{16c}$$

and based on (5),

$$\begin{aligned}\mathbb{P}(u_t^1|\sigma_t\in\Sigma_1^-)&=\begin{cases}\frac{e^{\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[0]}(=\hat{u}_t^{1,[1]}),\\ \frac{e^{-\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[2]}(=\hat{u}_t^{1,[3]}),\end{cases}\\ \mathbb{P}(u_t^1|\sigma_t\in\Sigma_2^-)&=\begin{cases}\frac{e^{-\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[0]}(=\hat{u}_t^{1,[1]}),\\ \frac{e^{\theta}}{e^{\theta}+e^{-\theta}} & \text{if } u_t^1=\hat{u}_t^{1,[2]}(=\hat{u}_t^{1,[3]}).\end{cases}\end{aligned}\tag{17}$$

Note that in computing (13) or (16), we need to use π_{t-1}, the belief distribution of x_{t-1} on $\{[0],[1],[2],[3]\}\times Q^4$ conditioned on the available data ξ_{t-1}. However, (12) or (15) only provides us with partial information of π_t, that is, $\pi_t(\Sigma_1^+\times\{q_1^+,q_2^+,q_1^-,q_2^-\})$ or $\pi_t(\Sigma_1^-\times\{q_1^+,q_2^+,q_1^-,q_2^-\})$. Note that $\pi_t(\Sigma_2^+\times\{q_1^+,q_2^+,q_1^-,q_2^-\})=1-\pi_t(\Sigma_1^+\times\{q_1^+,q_2^+,q_1^-,q_2^-\})$ and $\pi_t(\Sigma_2^-\times\{q_1^+,q_2^+,q_1^-,q_2^-\})=1-\pi_t(\Sigma_1^-\times\{q_1^+,q_2^+,q_1^-,q_2^-\})$.

To make the propagation (12) or (15) (which one is used depends on the game result r_t^1) recursively computable for all t, we need to reconstruct the distribution π_t from the partial information $\pi_t(\Sigma_1^+\times\{q_1^+,q_2^+,q_1^-,q_2^-\})$ or $\pi_t(\Sigma_1^-\times\{q_1^+,q_2^+,q_1^-,q_2^-\})$. To do so, we rely on the following assumptions:

$$\begin{aligned}&\pi_t([i]\times\{q_1^+,q_2^+,q_1^-,q_2^-\}):\pi_t([j]\times\{q_1^+,q_2^+,q_1^-,q_2^-\})=\\&\pi_{t-1}([i]\times\{q_1^+,q_2^+,q_1^-,q_2^-\}):\pi_{t-1}([j]\times\{q_1^+,q_2^+,q_1^-,q_2^-\})\end{aligned}\tag{18}$$

holds for the pairs $(i,j)=([0],[3])$ and $(i,j)=([1],[2])$ if $r_t^1=1$, and holds for the pairs $(i,j)=([0],[1])$ and $(i,j)=([2],[3])$ if $r_t^1=-1$, and for all $\{q_1^+,q_2^+,q_1^-,q_2^-\}\in Q^4$, meaning that our relative degree of belief in any two indistinguishable[2] levels follows its previous value.

On the basis of (12), (15), and (18), π_t can be computed using π_{t-1}, u_t^1, and r_t^1 for all t.

IV. Decision strategy for the AI player

Using the algorithm (12)-(18), after each round $t-1$, we can obtain a belief distribution π_{t-1} characterizing the human player's model. Then, we compute $\mathbb{P}(\sigma_t\in\Sigma_1^+)$ and $\mathbb{P}(\sigma_t\in\Sigma_2^+)=1-\mathbb{P}(\sigma_t\in\Sigma_1^+)$ using (10) if $r_{t-1}^1=1$, or compute $\mathbb{P}(\sigma_t\in\Sigma_1^-)$ and $\mathbb{P}(\sigma_t\in\Sigma_2^-)=1-\mathbb{P}(\sigma_t\in\Sigma_1^-)$ using (11) if $r_{t-1}^1=-1$.

Suppose that σ_t is known. Then, we let the AI mimic a human player's decision strategy, i.e., a "softmax" decision rule similar to (5) as follows:

$$\mathbb{P}(u_t^2=1-\hat{u}_t^{1,\sigma_t}|\sigma_t)=\frac{e^{\theta}}{e^{\theta}+e^{-\theta}},\tag{19a}$$

$$\mathbb{P}(u_t^2=\hat{u}_t^{1,\sigma_t}|\sigma_t)=\frac{e^{-\theta}}{e^{\theta}+e^{-\theta}},\tag{19b}$$

with $\theta>0$ being the same parameter as in (5). We remark that although the AI does not need to mimic the sub-optimality in human decision making, the strategy (19) creates some randomness in AI decisions, making it harder for the human player to identify the decision algorithm behind the AI, while guaranteeing that the probability of winning is higher than 0.5.

Since σ_t is not exactly known, we let the AI make decisions relying on the predicted distribution of σ_t as follows:

If $r_{t-1}^1=1$, then $\hat{u}_t^{1,[0]}=\hat{u}_t^{1,[3]}=u_{t-1}^2$, and we let

$$\begin{aligned}&\mathbb{P}(u_t^2=u_{t-1}^2)\\&=\sum_{\sigma_t\in\{[0],[1],[2],[3]\}}\mathbb{P}(u_t^2=u_{t-1}^2|\sigma_t)\mathbb{P}(\sigma_t)\\&=\sum_{\sigma_t\in\{[0],[3]\}}\frac{e^{-\theta}}{e^{\theta}+e^{-\theta}}\mathbb{P}(\sigma_t)+\sum_{\sigma_t\in\{[1],[2]\}}\frac{e^{\theta}}{e^{\theta}+e^{-\theta}}\mathbb{P}(\sigma_t)\\&=\frac{e^{-\theta}}{e^{\theta}+e^{-\theta}}\mathbb{P}(\sigma_t\in\Sigma_1^+)+\frac{e^{\theta}}{e^{\theta}+e^{-\theta}}\mathbb{P}(\sigma_t\in\Sigma_2^+)\\&=\frac{e^{\theta}}{e^{\theta}+e^{-\theta}}-\frac{e^{\theta}-e^{-\theta}}{e^{\theta}+e^{-\theta}}\,\mathbb{P}(\sigma_t\in\Sigma_1^+).\end{aligned}\tag{20}$$

Similarly, if $r_{t-1}^1=-1$, then $\hat{u}_t^{1,[0]}=\hat{u}_t^{1,[1]}=u_{t-1}^2$, and we let

$$\mathbb{P}(u_t^2=u_{t-1}^2)=\frac{e^{\theta}}{e^{\theta}+e^{-\theta}}-\frac{e^{\theta}-e^{-\theta}}{e^{\theta}+e^{-\theta}}\,\mathbb{P}(\sigma_t\in\Sigma_1^-).\tag{21}$$

In turn, $\mathbb{P}(u_t^2=1-u_{t-1}^2)=1-\mathbb{P}(u_t^2=u_{t-1}^2)$.

[2]In terms of corresponding to identical $\hat{u}_t^{1,\sigma_t}$.

V. Results

A. Game GUI

We design a Graphic User Interface (GUI), shown in Fig. 1, to represent the game $\mathcal{G}$. In each round, the human player makes a decision between left or right to dig and the AI player makes a decision between left or right to hide the treasure. The human player gains one virtual coin ($r_t^1 = 1$) if both players choose the same side, and loses one ($r_t^1 = -1$) otherwise. The decisions of the two players for the current round are displayed once both decisions have been made and until the human player has made her decision for the next round. The accumulated payoff of the human player up to the current round t, i.e., $\sum_{k=1}^{t} r_k^1$, is shown in the top-middle.

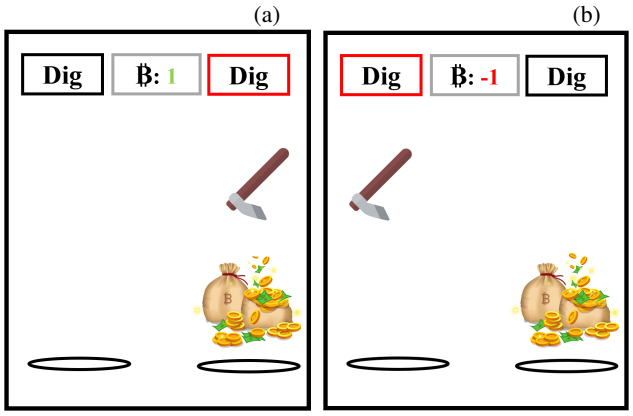

Fig. 1. Game GUI. (a) The human player and the AI player both choose the right side, and thus the human player wins; (b) The human player chooses the left side and the AI player chooses the right side, and thus the human player loses.

B. Results

We recruit human volunteers to play the game against the AI player. In particular, we let each human participant play the game two times, each time with 150 rounds. In one of the two game-runs, the AI uses our proposed strategy leveraging cognitive hierarchy theory and Bayesian learning, described in Sections III and IV. In the algorithm (12)-(21), we use the parameters $Q = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\theta = 1.5$. In the other game-run, the AI uses a Nash-equilibrium strategy, i.e., randomly chooses between left or right with equal probability in each round. The order of these two strategies used in the two game-runs is randomly determined and the human participant knows neither the decision algorithms behind the AI, nor the fact that the AI uses different strategies in the two game-runs.

We have collected data of 30 human participants. We plot the evolution of accumulated payoff of the AI player as the game progresses, $\sum_{k=1}^{t} r_k^2 = -\sum_{k=1}^{t} r_k^1$, in Fig. 2. The thick blue line represents the mean and the light blue shaded area represents the 95% confidence tube of the data for the game-run with our proposed strategy. The thick orange line represents the mean and the light orange shaded area represents the 95% confidence tube of the data for the game-runs with the Nash-equilibrium strategy. It can be observed that when the AI uses our proposed strategy, its accumulated payoff keeps increasing as the game progresses. In contrast, when the AI uses the Nash-equilibrium strategy, its accumulated payoff remains close to 0. This observation verifies the fact that as long as one of the two players applies such a Nash-equilibrium strategy, i.e., chooses between left or right with equal probability in each round, the game will end in a draw in expectation.

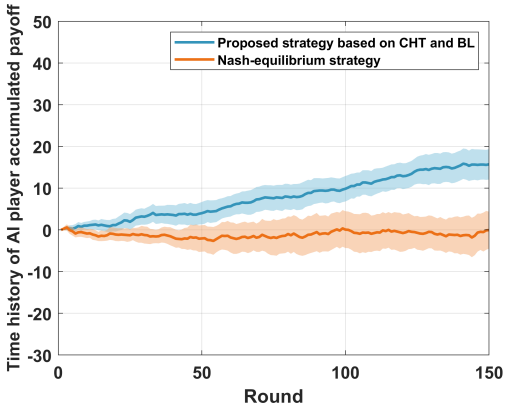

Fig. 2. The evolution of accumulated payoff of the AI player against human players.

Fig. 3 shows the histogram of accumulated payoffs after 150 rounds, $\sum_{t=1}^{150} r_t^1$, of the 30 human participants corresponding to their game-runs where the AI uses our proposed strategy. It can be observed that the AI using our proposed strategy beats 90% of the human players.

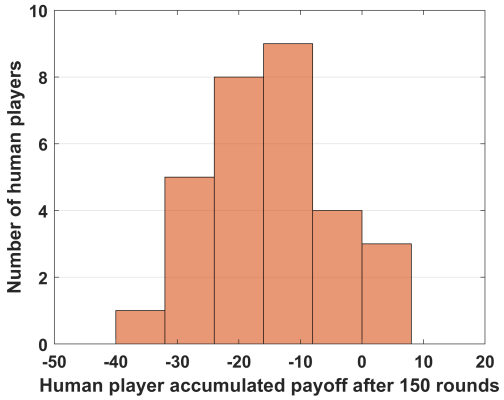

Fig. 3. The histogram of accumulated payoffs of human players against the AI player using our proposed strategy.

Finally, we are interested in the results when player 1 can be perfectly modeled by our proposed human player's model incorporating level-k reasoning and probabilistic reasoning level transitions. Therefore, we create a "fake human" by letting her make decisions according to the "softmax level-k decision rules" defined by (5) and Table II in each round, with her reasoning level σ_t probabilistically transitioned according to the transition model defined by (7) and (18) between consecutive rounds, i.e., a model that perfectly satisfies all of our assumptions. In particular, we randomly generate the values for q_1^+, q_2^+, q_1^-, and q_2^- according to uniform distributions on $[0, 1]$.

We plot in Figs. 4 and 5 the same results as the ones of Figs. 2 and 3 but with the data of real human players replaced by the data generated by "fake human" players. We remark that the q_1^+, q_2^+, q_1^-, q_2^- values are regenerated for each new game-run, thus their values may be different for different game-runs, representing the fact that different humans may have different transition models. Furthermore, their true values are unknown by the AI, and the AI has to estimate their values during the game.

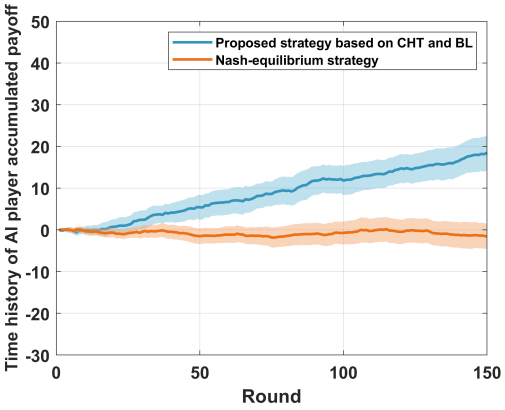

Fig. 4. The evolution of accumulated payoff of the AI player against "fake human" players.

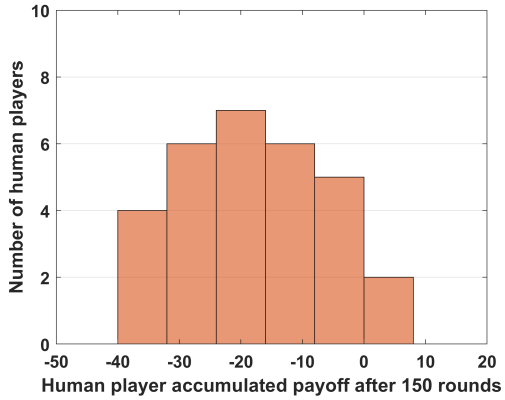

Fig. 5. The histogram of accumulated payoffs of "fake human" players against the AI player using our proposed strategy.

It can be observed that the results of AI against real human players in Figs. 2 and 3 are close to those of AI against "fake human" players in Figs. 4 and 5, although the growth of accumulated payoff of the AI player against real human players is slightly slower than that against "fake human" players. Note that the latter corresponds to the ideal case where the "human" player's behavior perfectly matches the model prediction. Nevertheless, the similarity of the results implies, indirectly, that our proposed human player's model may have captured some crucial features in human decision making in the game. And in the light of this observation, it is reasonable to envision that the proposed approach to modeling human behavior in interactive and sequential decision-making scenarios can find its utility in a broader range of applications involving human-machine interactions.

VI. Summary

By leveraging cognitive hierarchy theory and Bayesian learning, our AI algorithm beat most human players in a repeated penny-matching game. Our approach to modeling human behavior in the game may be extended and used in other applications involving human-machine interactions.

References

[1] C. E. Shannon, "Programming a computer for playing chess," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 314, pp. 256–275, 1950. [Online]. Available: https://doi.org/10.1080/14786445008521796

[2] M. Campbell, A. J. Hoane Jr, and F.-h. Hsu, "Deep blue," *Artificial intelligence*, vol. 134, no. 1-2, pp. 57–83, 2002.

[3] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of Go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[5] N. Brown and T. Sandholm, "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals," *Science*, vol. 359, no. 6374, pp. 418–424, 2018.

[6] C. E. Shannon, "A mind-reading machine," *Bell Laboratories memorandum*, 1953.

[7] D. W. Hagelbarger, "SEER, a sequence extrapolating robot," *IRE Transactions on Electronic Computers*, no. 1, pp. 1–7, 1956.

[8] F. F. Ali, Z. Nakao, and Y.-W. Chen, "Playing the rock-paper-scissors game with a genetic algorithm," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, vol. 1. IEEE, 2000, pp. 741–745.

[9] G. Pozzato, S. Michieletto, and E. Menegatti, "Towards smart robots: Rock-paper-scissors gaming versus human players," in *PAI@ AI*IA*. Citeseer, 2013, pp. 89–95.

[10] M. Zink, P. Friemann, and M. Ragni, "Predictive systems: The game rock-paper-scissors as an example," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2019, pp. 514–526.

[11] D. Mookherjee and B. Sopher, "Learning behavior in an experimental matching pennies game," *Games and Economic Behavior*, vol. 7, no. 1, pp. 62–91, 1994.

[12] Z. Wang, B. Xu, and H.-J. Zhou, "Social cycling and conditional responses in the rock-paper-scissors game," *Scientific reports*, vol. 4, p. 5830, 2014.

[13] D. O. Stahl and P. W. Wilson, "On players' models of other players: Theory and experimental evidence," *Games and Economic Behavior*, vol. 10, no. 1, pp. 218–254, 1995.

[14] M. A. Costa-Gomes and V. P. Crawford, "Cognition and behavior in two-person guessing games: An experimental study," *American Economic Review*, vol. 96, no. 5, pp. 1737–1768, Dec. 2006.

[15] C. F. Camerer, T.-H. Ho, and J.-K. Chong, "A cognitive hierarchy model of games," *The Quarterly Journal of Economics*, vol. 119, no. 3, pp. 861–898, 2004.

[16] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, "Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1782–1797, Sep. 2018.

[17] S. Li, N. Li, A. Girard, and I. Kolmanovsky, "Decision making in dynamic and interactive environments based on cognitive hierarchy theory, Bayesian inference, and predictive control," in *arXiv preprint arXiv:1908.04005*, 2019.

[18] Y. Yildiz, A. Agogino, and G. Brat, "Predicting pilot behavior in medium scale scenarios using game theory and reinforcement learning," in *AIAA Modeling and Simulation Technologies (MST) Conference*, 2013, p. 4908.

[19] A. Kanellopoulos and K. G. Vamvoudakis, "Non-equilibrium dynamic games and cyber–physical security: A cognitive hierarchy approach," *Systems & Control Letters*, vol. 125, pp. 59–66, 2019.

[20] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.