Modeling Human Behavior in a Strategic Network Game with Complex Group Dynamics

(Supplementary Material)

Jonathan Skaggs
Brigham Young University
Provo, UT, USA
jbskaggs12@gmail.com

Jacob W. Crandall
Brigham Young University
Provo, UT, USA
crandall@cs.byu.edu

This appendix contains supplementary documentation and results for the following paper:

Jonathan Skaggs and Jacob W. Crandall. 2026. Modeling Human Behavior in a Strategic Network Game with Complex Group Dynamics. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026)*, Paphos, Cyprus, May 25 – 29, 2026.
https://doi.org/10.65109/NQKD4743

1 OVERVIEW OF THE SUPPLEMENTARY MATERIAL

The supplementary material consists of two parts:

(1) Code and data sets used on the study. These are supplied in a zip file.
(2) The additional explanations and results provided below that support the explanations and results contained in the main paper.

The additional explanations that follow cover six topics:

(1) Section 2: Description of the parameter-based TFT algorithm.
(2) Section 3: Additional information about the EPDM algorithm.
(3) Section 4: Additional information about the error function used by the EPDM and PSO algorithms.
(4) Section 5: A formal description of the metrics used to evaluate the population dynamics of human and agent populations.
(5) Section 6: Additional notes about the first experiment reported in the main paper.
(6) Section 7: Additional notes and results related to the second experiment (user study) reported in the main paper.

2 A PARAMETERIZED-BASED TIT-FOR-TAT AGENT FOR THE JHG

In this section, we defined the parameter-based matching algorithm (tit-for-tat; TFT) discuss in this paper to model human behavior in the JHG.

Reciprocating the behavior of associates in the JHG is not as straightforward as it is in repeated games. A TFT algorithm for the JHG must specify three things. First, the algorithm must specify how the player will allocate tokens in the first round. Second, the algorithm must determine what the player will reciprocate. A player could reciprocate based on either the number of tokens given (or taken) or the amount of influence (number of tokens multiplied by popularity) given (or taken). Third, the algorithm must determine what to do when agents receive more or less tokens from other players than they are able to reciprocate.

The seven parameters specified in Table 1 dictate the token allocations made by the agent. In the hTFT agents used in this paper (hTFT-PSO and hTFT-EPDM), these parameters are tuned to match behavior specified in the training set as closely as possible using the PSO and EPDM modeling methods, respectively.

We define how TFT agents act given these parameters in this section. The algorithm is implemented in the file TFTAgent.h in the supplied code.

2.1 First-Round Token Allocations

The first three parameters in Table 1 (θ_{initKeep}, $\theta_{\text{initAllocSize}}$, and $\theta_{\text{initPercNeg}}$) impact how the agent allocates tokens in the first round. The following rules define the agent's behavior in this round:

- θ_{initKeep} specifies the percentage of tokens the agent keeps in the first round. That is, the agent keeps $\left(\frac{N \cdot \theta_{\text{initKeep}}}{100}\right)$ tokens, rounded to the nearest integer, in the first round. Here, N is the number of tokens it has to allocate. We used $N = 2|I|$ (two times the number of players) in our games.
- The remaining tokens are allocated by randomly selecting players in the game (who will be the recipient of an allocation) and then allocating a subset of tokens to them. The number of tokens allocated to the selected agent is $z = 1 + \text{round}\left((N-1) * \frac{R}{100}\right)$, where round$(\cdot)$ rounds the value to the nearest integer, $R = \max(0, \min(100, \theta_{\text{initAllocSize}} + r_{20} - 10))$, and r_{20} is an integer randomly selected in the interval $[0, 20]$. If z is greater than the number of tokens the agent has remaining to allocate, it is truncated to the remaining number of tokens.
- The allocations made in the previous bullet point can be positive (give) or negative (attack). $\theta_{\text{initPercNeg}}$ specifies the probability that the agent makes each allocation positive or negative.

2.2 Determining What to Reciprocate

In the JHG, there are two ways one can think about matching behavior. First, the agent can just reciprocate the number of tokens (positive or negative) that a player allocated to them in the previous round. However, because the tokens of more popular players are more influential than the token allocations of less popularity people, an agent could also choose to match influence instead of number

Table 1: The set of parameters Θ used in the TFT parameter-based algorithm. Each parameter can take on an integer value in the range $[0, 100]$.

Symbol	Usage
θ_{initKeep}	Specifies the percentage of its tokens the agent keeps in the first round
$\theta_{\text{initAllocSize}}$	Specifies the average number of tokens allocated to selected associates in the first round
$\theta_{\text{initPercNeg}}$	Specifies the probability that an initial token allocation is positive (give) or negative (attack)
$\theta_{\text{matchType}}$	Determines whether the agent matches tokens (< 50) or influence (≥ 50)
$\theta_{\text{notEnough}}$	Determines, in the case of not being able to reciprocate everything, whether the agent scales back all token allocations proportionally (< 50) or prioritizes higher impact relationships (≥ 50)
θ_{tooMany}	Determines what to do in case of having excess tokens. When < 50, the agent scales up all token allocations proportionally. Otherwise, it creates new random allocations.
$\theta_{\text{keepExtra}}$	Specifies the percentage of extra tokens (after reciprocating everything) that the agent keeps

of tokens. The influence exchanged in a token allocation is give by the popularity of the allocator P_i multiplied by the number of tokens allocated. $\theta_{\text{matchType}}$ specifies whether the TFT agent matches tokens or influence. If $\theta_{\text{matchType}} < 50$, then the agent matches tokens. Otherwise, it matches influence.

2.3 Dealing with Excess and Shortfall

Regardless of what the agent chooses to reciprocate (tokens or influence), it can receive more or less in a round than it has the ability to reciprocate. We deal with each case individually.

First, in the case of receiving more than it can reciprocate, the agent uses one of two methods. When $\theta_{\text{notEnough}} < 50$ the agent simply computes the number of tokens needed to match all transactions from each agent, and then scales back all allocations proportionally. When $\theta_{\text{notEnough}} \geq 50$, the agent first reciprocates fully with agents with whom it has historically had the most interactions, determined by $|\mathcal{I}(i,j)| + |\mathcal{I}(j,i)|$, where $|\mathcal{I}(i,j)|$ is the influence of player i on player j in the JHG influence matrix [10]. It continues to fully reciprocate to those agents until it has no remaining tokens. Thus, it will not tend to reciprocate the actions of associates with whom the overall strength of its interactions have been small in the past.

Second, in the case of having more tokens than are needed to reciprocate all of last round's transactions, the agent first ensures that it reciprocates all allocations made to it in the last round by other players. It then keeps a percentage of the remaining tokens specified by $\theta_{\text{keepExtra}}$. Finally, with the $100 - \theta_{\text{keepExtra}}$ percent of the remaining tokens, it simply scales up all token allocations proportionally when $\theta_{\text{tooMany}} < 50$, or, otherwise, randomly allocates tokens (positively or negatively based on $\theta_{\text{initPercNeg}}$) to randomly selected associates.

3 EPDM

The implementation of the EPDM algorithm we used is given in the file EPDM.cpp of the supplied code. Note that we used a gene pool of size $N = 100$ parameterizations in each generation.

4 ERROR FUNCTIONS

Both the PSO and EPDM algorithms, which learn parameterizations from data, require an error or distance function (for example, see Eq. 2 in the main paper). This function takes as input two token allocation profiles, and returns a value specifying the difference between them. We experimented with several such functions, including mean-squared error (MSE) and a custom built function, which we call *property scoring* (PS). In PS, the error is based on how well the two token allocation profiles match across a variety of different properties. Let $\mathbf{a} = (a_1, \cdots, a_n)$ be the target token allocation profile and let $\mathbf{b} = (b_1, \cdots, b_n)$ be a proposed token allocation profile. Here, $n = |I|$ is the number of players, and a_i and b_i are the number of tokens allocated to player i in $\mathbf{a}$ and $\mathbf{b}$, respectively. Then, PS scores the degree to which the token allocation profile $\mathbf{b}$ matches the target token allocation $\mathbf{a}$ as follows:

$$\mathcal{S}(\mathbf{a},\mathbf{b}) = \mathcal{S}^{+}(\mathbf{a},\mathbf{b}) + \mathcal{S}^{-}(\mathbf{a},\mathbf{b}) + \mathcal{S}^{\text{keep}}(\mathbf{a},\mathbf{b}) - \mathcal{P}(\mathbf{a},\mathbf{b}). \quad (1)$$

In this equation, $\mathcal{S}^{+}(\mathbf{a},\mathbf{b})$ is a measure of the degree to positive token allocaitons match. It is derived as a sum of four quantities (each of which produces a value in the range $[0,1]$):

(1) The degree to which $\mathbf{b}$ has positive allocations to the same number of players as $\mathbf{a}$. This quantity is given by

$$\max\left(0, \frac{n - |g^{+}_{\mathbf{a}} - g^{+}_{\mathbf{b}}|}{n}\right),$$

where $g^{+}_{\mathbf{c}} = |S^{+}_{\mathbf{c}}|$, given that $S^{+}_{\mathbf{c}} = \{i | c_i > 0\}$ is set of players that receive positive allocations in profile $\mathbf{c}$.

(2) When $g^{+}_{\mathbf{a}} > 0$, the degree to which $\mathbf{b}$ has positive allocations to the same players as $\mathbf{a}$. This is given by

$$\max\left(0, \frac{\sum_{i \neq k} d(a_i, b_i)}{g^{+}_{\mathbf{a}}}\right),$$

where k is the player allocating the tokens and

$$d(a_i, b_i) = \begin{cases} 1, & \text{if } a_i > 0 \text{ and } b_i > 0 \\ -2, & \text{if } a_i \leq 0 \text{ and } b_i > 0 \\ 0, & \text{otherwise} \end{cases}.$$

(3) When $g^+_{\mathbf{a}} > 0$, the degree to which $\mathbf{b}$ matches the total number of tokens allocated for giving (i.e., positive allocations) as $\mathbf{a}$. This is given by

$$\max\left(0, \frac{g^{\text{sum}}_{\mathbf{a}} - |g^{\text{sum}}_{\mathbf{a}} - g^{\text{sum}}_{\mathbf{b}}|}{g^{\text{sum}}_{\mathbf{a}}}\right),$$

where $g^{\text{sum}}_{\mathbf{c}} = \sum_{i \neq k} \max(0, \mathbf{c}_i)$.

(4) When $g^+_{\mathbf{a}} > 0$, the degree to which $\mathbf{b}$ gives the same number of tokens to each as player as $\mathbf{a}$. This is given by

$$\max\left(0.0, \frac{g^{\text{sum}}_{\mathbf{a}} - \sum_{k \in S^+_{\mathbf{a}}} |a_k - b_k|}{g^{\text{sum}}_{\mathbf{a}}}\right).$$

The quantity $\mathcal{S}^-(\mathbf{a}, \mathbf{b})$ measures how well the profile $\mathbf{b}$ allocates tokens negatively compared to $\mathbf{a}$ (in Eq. 1). It is formed from the sum of the following three quantities:

(1) The degree to which $\mathbf{b}$ has a negative allocation when $\mathbf{a}$ has a negative allocation. If there exists some $i \in I$ such that $a_i < 0$, then this quantity returns a 1 if $b_j < 0$ for some $j \in I$. Otherwise, this quantity returns 0.

(2) The degree to which $\mathbf{b}$ has the same number of tokens used for taking as $\mathbf{a}$. Let $S^-_{\mathbf{c}} = \{i | c_i < 0\}$ be the set of players that receive negative tokens in allocation $\mathbf{c}$. Then this quantity is given by

$$\max\left(0, \frac{n - |\sum_{k \in S^-_{\mathbf{a}}} a_k - \sum_{k \in S^-_{\mathbf{b}}} b_k|}{n}\right).$$

(3) The degree to which $\mathbf{b}$ takes tokens from the same players as $\mathbf{a}$. This is given by $|S^-_{\mathbf{a}} \cap S^-_{\mathbf{b}}|$ (the cardinality of the intersection of the two sets).

Next, $\mathcal{S}^{\text{keep}}(\mathbf{a}, \mathbf{b}) = \max\left(0, \frac{n - |a_k - b_k|}{n}\right)$ compares the number of tokens kept in allocations $\mathbf{a}$ and $\mathbf{b}$.

Finally, $\mathcal{P}(\mathbf{a}, \mathbf{b})$ is a penalty term when profile $\mathbf{b}$ gives to some player i but profile $\mathbf{b}$ takes from player i and vice versa. That is:

$$\mathcal{P}(\mathbf{a}, \mathbf{b}) = \begin{cases} 2, & \text{if } S^-_{\mathbf{a}} \cap S^+_{\mathbf{b}} \neq \emptyset \text{ and } S^+_{\mathbf{a}} \cap S^-_{\mathbf{b}} \neq \emptyset \\ 1, & \text{if } S^-_{\mathbf{a}} \cap S^+_{\mathbf{b}} \neq \emptyset \text{ xor } S^+_{\mathbf{a}} \cap S^-_{\mathbf{b}} \neq \emptyset \\ 0, & \text{otherwise} \end{cases}.$$

The implementation of this property scoring function is given in the file EPDM.cpp of the supplied code (see the function *scoreProposedAllocation*). Note that the code divides $\mathcal{S}(\mathbf{a}, \mathbf{b})$ by 5. This is not strictly necessary, but impacts the specification of δ in determining thresholds for *top performers*.

Because the PS error function always performed better than MSE, we used this scoring function to derive the error function (i.e., $-S(\mathbf{a}, \mathbf{b})$) in training of all of our models.

5 METRICS

In this section, we define how we computed each of the metrics used in the paper to evaluate population dynamics. These metrics are derived from the known quantities of the JHG, including the transaction matrix X, the popularity vector $\mathcal{P}$, and the influence matrix $\mathcal{I}$. Additionally, to capture longer-term (non-discounted) interactions between players, we use the impact matrix. We define the impact matrix prior to discuss each of the metrics used in the paper.

We define the economic actions taken by a player at time τ as a matrix $X(\tau)$. Some metrics, discussed below, use both positive and negative connections while others only one or the other. To distinguish between each case while trying to simplify explanations we distinguish between $X(\tau)$, $X^+(\tau)$, and $X^-(\tau)$ as all connections, positive connections, and negative connections. We apply this to the different derived matrices as well, such as the influence matrix or impact matrix presented below.

5.1 The Impact Matrix

To examine the evolution of friendships and interaction patterns throughout the Junior High Game, we construct the impact graph, a network representation that aggregates player interactions over some duration of the game. This graph encodes social mixing patterns by weighting interactions based on their frequency, the relative popularity of the involved players at the time of interaction, and the corresponding economic action coefficient.

To calculate the impact matrix, we first calculate $J'_{ij}(\tau)$ representing the interaction strength between players i and j at time step τ. It is defined as:

$$J_{ij}(\tau) = \begin{cases} \mathcal{P}(\tau) X_{ij}(\tau) c^{keep}, & \text{if } i = j \\ \mathcal{P}(\tau) X_{ij}(\tau) c^{give}, & \text{else if } x_{ij} \geq 0 \\ \mathcal{P}(\tau) X_{ij}(\tau) c^{take}, & \text{else if } x_{ij} < 0 \end{cases} \tag{2}$$

where $\mathcal{P}_i(\tau)$ represents the popularity of player i at time τ adjusting the weight of interactions based on social influence. $X_{ij}(\tau)$ is the economic action between players i and j with c^{give}, c^{keep}, and c^{take} representing the economic action coefficients that assign weights based on the type of interaction (giving, keeping, or taking) respectively.

The aggregated impact matrix over a time interval $[q, r]$ is given by:

$$J'(q, r) = \sum_{\tau=q}^{r} J(\tau) \tag{3}$$

which sums the impact matrices over all time steps within the interval. We then normalize the interaction weights so that we can compare relative influences across different context as follows:

$$J'(q, r) = \frac{J(q, r)}{\|J(q, r)\|_1} \tag{4}$$

where $\|J'(q, r)\|_1$ is the L1 norm of the matrix, ensuring that the sum of the magnitude of all entries in $J(q, r)$ is constrained within a unit scale.

Additionally, we often look at the impact of just one round. That is, if $q = r$, we use the notation $J'(r)$ to mean $J'(q, r)$ to simplify understanding of our equations.

5.2 Metrics for Wealth and Power

Because the JHG uses popularity (an approximation of Katz centrality [3]) as the primary commodity, we utilize the *mean popularity*, $\mu_{\mathcal{P}}$, as a metric to quantify the productivity of a society. To apply this measure, we first constructed a popularity vector $\mathcal{P} = [\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{n-1}]$ representing the popularity of each agent i at a particular time τ. We then define average popularity as follows:

$$\mu_{\mathcal{P}}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{P}_i(\tau) \tag{5}$$

We additionally employ the Gini Index, $G(\tau)$, as a quantitative metric to assess inequality of popularity across a society. The Gini Index, originally developed as a measure of income inequality [2], ranges from 0 to 1, where 0 indicates perfect equality (i.e., all individuals possess equal popularity) and 1 denotes maximal inequality (i.e., one individual possesses all of the popularity while others have none). $G(\tau)$ was then computed using the following standard formula:

$$G(\tau) = \frac{1}{2n^2 \mu_{\mathcal{P}}(\tau)} \sum_{i,j=0}^{n-1} |\mathcal{P}_i(\tau) - \mathcal{P}_j(\tau)| \tag{6}$$

where n is the number of agents and $\mu_{\mathcal{P}}$ is the mean of the values in vector $\mathcal{P}$. In our analysis, we used the Gini Index (as opposed to the variance of the population) because it is a preferred measure used in measuring inequality in societies in economics [1, 9] and sociology [5].

We used average popularity and the Gini Index as an interpretable and sensitive measures for assessing the mean and spread of popularity in a society.

5.3 Metrics for Quantifying of Economic Behavior

To help understand how individuals in a society interact with each other, we analyze their economic actions using the percent of give, keep and take allocations over societies of each type of agent. Precise definitions of these metrics are as follows:

$$\%\ \text{Give} = \frac{\sum_{i,j}^{|I|} \sum_{\tau=0}^{T} \max(0, X_{ij}(\tau)(1 - \delta_{ij}))}{T * |I|} \tag{7}$$

$$\%\ \text{Keep} = \frac{\sum_{i}^{|I|} \sum_{\tau=0}^{T} X_{ii}(\tau)}{T * |I|} \tag{8}$$

$$\%\ \text{Take} = -\frac{\sum_{i,j}^{|I|} \sum_{\tau=0}^{T} \min(0, X_{ij}(\tau)(1 - \delta_{ij}))}{T * |I|} \tag{9}$$

where $X_{ij}(\tau)$ represents the token allocations of player i toward player j at time τ. The diagonal of the transaction matrix $X(\tau)$ specifies agent keeping, while positive non-diagonal values represent giving and negative non-diagonal values represent taking. T is the number of rounds in the game, $|I|$ is the cardinality of the set I (specifying the number of players in the game), and δ is the Kronecker delta function.

5.3.1 The evolution coefficient. To capture the evolution of economic behaviors in the JHG over time, we use the evolution coefficient. The evolution coefficient quantifies structural changes in the impact matrix over time. Given that the impact matrix $J'(q, r)$ represents normalized interaction patterns over a specified window. This coefficient, denoted C_d, is computed by evaluating differences in consecutive impact matrices, shifted by a delay d:

$$C_d = \sum_{\tau=0}^{r-p-1} \sum_{ij} \left| J'_{ij}(\tau) - J'_{ij}(\tau + d) \right| \tag{10}$$

In words, the evolution coefficient quantifies the cumulative absolute differences between the impact matrices over a specified number of rounds d. A higher value of C_d indicates more substantial shifts in the interaction structure, suggesting dynamic social mixing or evolving economic strategies among players. Conversely, lower values suggest stability in social interactions and player behaviors.

We note that, in the computation of the Mahalanobis distance, the evolution coefficient is measured as the average change in token allocations across one and two rounds of play.

5.4 Mixing Patterns: Measuring Reciprocation

Reciprocity is an important component of cooperative dynamics within agent societies [8]. High reciprocity suggests that players consistently return favors and maintain balanced exchanges, whereas low reciprocity indicates asymmetric relationships, where some players give or take more than they receive. We define the pairwise global reciprocity between agents i and j over the course of an interaction as:

$$R_{ij}^{global} = \begin{cases} 0 & \text{if } i = j; J_{ij}(r_0, r_f) \leq 0; J_{ji}(r_0, r_f) \leq 0 \\ J'_{ji}(r_0, r_f) \frac{J_{ij}(r_0, r_f)}{J_{ji}(r_0, r_f)} & \text{else if } J_{ij}(r_0, r_f) < J_{ji}(r_0, r_f) \\ J'_{ji}(r_0, r_f) & \text{else if } J_{ij}(r_0, r_f) \geq J_{ji}(r_0, r_f) \end{cases} \tag{11}$$

Here, $J_{ij}(r_0, r_f)$ denotes the cumulative reciprocated impact from agent j to agent i over the interval $[r_0, r_f]$, where r_0 if the first round of the game and r_f is the final round of the game. The metric R_{ij}^{global} thus captures the extent to which agent j reciprocates the contributions made by agent i, normalized by the directionally lesser interaction.

We use the overall *global reciprocity coefficient* to define *overall reciprocity*. It is summarizes the aggregate reciprocity across all agent pairs in the population as follows:

$$R_{coeff}^{global} = \frac{\sum_{ij} R_{ij}^{global}}{n(n-1)} \tag{12}$$

where $n = |I|$ denotes the number of players in the game. This coefficient provides a population-level estimate of the prevalence of mutual exchanges, weighted by the magnitude of impact transferred between agents over the duration of the game.

In addition to this cumulative measure, we also define *immediate reciprocity* to capture short-term reciprocation dynamics. The pairwise local reciprocity at round r_m is defined as:

$$R_{ij}^{local} = \begin{cases} 0 & \text{if } i = j; J_{ij}(r_m) \leq 0; J_{ji}(r_m) \leq 0 \\ J'_{ji}(r_{m-1}) \frac{J_{ij}(r_m)}{J_{ji}(r_{m-1})} & \text{else if } J_{ij}(r_m) < J_{ji}(r_{m+1}) \\ J'_{ji}(r_{m-1}) & \text{else if } J_{ij}(r_m) \geq J_{ji}(r_{m-1}) \end{cases} \tag{13}$$

The corresponding *local reciprocity coefficient* is:

$$R^{local}_{coeff} = \frac{\sum_{ij} R^{local}_{ij}}{n(n-1)} \tag{14}$$

This formulation evaluates how frequently short-term exchanges are reciprocated from one round to the next, capturing the temporal responsiveness of agents to recent interactions.

5.5 Mixing Patterns: Measuring Connectivity Using Density and Entropy

Graph density provides a measure of how interconnected a network is at a given time, relative to the maximum possible number of directed edges (excluding self-loops). Just to note, when average across the whole network average in/out degree are proportional to density and therefore we exclude the redundant comparison. Density is defined as:

$$D(\tau) = \frac{1}{n(n-1)} \sum_{i,j=0;i\neq j}^{n-1} A^{+}_{ij}(\tau) \tag{15}$$

Here, $A^{+}_{ij}(\tau)$ represents an adjacency matrix which is 1 if a positive economic action from node i to node j took place in round τ. Otherwise it is 0. The denominator $n(n-1)$ reflects the maximum number of actions in a loop-free graph.

While density describes the average number of associates a player is interacting with, entropy quantifies the distribution of a player's actions among agents. We define a normalized *entropy* score that captures how evenly distributed its outgoing edge weights are. For node i, let the set of *positive* outgoing edge weights be $\{X^{+}_{ij}\}^{n}_{j=1}$. Define the normalized edge weight distribution as:

$$w_{ij} = \frac{X^{+}_{ij}}{\sum_{k=1}^{n} X^{+}_{ik}} \tag{16}$$

The entropy for node i is then given by:

$$H_i = -\sum_{j=0;X^{+}_{ij}>0}^{n-1} w_{ij} \log w_{ij} \tag{17}$$

The maximum possible entropy $H^{\max}_i$ occurs when all outgoing edges from node i are equally weighted, i.e.,

$$H^{\max}_i = \log k_i, \tag{18}$$

where k_i is the out-degree of node i (number of positive-weight outgoing edges). Then entropy is then computed as:

$$H^{\text{norm}}_i = \begin{cases} 0 & \text{if } \sum_j X^{+}_{ij} = 0 \text{ or } k_i = 1 \\ \frac{H_i}{H^{\max}_i} & \text{otherwise} \end{cases} \tag{19}$$

Finally, the average entropy across all nodes provides a summary statistic for the overall spread of edge weight distributions in the network:

$$H^{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} H^{\text{norm}}_i \tag{20}$$

5.6 Measuring Group Formation: Polarization

Finally, we desire to quantify how players cluster and separate. Known metrics for (partially) quantifying such behaviors include the clustering coefficient [6, 11] and modularity [7]. However, many metrics either (1) are not designed for signed, weighted, and time-varying graphs, (2) do not provide comparisons of grouping behavior from graph to graph (e.g., modularity measures how well a community partition describes a graph, not how divided or grouped a graph is), or (3) conflate multiple concepts (e.g., high density tends to produce a high clustering coefficient and thus may not indicate how members of society actually group and segregate). Thus, in this paper, we use a customized metric called *polarization* to quantify how individuals separate into different groups and sub-groups. Related somewhat to the idea of modularity, polarization measures how much individuals would consistently prefer individuals within a designated group as opposed to individuals outside of that group.

Polarization measures the maximum *preference separation* between two distinct (and sufficiently important) sub-groups in the society. *Preference separation* is a method for measuring how separated two distinct sub-groups within society are in a weighted and signed graph. Specifically, it measures how much individuals in a sub-group would prefer (based on immediate connections) any one member of their own sub-group to any one member of the other sub-group. Preference separation between two groups is highest when (1) nodes with each sub-group are well connected, (2) connections between the two sub-groups are weaker than connections within the sub-groups, and (3) the sub-groups cover a larger proportion of the entire society.

Before giving details for how preference separation and polarization are computed, we first define several quantities. Let

$$m_i = \sum_j |\mathcal{I}_{ji}| \tag{21}$$

be the magnitude of influence (both positive and negative) flowing to individual i from all other individuals in society. Furthermore, let

$$w_i = \frac{m_i}{\sum_{j\in I} m_j}, \tag{22}$$

where I is the set of all individuals in the society, be the total proportion of influence that is flowing to individual i.

Next we define a set of dueling sub-groups in society by comparing how society would partition itself if all individuals had to choose between individuals i and j (where $i \neq j$. Let

$$G_{ij} = \{i\} \cup \{k \in I : \mathcal{I}_{ik} > \mathcal{I}_{jk} + \delta\}, \tag{23}$$

where $\delta = 0.04 m_k$, denote the group of individuals in society that would prefer i over j. Similarly, G_{ji} denotes the group of individuals that prefer j over i. Finally, let $G^{\sim}_{ij} = I - (G_{ij} \cup G_{ji})$ denote the set of individuals in society that are indifferent between individuals i and j. Finally, let $W_{ij} = \sum_{k\in G_{ij}} w_i$ denote the collective weight of group G_{ij}.

Preference separation between sub-groups G_{ij} and G_{ji}, denoted $\mathcal{S}(G_{ij}, G_{ji})$, measures the proportion of individual in G_{ij} and G_{ji} that would prefer to stay in the same group if the choice were

between other individuals from the respective groups (other than just i and j). Formally, it is computed as follows:

$$\mathcal{S}(G_{ij}, G_{ji}) = \left(\frac{\sum_{k \in G_{ij}} \sum_{\ell \in G_{ji}} |G_{ij} \cap G_{k\ell}| + |G_{ji} \cap G_{\ell k}| + 0.5|G^{\sim}_{k\ell}|}{|G_{ij}||G_{ji}||I|} \right) \times \left(\frac{\min(W_{ij}, W_{ji}, 0.2}{0.2} \right), \tag{24}$$

where $|\cdot|$ denotes the cardinality of a set. Note that the last item in the numerator of left term gives half credit when individuals are neutral between the two groups for a particular pairing. The right component of the equation scales down preference separation if one of the groups is too small (determined as having a weight of less than 0.2).

Note that measures of separation tend to be between 0.5 and 1. As such, we base polarization on a normalized version of preferences separation, given by $\mathcal{S}'(G_{ij}, G_{ji}) = \frac{\mathcal{S}(G_{ij}, G_{ji}) - 0.5}{0.5}$.

Once preference separation is computed for all pairs of subgroups, polarization is determined to be the max preference separation for pairs of subgroups. That is, polarization (denoted Γ) is given by $\Gamma = \max_{i,j} \mathcal{S}'(G_{ij}, G_{ji})$. The constraint to only consider subgroups of a particular size ensures that the preference separation between the groups could have a meaningful impact on society.

All of the above description omit the time t – polarization is computed for each time period. The actual metric we report is the average polarization over all rounds considered. We note that this metric is not computationally efficient in this formulation, but it can be computed and utilized effectively for small societies such as those we consider in this paper.

5.7 Summary Metric: Mahalanobis Distance

Each of the above metrics represent important aspects of population dynamics. To further compare and contrast human and agent populations, we aggregate these metrics into a multivariate Gaussian. That is, we quantify the dissimilarity between populations with respect to these metrics using the Mahalanobis distance [4]. Unlike the Euclidean distance, which assumes independence across dimensions, the Mahalanobis distance incorporates the covariance structure of the data, making it well-suited for comparing complex, interdependent societal metrics.

Given two societal types, we compute the Mahalanobis distance between them as:

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})} \tag{25}$$

where $\mathbf{x}$ is the vector of observed societal metrics for a given type, $\boldsymbol{\mu}$ is the mean vector of another societal type, and Σ is the covariance matrix of the distribution of human behavior.

To complement the Mahalanobis distance, we compute the Chi-Squared p-value, which indicates the statistical significance of the observed differences. Since the squared Mahalanobis distance follows a Chi-Squared distribution with degrees of freedom equal to the number of dimensions, the p-value provides a probabilistic interpretation of the similarity between societal types. Lower p-values suggest significant differences between the compared distributions, while higher p-values indicate that the societal structures are statistically indistinguishable.

This methodology enables us to identify when two societies exhibit fundamentally different organization patterns and to characterize the extent of these differences.

6 EXPERIMENT A: ADDITIONAL DETAILS

Additional implementation details and results from the first experiment reported in the main paper (Section 4):

- We use the same JHG parameter settings as was used by Skaggs et al. [10]. That is, $\alpha = 0.2$, $\beta = 0.5$, $c_{\text{give}} = 1.3$, $c_{\text{keep}} = 0.95$, $c_{\text{take}} = 1.6$. Furthermore, each agent had $2 * |I|$ tokens to allocate in each round.
- eCABs were evolved with a single set of genes (rather than the 3 sets used by Skaggs et al. [10]) and were trained for 200 generations with variable initial probabilities. Simulation results were conducted with randomly selected parameterizations from the 100 parameterizations that were in the final generation.
- No GPU was used in the study. No high-powered computational equipment were used (only standard computing equipment with multiple cores). Training using the PSO algorithm was done on a 32-core machine, which took about 3 hours per run. Training using the EPDM algorithm took about 30 minutes on a machine with 16 cores.

7 EXPERIMENT B: USER STUDY DETAILS

Additional information about the second experiment (user study) reported in the main paper (Section 5):

- The user study was approved by the authors' institutional review board (it was given exempt status). Participants gave consent to participated in the study through the approved process.
- Participants were paid approximately $16 (USD) per hour while participating in the study. A high-score list (based on final popularity in each game) was kept to motivate participants to perform well.
- Games were played using the publicly available online platform published by Skaggs et al. [10] (permission was obtained from the authors of that work to use the platform for this study). This online platform can be found at juniorhighgame.com.
- Story plots for all eight games played in the user study are provided below. In the graphs, arrows indicate token transactions in the current round (green = give; red = take), while nodes more connected historically tend to be closer to each other.

REFERENCES

[1] A. B. Atkinson et al. 1970. On the measurement of inequality. *Journal of Economic Theory* 2, 3 (1970), 244–263.

[2] C. Gini. 1921. Measurement of Inequality of Incomes. *The Economic Journal* 31, 121 (1921), 124–126.

[3] Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.

[4] Prasanta C. Mahalanobis. 1936. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India* 2, 1 (1936), 49–55.

[5] B. Milanovic. 2011. *Worlds apart: Measuring international and global inequality.* Princeton University Press.

[6] Mark E. J. Newman. 2003. The Structure and Function of Complex Networks. *SIAM Rev.* 45, 2 (2003), 167–256.

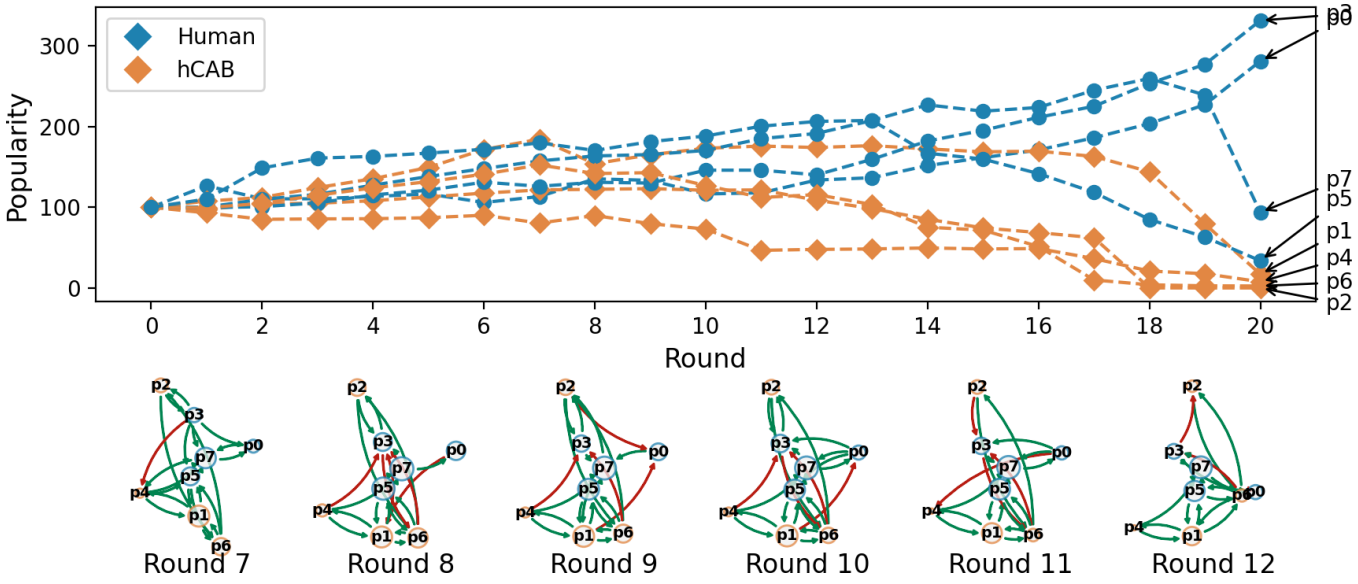

Figure 1: Game code: DKWH

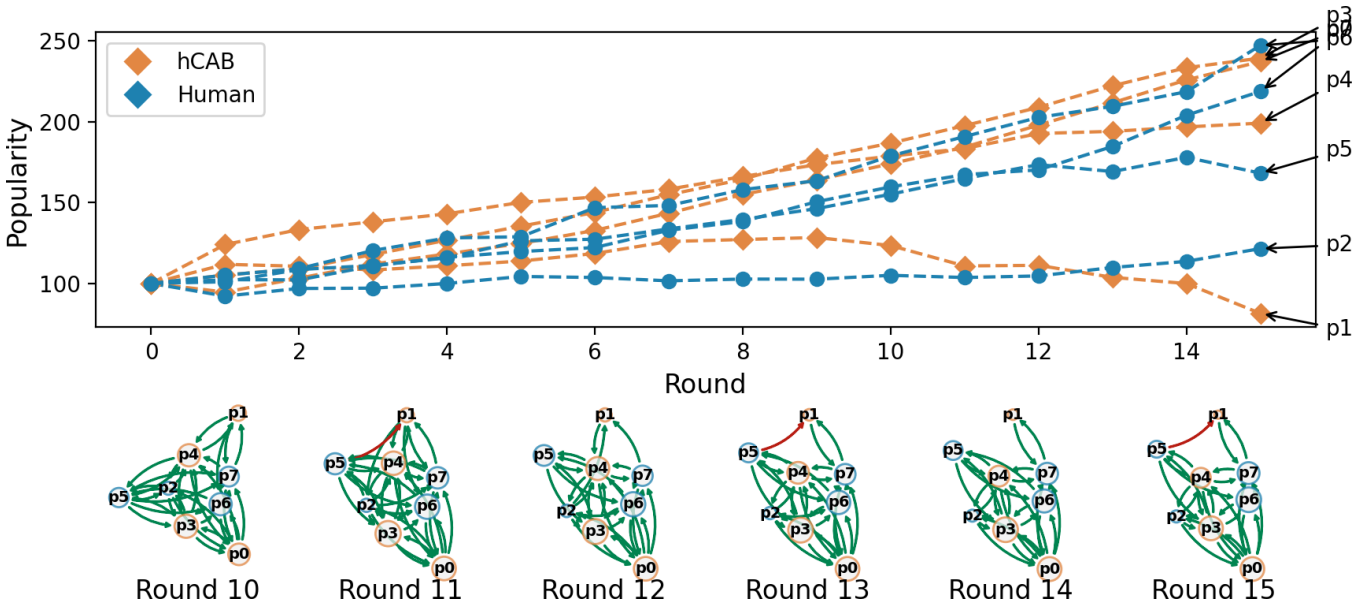

Figure 2: Game code: GPZQ

[7] M. E. J. Newman and M. Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69, 2 (2004), 026113.

[8] Martin A. Nowak. 2006. Five Rules for the Evolution of Cooperation. *Science* 314, 5805 (2006), 1560–1563.

[9] A. Sen and J. Foster. 1973. *On Economic Inequality.* Oxford University Press.

[10] Jonathan Skaggs, Michael Richards, Melissa Morris, Michael A. Goodrich, and Jacob W. Crandall. 2024. Fostering Collective Action in Complex Societies using Community-Based Agents. In *Proceedings of the International Joint Conference on Artificial Intelligence.* IJCAI, Jeju, South Korea, 211–219.

[11] Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393, 6684 (1998), 440–442.

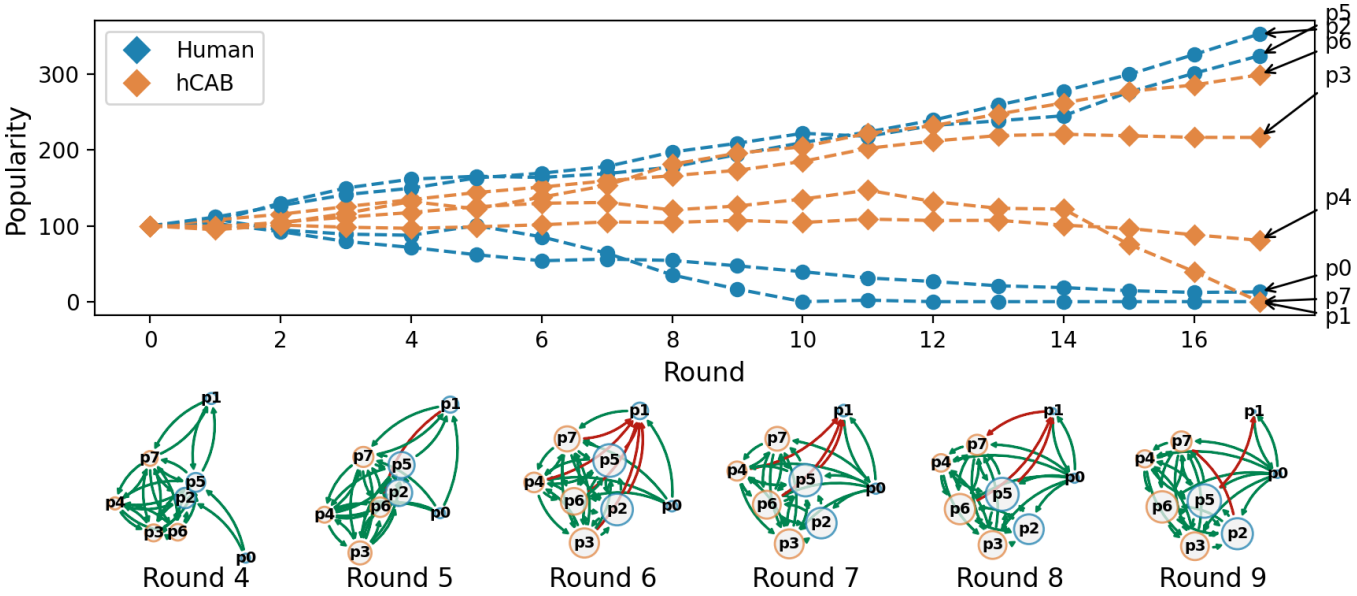

Figure 3: Game code: MGQZ

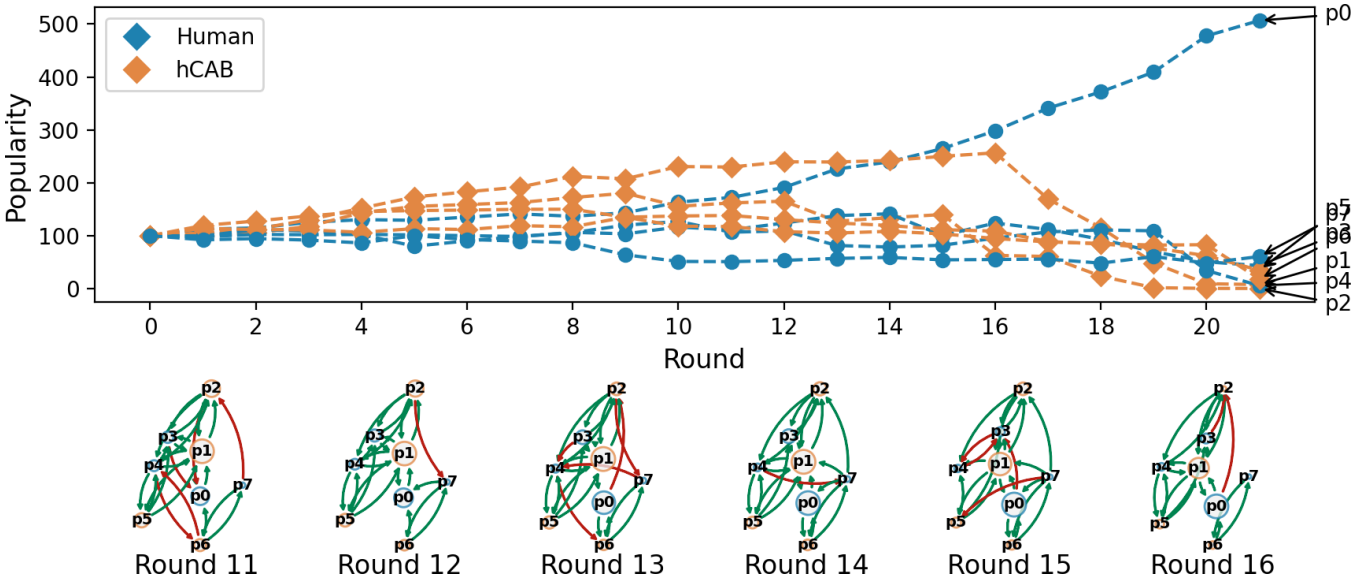

Figure 4: Game code: MSVL

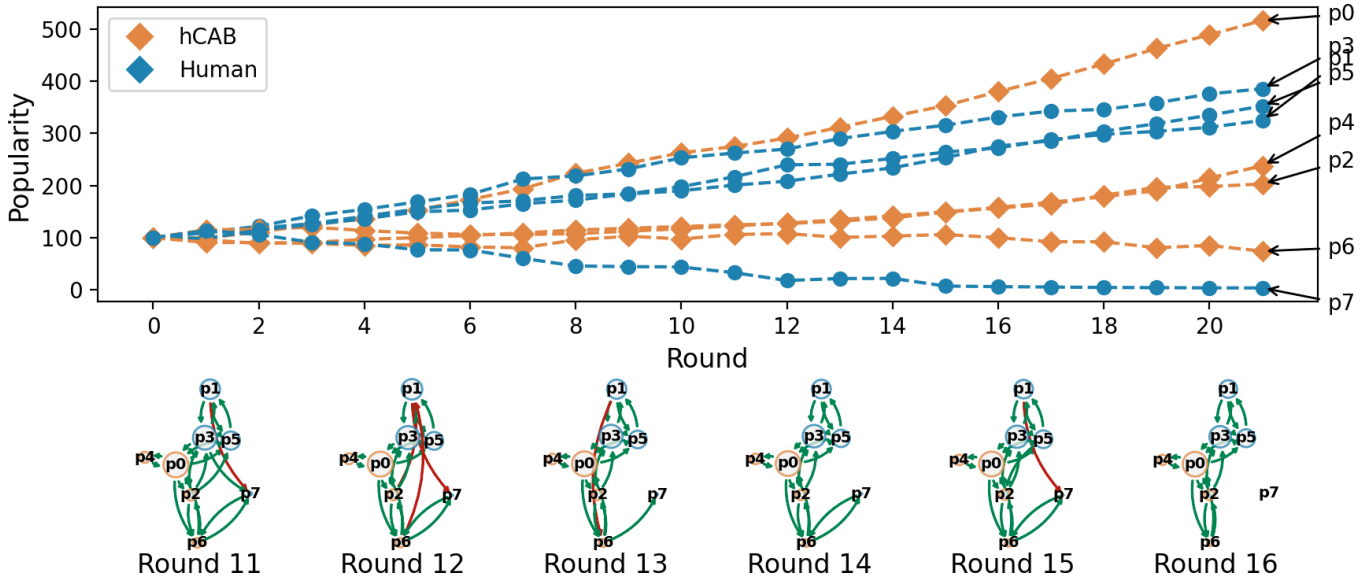

Figure 5: Game code: NSRL

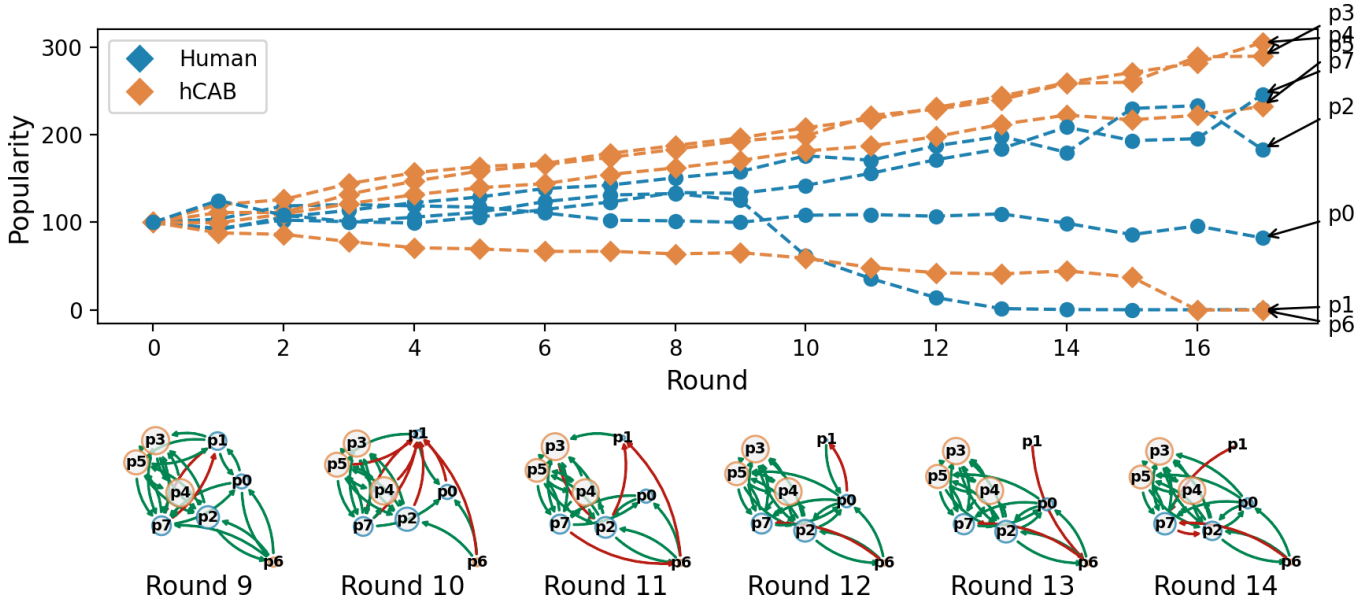

Figure 6: Game code: VKTJ

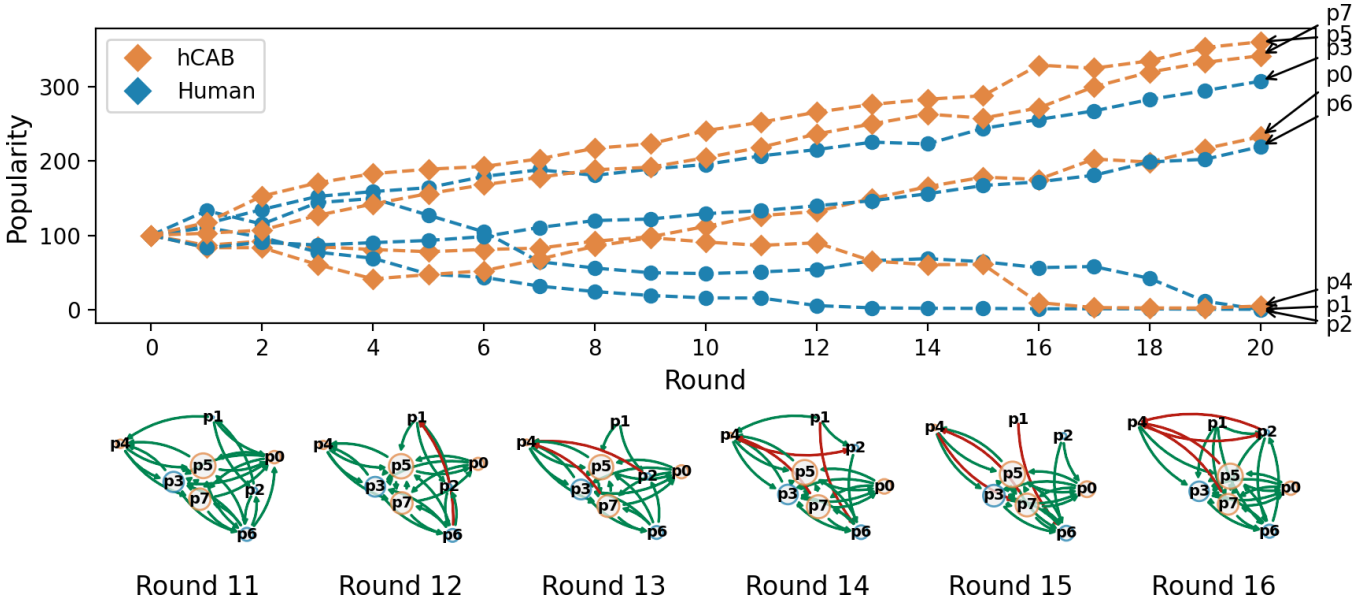

Figure 7: Game code: WTHJ

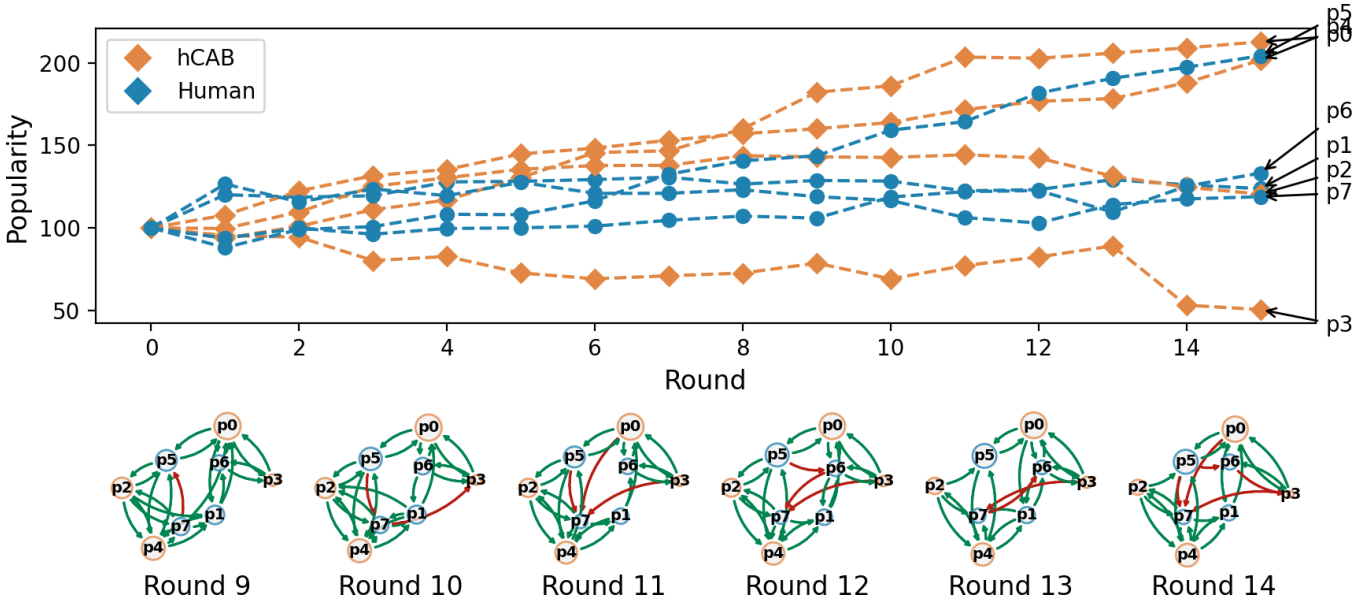

Figure 8: Game code: ZSMT