

UNCERTAINTY VISUALIZATION FOR CHARACTERIZING HETEROGENEOUS HUMAN BEHAVIORS IN DISCRETE DYNAMICAL SYSTEM MODELS

XUEYING LIU*, ZHIHAO HU[†] and XINWEI DENG[‡]

*Department of Statistics, Virginia Tech, 800 Washington St. SW,
Blacksburg, VA 24061, USA*

**xliu96@vt.edu*

†huzhihao@vt.edu

‡xdeng@vt.edu

CHRIS J. KUHLMAN

*Biocomplexity Institute, University of Virginia,
1827 University Avenue,
Charlottesville, VA 22904, USA
hugo3751@gmail.com*

Received 10 March 2023

Revised 29 June 2023

Accepted 6 July 2023

Published 24 August 2023

When modeling human behavior in multi-player games, it is important to understand heterogeneous aspects of player behaviors. By leveraging experimental data and agent-based simulations, various data-driven modeling methods can be applied. This provides a great opportunity to quantify and visualize the uncertainty associated with these methods, allowing for a more comprehensive understanding of the individual and collective behaviors among players. For networked anagram games, player behaviors can be heterogeneous in terms of the number of words formed and the amount of cooperation among networked neighbors. Based on game data, these games can be modeled as discrete dynamical systems characterized by probabilistic state transitions. In this work, we present both Frequentist and Bayesian approaches for visualizing uncertainty in networked anagram games. These approaches help to elaborate how players individually and collectively form words by sharing letters with their neighbors in a network. Both approaches provide valuable insights into inferring the worst, average, and best player performance within and between behavioral clusters. Moreover, interesting contrasts between the Frequentist and Bayesian approaches can be observed. The knowledge and inferences gained from these approaches are incorporated into an agent-based simulation framework to further demonstrate model uncertainty and players' heterogeneous behaviors.

Keywords: Networked anagram game data; discrete dynamical systems; models of heterogeneous behaviors; uncertainty visualization; agent-based simulation; interpretable inference.

* Corresponding author.

1. Introduction

Anagram games have been widely and persistently used for studying human behaviors in every decade since the 1960s, e.g. [8, 20, 21, 23, 27, 29, 30, 32–34, 38, 39, 62, 75–77, 79, 82, 86, 87]. One recent study of group anagram games [23] attracted large attention, where players are connected in an in-person group and cooperate to form words. The work in [22] considered *online* group anagram games, which imposes a *network* on game players to control their interactions. The experimental networked group anagram game (NGrAG) setup is shown in Fig. 1(b). In this work, our focus is to characterize the heterogeneous aspects of player behaviors in the group anagram games.

1.1. Setup of networked anagram game: Human subjects experiments

Human subjects were recruited using Amazon Mechanical Turk (AMT), where players agreed to meet at particular dates and times to play a NGrAG. Subjects played the game on a customized software platform that was designed and constructed specifically for this work. This platform recorded all user button clicks (i.e. user activities) in time during a game. Players (i.e. human subjects) play the game through web browsers. Here we provide a brief description of the game setup. More details of this game can be found in [22]. A random regular network (where all nodes have the same number of neighbors) is prescribed by researchers, where this degree is changed across games. Players are the nodes in the network and edges represent communication channels for pairs of players to interact. A game has a 5-min duration, and each player is provided three initial letters. The players’ objective is to form

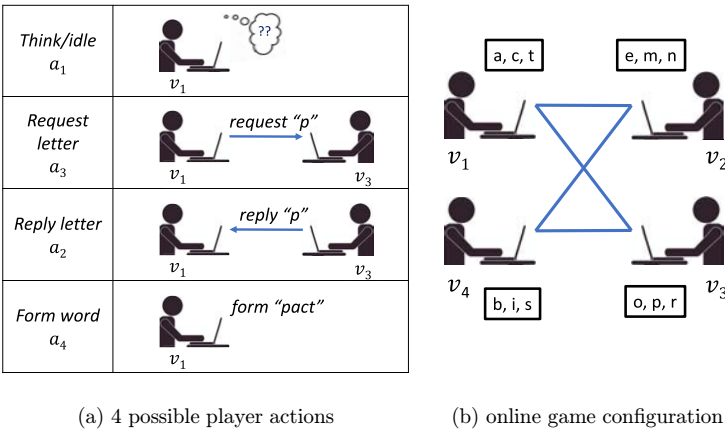


Fig. 1. (Color online) (a) Four actions that may be taken by any player, at any time during the 5-min duration networked group anagram game (NGrAG). Actions can be repeated by a player any number of times. The action vector a is $a = (a_1, a_2, a_3, a_4)$, with a_i given in the graphic. (b) Illustrative NGrAG with four remote players (v_1 through v_4) and four communication channels (in blue). Players participate through their web browsers. A player’s initially assigned letters are in boxes.

as many words as possible *as a team*. Words must be at least three letters. A key feature in this game is that players can share their initial letters with their distance-1 neighbors: a player requests letters, one at a time, and the receiver of a request decides whether she will supply the requested letter. When a player v_i shares a letter, she retains a copy of the letter. Such a setting is to motivate (mutual) assistance among players. Also, once a letter is acquired, there is no mechanism or action by which a player loses a letter (e.g. a player that uses letters to form one word can use those letters again to form other words). A player is free to take all of the actions over time in Fig. 1(a). An illustrative game setup is provided in Fig. 1(b).

Over 240 experiments (i.e. games) were performed using between 4 and 15 players per game. Game duration of 5 min was initially based on in-person group anagram game experiments [23]; that study was the inspiration for our online experiment. The game duration was also evaluated from preliminary experimental data. One concern was that players would become bored and therefore disengage before the game ended. Early experiments demonstrated that players were engaged, since they continued to execute actions during the 300 s. The data also indicated that 300 s was sufficient to see patterns of human behavior in the data.

1.2. Key contributions

Our first contribution of this work is to formulate players' behaviors in NGrAGs as discrete dynamical systems (Sec. 2) and to develop both a Frequentist approach and a Bayesian approach for visualizing heterogeneous player behavior. The proposed methods emphasize the visualization of uncertainty in a comprehensive manner using a two-dimensional bubble plot described in Sec. 3. In such a plot, the location of each bubble represents the probability of the players' next action and the size of the bubble allows for easy detection of variability among the players. Therefore, such plots can reflect uncertainties of a player's activeness (i.e. non-idle actions) and ability to form words (on which game awards are based). Moreover, the plots provide different types of insights at different scales. For a smaller collection of data, differences among data points can be observed, including worst, average, and best behaviors, and these differences can be plotted as a function of modeling inputs. At larger scales, different collections of data can be compared, e.g. different behavioral clusters.

Second, the behavior models are used to formulate agent-based models (ABMs); these ABMs are integrated into a discrete time agent-based simulation (ABS) system (Sec. 5). We study the variability in replicate simulations; heterogeneous behaviors across different clusters; differences in performance for worst, average, and best behaviors within clusters; and heterogeneity and variability in words formed by agents under the Frequentist and Bayesian modeling approaches. For example, within a behavioral cluster, the percentage difference in numbers of words formed between worst and best behaviors can be 70% to 90%, thus demonstrating the utility of using uncertainty visualization to identify these different behaviors in clusters.

The ABM results based on Frequentist and Bayesian methods are consistent with the patterns of the bubble plots described in Sec. 4. We also demonstrate interactions among agents in a game network by varying the models of the neighbors of an agent v_k (keeping v_k 's behavior fixed), and illustrating that the performance of v_k changes.

Our third contribution is to show that the proposed uncertainty visualization methods greatly enhance the explainability of behaviors and variance in complex systems, which herein is the NGrAG. By presenting the bubble plots and simulation results of both Frequentist and Bayesian approaches, we compare the modeling of NGrAGs between the two methods and gain meaningful insights into the similarities and differences in their uncertainty quantification. The comparison reveals that the Frequentist approach is appropriate when the sample size is large, while the Bayesian approach is more suitable for small sample sizes or when prior information is available. This is because when the sample size is small, the MLE of the Frequentist approach may have a large variance, and the parameter may not be well-approximated by a normal distribution. However, the Bayesian approach can alleviate the data scarcity issue since it does not rely on asymptotic normal properties and the memorylessness property of the MCMC sampling technique can avoid extreme values.

Modeling NGrAGs is valuable because data from anagram games are used for various purposes. First, anagram games are considered to be complex tasks [20], so anagram-based studies have been undertaken to understand how people reason under various constraints. For example, single-word anagram games (where an individual must form a unique word from a given set of letters) use different amounts of shuffling of letters to test how much longer it takes participants to form words with increased shuffling [62]. Researchers have used anagram games to study the effects of anxiety on task performance [68]. Other studies have investigated attribution of performance, finding that people who perform well in anagram games attribute their success to skill, while those who do not perform well attribute their results to bad luck, e.g. [79]. Still others have studied individual versus group performance [8]. Group anagram games were studied in [23] as a priming activity to induce collective identity within a group. (Collective identity is an individual's cognitive, moral, and emotional connection with a broader community, category, practice, or institution [70]). Our work was motivated by [23]. It is evident that anagram games are used to explore wide ranging human behaviors, and these efforts provide interesting topics and motivation for modeling.

1.3. Paper organization

The remainder of this paper is organized as follows. In Sec. 2, we present our modeling of the NGrAG as a discrete dynamical system (DDS), including Frequentist uncertainty quantification and an extension to a Bayesian approach. In Sec. 3, we describe uncertainty visualization methods for both Frequentist and Bayesian approaches. We then present visualization results and an explanation of

players' behaviors in Sec. 4. In Sec. 5, we go beyond the conditions for which experiments were conducted and provide simulations that demonstrate results for the two approaches. Section 6 contains our literature review, and Sec. 7 summarizes our work and discusses possible future directions.

This work extends a preliminary version of the work [60] by expanding related work; extending a Bayesian model of the NGrAG; introducing a Frequentist uncertainty visualization approach; explaining in more detail the Bayesian uncertainty visualization method; comparing the Frequentist and Bayesian uncertainty visualization results; and greatly extending the ABS results to more behavioral groups and clusters, contrasting worst, average, and best behavior results, and comparing results from both Frequentist and Bayesian approaches.

2. Discrete Dynamical Systems Model of Individual Player Behavior in Group Anagram Games

Since the NGrAG action history of a player is complex (multiple repeated actions over time), there is insufficient data to generate a model from one player's data, without suffering impractically large uncertainties in the model. Therefore, we group players by their experimental performance in the game along two dimensions: their propensity to cooperate with other players in the game (a measure of cooperation) and the number of words formed (which is how team performance is rewarded). Then we generate a model for each group or cluster of players, resulting in heterogeneous quantification of player behaviors.

Our model of the NGrAG is formulated as a discrete dynamical system (DDS) (see [1, 2, 10, 40, 44, 45, 89] for representative works), and in particular, as a graph dynamical system (GDS). GDSs are also called *Automata Networks* [41] and, if nodes have only two states, *Boolean Networks* [45]. While GDS has been primarily used for analysis problems, e.g., [12–15, 74] and proving or discovering features of different systems, e.g. [3, 5, 6, 31, 55, 72], we use it in Sec. 2.1 as a framework for building human behavior models.

After specifying the GDS, the equations for the behavior model (i.e. the local functions for the state transition model of the GDS) are presented. This includes methods to cluster game player behaviors so that the state transition model can be tailored to each cluster. A Frequentist method is presented in Sec. 2.2 and a Bayesian method is described in Sec. 2.3. These topics set the stage for the uncertainty visualization methods in Sec. 3.

2.1. Discrete dynamical system model formulation

The state transition diagram in Fig. 2 captures the permissible states and transitions of a player in a NGrAG. Given that a player is in some state $\sigma_\ell (\ell \in \{1, 2, \dots, q\})$ at time t , it can transition to another state $\sigma_m (m \in \{1, 2, \dots, q\})$ at time $(t + 1)$. The state $\sigma_\ell = (a_i, \mathbf{z})$ is a concatenation of player action a_i ($i \in \{1, 2, 3, 4\}$) represented in

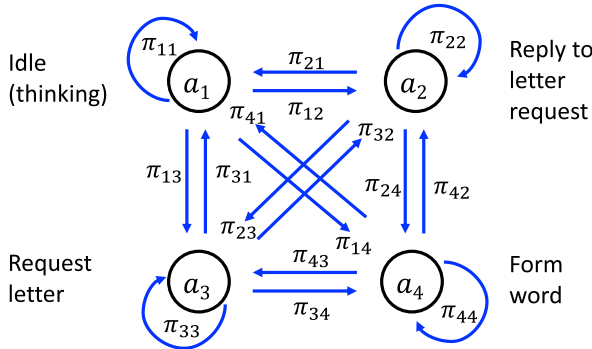


Fig. 2. Four-node state transition diagram for one player or agent in a NGrAG. Each $a_i (i \in \{1, 2, 3, 4\})$ is given in Fig. 1(a). In a graph $G(V, E)$ of a NGrAG, a player or agent $v_k \in V$ is always in one of the states $K = \{\sigma_1, \sigma_2, \dots, \sigma_q\}$, where $\sigma_\ell = (a_i, \mathbf{z})$. While there are only four a_i values, the vector $\mathbf{z}$ (see text), which can take thousands of values, is not displayed in this diagram for clarity. The transition probability π_{ij} refers to the probability of a player moving from his/her current state σ_ℓ which includes action a_i to a new state σ_m which includes action a_j in the next time step (1 s in our model).

Fig. 1(a) and a vector $\mathbf{z}$ of counts that will be explained in what follows. Each transition from σ_ℓ with action a_i to σ_m with action a_j is characterized by a probability π_{ij} . This diagram represents a (probabilistic) DDS. The system is discrete in time and in state. In particular, since the players and their interactions can be represented as a network, the DDS can be specialized to a graph dynamical system (GDS). Since $\mathbf{z}$ can take on thousands of values, we show only a_i in Fig. 2 to convey the essential ideas.

A *graph dynamical system* (GDS) [4, 65], denoted S , is a four-tuple $S = (G, K, F, W)$ where G is a (social) network $G(V, E)$ with vertex (or node or agent or player) set V and edge set E , where $n = |V|$ and $m = |E|$. An undirected edge between $v_k \in V$ and $v_\ell \in V$, denoted $\{v_k, v_\ell\} \in E$, means that v_k and v_ℓ interact and therefore can influence each other. The set K of *vertex states* is the set of admissible states of a node. The state s_k of an agent v_k is assigned a value from K at each time t . A *sequence of local functions* $F = (f_1, f_2, \dots, f_n)$, with $|F| = |V|$, contains a local function f_k for each $v_k \in V$. The local function f_k specifies how agent v_k updates its state s_k , and will be explained below. The *update scheme* W is the scheme for the order in which the f_k are invoked at each time. For this work, we assume a *synchronous* or *parallel* scheme whereby all agents update their states in parallel (i.e. simultaneously).

Other update schemes include *sequential*, where each node updates its state in a permutation order; *block-sequential*, which generalizes synchronous and sequential where each block of nodes invokes their local functions synchronously and blocks are arranged sequentially; and *unfair word orderings*, where some nodes do not execute their local functions as often as other nodes [41, 56, 65].

The *system state*, s , also called a *configuration*, with $s = (s_1, s_2, \dots, s_n)$, is the n -vector of all vertex states. Let N_k be the set of vertices in the closed neighborhood

of v_k (that is, N_k is the set containing v_k and all of v_k 's distance-1 neighbors in G). Let $s[v_k]$ be the sequence of states of the vertices in N_k ; the sequence is of length $d(v_k) + 1$, where $d(v_k)$ is the degree of v_k in G . Then the next state s'_k of v_k is computed with the local function f_k on the current states of the vertices in its closed neighborhood, i.e. $s'_k = f_k(s[v_k])$. Introducing time t as a superscript, then the state $s_k^{(t+1)}$ at time $t + 1$ is computed by f_k from the states $s^t[v_k]$ at time t thusly $s_k^{(t+1)} = f_k(s^t[v_k])$.

In the following development, the vertex state $\sigma_\ell = (a_i, \mathbf{z})$ is explained and the local function f_k for each $v_k \in V$ is developed from NGrAG data and takes the form of π_{ij} as suggested in Fig. 2.

2.2. Frequentist method of modeling state transition

A Frequentist approach for quantifying uncertainty has been developed in previous studies [47, 48] to incorporate the diverse behaviors of multiple players into the model as shown in Fig. 3. They defined two variables x_e and x_w to quantify each player's activity level. Number of *engagements* x_e equals the sum of the number of requests and the number of replies that a player sends, representing the player's interactions with neighbors in the game; x_w equals the number of words a player forms in a game, representing the player's ability to form words. According to the hypothesis testing results of these two variables from the game data, players were first divided into two groups: group 1 ($g = 1$) consisted of those with fewer than three neighbors, while group 2 ($g = 2$) comprised those with three or more neighbors.

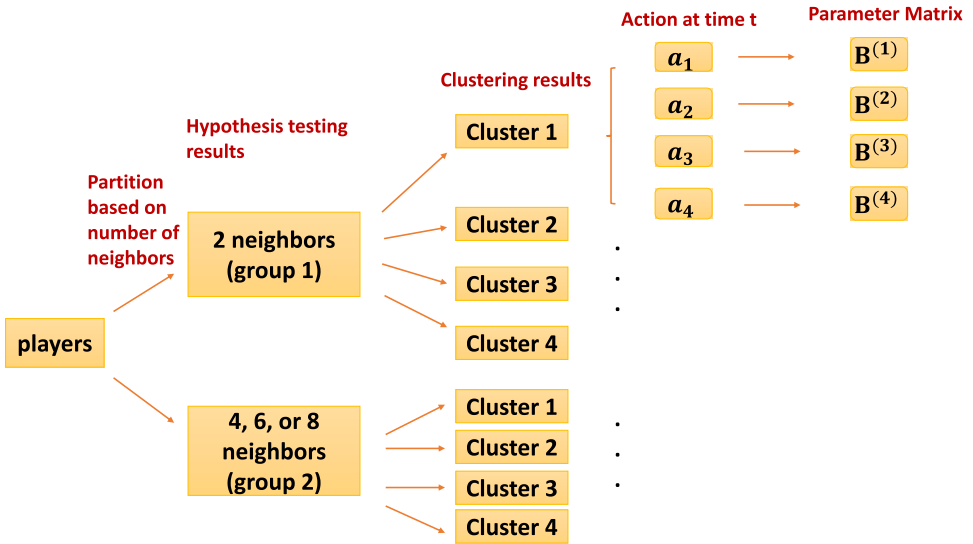


Fig. 3. Flowchart for partitioning NGrAG data to produce state-transition models. A model of the form of Eq. (2) is generated for each combination $[g, c]$ of group g and cluster c .

Subsequently, k -means clustering was employed to further classify the players in each group into four clusters c , with individuals in the same cluster exhibiting similar levels of activity in the game.

In order to predict a player's next action at time $t + 1$, four potential predictors are identified:

- (1) the count of letter requests that remain unanswered by the player at time t ($Z_B(t)$);
- (2) the number of letters available to the player for constructing words at time t ($Z_L(t)$);
- (3) the number of valid words that the player has formed up through time t ($Z_W(t)$);
- (4) the number of consecutive time steps that the player has repeated an action up through time t ($Z_C(t)$).

Combining these four predictors with a constant, we can define the predictor vector $\mathbf{z}(t) = (1, Z_B(t), Z_L(t), Z_W(t), Z_C(t))^T_{5 \times 1}$. The dependent variable is the transition probability

$$\pi_{ij} = Pr(X_{t+1} = a_j | X_t = a_i, \mathbf{z}(t)), \quad (1)$$

which is the conditional probability of a player selecting action a_j at time $t + 1$, given that the player took action a_i at time t and has predictor vector $\mathbf{z}(t)$.

Then within each cluster, per Fig. 3, for a given action a_i at time t , the player's next action at time $t + 1$ is modeled using multinomial logistic regression as follows, where π_{i1} is chosen as the reference level for other categories to compare

$$\log\left(\frac{\pi_{ij}}{\pi_{i1}}\right) = \mathbf{z}(t)^T \boldsymbol{\beta}_j^{(i)}, \quad j = 2, 3, 4, \quad (2)$$

where $\boldsymbol{\beta}_j^{(i)} = (\beta_{j1}^{(i)}, \dots, \beta_{j5}^{(i)})^T_{5 \times 1}$ is the coefficient parameter vector. The parameters can be expressed as a matrix $\mathbf{B}^{(i)} = (\boldsymbol{\beta}_2^{(i)}, \boldsymbol{\beta}_3^{(i)}, \boldsymbol{\beta}_4^{(i)})^T_{3 \times 5}$ for $i = 1, \dots, 4$. After determining the group g ($g = 1$ or 2) and performance cluster c ($c = 1, 2, 3$, or 4) by the game network structure and the activity level, respectively, a particular model based on Eq. (2), with parameter matrix $\mathbf{B}^{(i)}$, is assigned to a game player to predict the probabilities of next actions as illustrated in Fig. 2.

Note that depending on game conditions, the number of words a player has formed can easily be up to 15 or 20 (Z_W), the number of consecutive idle or thinking times steps can easily be up to 20 (i.e. 20 s) for Z_C , the number of letters that a player has in hand can be from three to seven (Z_L), and the number of letter requests unanswered can be in the range [3, 7] (for Z_B). Consequently, the number of values of $\mathbf{z}$ can be roughly 10^4 ($= 20 \times 20 \times 5 \times 5$). This is why $\mathbf{z}$ was not included in Fig. 2, focusing instead on the four actions.

2.3. Bayesian modeling method

In our previous work [59], a Bayesian framework of uncertainty quantification (UQ) was developed to address some shortcomings of traditional UQ methods [47, 48].

First, by using k-means clustering, the number of clusters has to be pre-specified, which is an issue without a definitive solution. Kulis and Jordan [57] proposed a Bayesian nonparametric model utilizing a Dirichlet Process (DP) prior, referred to as *DP-means clustering*. This approach is based on the Dirichlet Process Mixture (DPM) model under the Gaussian mixture distribution as shown in Fig. 4.

A Dirichlet Process (DP) generates a probability distribution G given the base distribution G_0 and a scale parameter α . Each ϕ_i ($i = 1, \dots, n$) is drawn independently from G and is taken as the mean of the Gaussian distribution that observations followed.

Based on this DPM model, Kulis and Jordan [57] developed a DP-means clustering algorithm such that if the shortest distance between a data point x_i and at least one of the existing cluster centers is less than λ , then the data point will be assigned to the closest cluster, which is the same as the conventional k-means clustering method. However, if all the distances are greater than λ , the algorithm will start a new cluster with x_i being the first point, as shown in Fig. 5. As a result, this approach can automatically partition the data without requiring pre-specifying the number of clusters, thus mitigating the problems caused by inaccurate pre-specification in k-means clustering. After standardizing the two variables x_e and x_w and applying the DP-means clustering with a specific penalty parameter $\lambda = 2.5$, we further partitioned the players in the same group into four clusters as in the procedures of Sec. 2.2.

Second, the Frequentist approach to fit a multinomial regression uses the maximum likelihood estimation (MLE) method [35] and relies on the asymptotic properties of MLE to conduct inferences. However, the asymptotic normal approximation

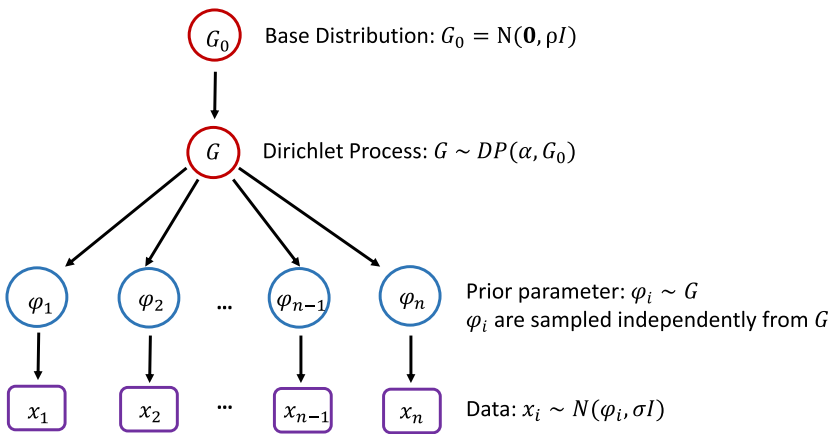


Fig. 4. DPM model under the Gaussian mixture distribution. The base distribution G_0 is assumed to be zero-mean Gaussian with covariance a diagonal matrix ρI , i.e. $\mathcal{N}(\mathbf{0}, \rho I)$. After drawing a distribution G from DP, ϕ_i is drawn independently from G and is taken as the mean of the Gaussian distribution that observations followed. This construction is an integral part of the DP-means clustering method.

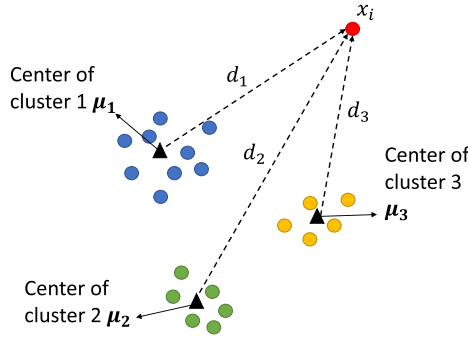


Fig. 5. DP-means clustering. A new cluster is formed whenever a point is further than distance λ away from every existing cluster center. In this example, x_i will be the first point in a new cluster if $d_i > \lambda$, $i = 1, 2, 3$. In this way, the clustering method determines the number and composition of clusters of players' behaviors.

of MLE is not effective when the sample size is small [42, 88]. In 1993, Albert and Chib [7] switched to the Bayesian approach and used Markov Chain Monte Carlo (MCMC) techniques to obtain exact inferences for the parameters, thus avoiding the limitations of MLE with small sample sizes.

The fundamental principle underlying Bayesian estimation is that even prior to examining any data, one already possesses certain prior information regarding the distribution from which it is derived. According to Bayes' theorem [16], the availability of a prior distribution of parameter matrix $P(\mathbf{B}^{(i)})$ allows the computation of the posterior distribution

$$P(\mathbf{B}^{(i)}|y_1, \dots, y_n) = \frac{P(y_1, \dots, y_n|\mathbf{B}^{(i)}) \times P(\mathbf{B}^{(i)})}{P(y_1, \dots, y_n)} \\ \propto P(y_1, \dots, y_n|\mathbf{B}^{(i)}) \times P(\mathbf{B}^{(i)}).$$

In our model, since the available prior information is weak, we choose Jeffreys' prior which is a non-informative prior that is invariant under the transformation of parameters [49]. This prior assigns equal weight to all possible values of the parameters, and therefore, will not bias the posterior estimates towards any particular value. We then quantify the uncertainty of parameters by conducting posterior inference. MCMC techniques are frequently employed to derive samples from the posterior distribution, which can be used for posterior inference, such as parameter estimation, uncertainty quantification, and prediction. MCMC methods are particularly useful when the posterior distribution cannot be computed analytically, or when the model contains a large number of parameters. The memorylessness property of MCMC [84] can alleviate the extreme value problem caused by data scarcity since every sample is only generated based on the previous one. These considerations and approaches lead to the new uncertainty visualization method in Sec. 3.

3. Uncertainty Visualization Methods

The proposed methods for uncertainty visualization are described in this section. The objective is to visualize the uncertainty within and across clusters, and to identify the diverse (i.e. worst, average, and best) behaviors of players in each cluster.

3.1. Frequentist uncertainty visualization

In the state transition model, if a player's most recent action is a_i , then the determination of his/her next action is performed through the mapping of the parameter matrix $\mathbf{B}^{(i)}$ and input vector $\mathbf{z}$ onto a probability vector $\boldsymbol{\pi}_i = (\pi_{i1}, \pi_{i2}, \pi_{i3}, \pi_{i4})^T$, following Eq. (2). With the aim of examining the diverse behaviors of players within the same cluster, we employ the asymptotic normality property of MLE to quantify the uncertainty for parameter matrix $\mathbf{B}^{(i)}$. Through this approach, a range of $\mathbf{B}^{(i)}$ matrices can be obtained by sampling from an asymptotic normal distribution, thereby representing the various behaviors of players. Without loss of generality, we omit the superscript i in parameter matrix $\mathbf{B}^{(i)}$ and transform it to the parameter vector $\boldsymbol{\beta} = (\boldsymbol{\beta}_2^{(i)T}, \boldsymbol{\beta}_3^{(i)T}, \boldsymbol{\beta}_4^{(i)T})_{15 \times 1}^T$. Action idling a_1 is treated as a reference group, thus $\boldsymbol{\beta}_1^{(i)}$ will not be included in the parameter vector. Subsequently, the asymptotic property of MLE [80] is utilized. This property states that as sample size increases, the maximum likelihood estimator $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ of parameter $\boldsymbol{\beta}$ converges to a multivariate normal distribution

$$\hat{\boldsymbol{\beta}}_{\text{MLE}} \xrightarrow{d} \text{MN}(\boldsymbol{\beta}, \boldsymbol{\Sigma} = I(\boldsymbol{\beta})^{-1}/n), \quad (3)$$

with mean equal to the true parameter value and variance equal to the inverse of the Fisher information matrix $I(\cdot)$ [35] over the number of observations in the cluster. The parameter $\boldsymbol{\beta}$ can be estimated by the MLE $\hat{\boldsymbol{\beta}}_{\text{MLE}}$ and covariance matrix $\boldsymbol{\Sigma}$ can be estimated by $\hat{\boldsymbol{\Sigma}} = I(\hat{\boldsymbol{\beta}}_{\text{MLE}})^{-1}/n$. Based on this asymptotic property, one can use the Wald statistic [85] to construct the confidence region $S_{\boldsymbol{\beta}}$, which is a region in parameter space of $\boldsymbol{\beta}$ that is likely to contain the true value of the parameter with a specified level of confidence $((1 - \alpha) \times 100\%)$:

$$\Pr(\hat{\boldsymbol{\beta}} \in S_{\boldsymbol{\beta}}) = (1 - \alpha) \times 100\%, \quad (4)$$

$$(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{\text{MLE}})^T \hat{\boldsymbol{\Sigma}}^{-1} (\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{\text{MLE}}) < \chi_p^2(1 - \alpha), \quad (5)$$

where $\chi_p^2(1 - \alpha)$ is the $(1 - \alpha)$ quantile of the Chi-squared distribution with degree of freedom p ($p = 15$ is the dimension of parameter vector $\boldsymbol{\beta}$) [63]. Given a $\hat{\boldsymbol{\beta}}$, a probability vector can be computed for each observation by Eq. (2). By averaging the probability vectors of all observations possessing the same initial action within a cluster, we can obtain the mean probability vector and associated standard error for that sampled $\hat{\boldsymbol{\beta}}$. By uniformly drawing a sequence of $\hat{\boldsymbol{\beta}}$ from the $(1 - \alpha) \times 100\%$ confidence region, a corresponding sequence of mean probability vectors and their

standard errors can be obtained. To effectively visualize the uncertainty among these mean probability vectors, a bubble plot is created, with the center of each bubble representing the mean probability vector for a given parameter matrix. The width of the bubble is defined as $2 \times \text{SE}(\bar{\pi}_4^r)$ and the height is defined as $2 \times \text{SE}(\bar{\pi}_1^r)$, where $\bar{\pi}_4^r$ and $\bar{\pi}_1^r$ are defined in Algorithm 1, and $\text{SE}(\cdot)$ denotes standard error.

A key advantage of this proposed method is that we can visually analyze the uncertainty among data. In the bubble plot, it is easy to find the best and the worst behaviors and view the heterogeneous behaviors within each cluster. The probability of forming words (π_4) in the probability vector represents the player's ability to form words in the game, while the probability of not being idle ($1 - \pi_1$) represents the player's activity level. Moreover, the size of the bubble can help us visually detect the variability among the observations. One can also quantitatively compare the activity

Algorithm 1. Frequentist Uncertainty Visualization Method

Input: A sequence of $\hat{\beta}_r$ ($r = 1, \dots, R$) sampled from the 95% confidence region of $\hat{\beta}$.

for r in $1 : R$ **do**

1. Based on $\hat{\beta}_r$, apply the n observations that possess the same initial action within a cluster to Eq. (2) to produce n probability vectors:

$$\hat{\pi}^{r,l} = (\hat{\pi}_1^{r,l}, \hat{\pi}_2^{r,l}, \hat{\pi}_3^{r,l}, \hat{\pi}_4^{r,l}), \quad l = 1, \dots, n.$$

2. Calculate the mean of these probability vectors and the associated standard error:

$$\bar{\pi}^r = \frac{1}{n} \sum_{l=1}^n \hat{\pi}^{r,l} = (\bar{\pi}_1^r, \bar{\pi}_2^r, \bar{\pi}_3^r, \bar{\pi}_4^r)^T,$$

$$\text{SE}(\bar{\pi}^r) = \frac{1}{n} \sqrt{\sum_{l=1}^n (\hat{\pi}^{r,l} - \bar{\pi}^r)^2} = (\text{SE}(\bar{\pi}_1^r), \text{SE}(\bar{\pi}_2^r), \text{SE}(\bar{\pi}_3^r), \text{SE}(\bar{\pi}_4^r))^T.$$

3. Draw a bubble, which is an ellipse centered at the mean probability $(\bar{\pi}_4^r, 1 - \bar{\pi}_1^r)$ with width $2 \times \text{SE}(\bar{\pi}_4^r)$ and height $2 \times \text{SE}(\bar{\pi}_1^r)$.

end

Output: A bubble plot where each bubble represents the mean probability of the next action for one sampled parameter vector $\hat{\beta}$.

Select the B_r matrix with the maximum $\bar{\pi}_1^r$ as the matrix of the worst behavior, and the one with the minimum $\bar{\pi}_1^r$ as the matrix of the best behavior. The B_r matrix of the average behavior is one that produces the mean of $\bar{\pi}_1^r$, $r = 1, \dots, R$.

ranges of clusters with players that have the same number of neighbors to further discover the differences between clusters within the same group ($g = 1$ or 2). Note that this visualized uncertainty quantification was not contained in the previous work [48].

Algorithm 1 summarizes the proposed Frequentist approach of uncertainty visualization within a cluster.

3.2. Bayesian uncertainty quantification

MCMC methods can be employed to conduct sampling from the exact posterior distribution. The Metropolis–Hastings (MH) [43, 63] and Gibbs sampling [37] algorithms are the two commonly used MCMC algorithms. By applying the MH algorithm, a sequence of parameter θ can be sampled from the target distribution $f(\theta)$, in which θ_{t+1} is generated by drawing a random variable θ^* from a proposal distribution $g(\theta|\theta_t)$ and accepting $\theta_{t+1} = \theta^*$ with probability $\alpha = \min\{1, \frac{f(\theta^*)/g(\theta^*|\theta_t)}{f(\theta_t)/g(\theta_t|\theta^*)}\}$. For this paper, a Gaussian distribution was selected as the proposal distribution, resulting in a random walk MH algorithm since $\mathcal{N}(\theta^*|\theta_t) = \mathcal{N}(\theta_t|\theta^*)$. Without loss of generality, we omit the superscript i in parameter matrix $\mathbf{B}^{(i)}$ and specify

$$g(\mathbf{B}|\mathbf{B}_{t-1}) = \mathcal{N}(\mathbf{B}_{t-1}, (B_0 + C^{-1})^{-1}),$$

where B_0 is the prior precision matrix, and C is the sample variance–covariance matrix of the MLEs. We present the proposed method for visualizing uncertainty within a cluster in the Bayesian approach using Algorithm 2.

A key advantage of our Bayesian approach is that it alleviates the extreme value problem caused by data scarcity in the previous model [48]. When the size of the training data in each category is unbalanced (e.g. 556 observations have final state idle while only 4 observations have final state reply (a_2) and 4 observations have final state request (a_3) in group $g = 1$ cluster $c = 2$ with initial state a_3), the asymptotic normal distribution of $\mathbf{B}$ would have a very large variance. Thus, the estimated parameter in $\mathbf{B}$ can be unexpectedly large and cause an extreme value in the probability vector $\boldsymbol{\pi}$ and an infinite loop in state transitions in the ABM. However, the memorylessness property of MCMC can avoid this problem since every sample is only generated based on the previous one. For this reason, the Bayesian approach avoids the extreme scenarios of players' actions.

4. Visualization of Heterogeneous Behaviors

This section investigates variations in behavior across different groups, differences in activity levels between clusters, and uncertainties within clusters, using both game data and the models presented in Secs. 2 and 3.

Under the Bayesian framework, we utilize the MH algorithm to sample 1000 $\mathbf{B}$ matrices after 1000 burn-in iterations, for each initial state in a cluster. In the

Algorithm 2. Bayesian Uncertainty Visualization Method

Input: Initialized $\mathbf{B}_0 = \mathbf{0}$.

for t in $1 : T$ **do**

1. Generate a sample $\mathbf{B}^*$ from proposal distribution $N(\mathbf{B}|\mathbf{B}_{t-1})$.
2. Calculate the acceptance ratio $\alpha = \min\{1, \frac{P(\mathbf{B}^*|y_1, \dots, y_n)}{P(\mathbf{B}_{t-1}|y_1, \dots, y_n)}\}$.
3. $\mathbf{B}_t = \begin{cases} \mathbf{B}^* & \text{with probability } \alpha \\ \mathbf{B}_{t-1} & \text{with probability } 1 - \alpha \end{cases}$.

end

We will get a sequence of T parameter matrices. After 1000 burn-in (throwing away some iterations at the beginning of an MCMC run), we can redefine the remaining matrices as $\mathbf{B}_r$. ($r = 1, \dots, R = T - 1000$).

for r in $1 : R$ **do**

1. Based on $\mathbf{B}_r$, apply the n observations that possess the same initial action within a cluster to Eq. (2) to produce n probability vectors.
2. Calculate the mean of these probability vectors and the associated standard error using equations in Algorithm 1.
3. Draw a bubble, which is an ellipse centered at the mean probability $(\bar{\pi}_4^r, 1 - \bar{\pi}_1^r)$ with width $2 \times \text{SE}(\bar{\pi}_4^r)$ and height $2 \times \text{SE}(\bar{\pi}_1^r)$.

end

Output: A bubble plot where each bubble represents the mean probability of the next action for one sampled parameter vector $\hat{\beta}$.

Select the $\mathbf{B}_r$ matrix with the maximum $\bar{\pi}_1^r$ as the matrix of the worst behavior, and the one with the minimum $\bar{\pi}_1^r$ as the matrix of the best behavior. The $\mathbf{B}_r$ matrix of the average behavior is one that produces the mean of $\bar{\pi}_1^r$, $r = 1, \dots, R$.

Frequentist approach, we draw 1000 samples uniformly from the 95% confidence region of the parameter matrix. The comparison between the Bayesian approach and the Frequentist approach is presented in what follows.

Figures 6(a) and 6(b) show the Frequentist and Bayesian bubble plots for group 2 cluster 3 with the initial state being idle (a_1), respectively. The ranges of values on x - and y -axes are different in order to zoom in on the data. Clearly, there exists uncertainty within the clusters. The samples that generate the best, worst, and average behaviors have been labeled on the plots. Additionally, the size of a bubble reflects the variability in the players and the color indicates the number of replications. Specifically, the darker bubbles represent a greater number of samples having that particular transition probability. In the Bayesian plot, samples are more gathered around the maximize a posteriori (MAP) estimation (the blue bubble), while

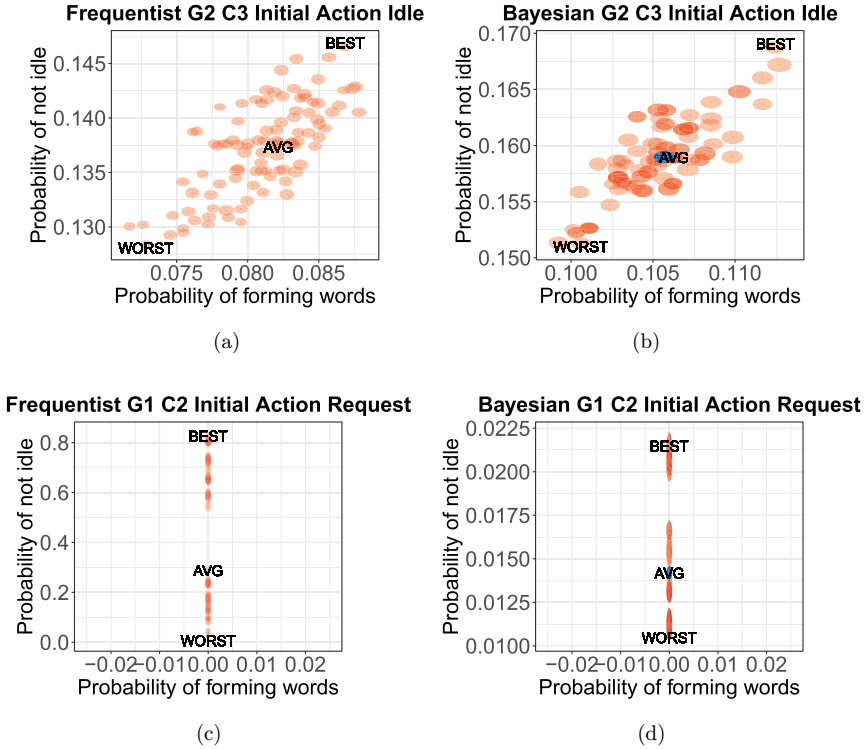


Fig. 6. (Color online) (a) Bubble plot of group 2, cluster 3, where the initial state is idle (a_1) for the Frequentist approach. (b) Bubble plot of group 2, cluster 3, where the initial state is idle (a_1) for the Bayesian approach. (c) Bubble plot of group 1, cluster 2, where the initial state is request (a_3) for the Frequentist approach. (d) Bubble plot of group 1, cluster 2, where the initial state is request (a_3) for the Bayesian approach. The probabilities of forming words are 0 for (c) and (d). Note that (a) and (b) have different y -scales, as do (c) and (d). Each bubble is an ellipse centered at the mean probability ($\bar{\pi}_4^r, 1 - \bar{\pi}_1^r$) with width $2 \times \text{SE}(\bar{\pi}_4^r)$ and height $2 \times \text{SE}(\bar{\pi}_1^r)$, where $\text{SE}(\bar{\pi}_4^r)$ and $\text{SE}(\bar{\pi}_1^r)$ are standard errors of mean to-word probability and mean to-idle probability, respectively. The blue bubbles in Bayesian plots are the maximize a posteriori (MAP) estimations. The bubbles indicating the worst, average, and best performance are marked. The value in these plots is that for a given approach (Frequentist, Bayesian) and a specified action a_i , one sees immediately two of the major performance parameters of the model: probability of forming words (on which game reward is based) and probability of taking some action, at the next time step. Also, worst, average, and best behaviors are visually comprehended, and uncertainty is quantified. Finally, one can compare across approaches and actions.

Frequentist approach bubbles are more scattered and have fewer replications. This is because in the Bayesian method, the MH sampling algorithm uses the previous sample as the starting point for the next sample. As a result, the samples are correlated and tend to be more similar to each other than those from the Frequentist method, where samples are drawn uniformly and independently.

In the same way, Frequentist and Bayesian bubble plots for group 1 cluster 2 with the initial state being request (a_3) are presented in Figs. 6(c) and 6(d), respectively. The scales of the y -axis are different in Figs. 6(c) and 6(d). The probability of being

non-idle ranges from 0.006 to 0.83 in the Frequentist bubble plot, while it only ranges from 0.01 to 0.022 in the Bayesian bubble plot. This scale issue is unusual and can be attributed to the small sample size for this initial action in this cluster. Among 536 observations that have initial action request (a_3) in group 1 cluster 2, 530 of them transit to idle (a_1), while only three observations choose to reply (a_2) and three choose to request (a_4) as the next action. When the sample size is small, the MLE may have a large variance, and the distribution may not be well approximated by a normal distribution, which is assumed in many of the asymptotic results for MLE. However, the Bayesian approach can alleviate this data scarcity problem since it does not rely on asymptotic properties. The Bayesian approach is more flexible and can provide more accurate estimates of the parameters even with a small sample size. Therefore, the Bayesian bubble plot may be more reliable in this particular case. These data were chosen to highlight differences in the two approaches.

Figure 7 presents bubble plots of all clusters in one group. In the first row of plots, where the initial state is idle (a_1), Figs. 7(a) and 7(b) show that four clusters in group 2 are well separated and the activity level is ascending, supporting the rationality of clustering players by behavior. It is also seen that players in clusters 3 and 4 in the Bayesian framework have higher probabilities of forming words and being non-idle than those using the Frequentist approach, so they are more active, indicating a better performance of DP-means clustering than k -means clustering. In the second row of plots, where the initial state is request (a_3) in group 1, all players' next action is idle (a_1) in cluster 1. Therefore, the probability of being non-idle in this cluster is 0, and it is not shown in the figures. In the other three clusters (clusters 2, 3, and 4), none of the players' next action is forming words, resulting in probabilities of forming words being 0. To avoid overlapping, we assign a different value of probabilities of forming words for bubbles in clusters 3 and 4 and compare their probabilities of being non-idle. Notably, Figs. 7(c) and 7(d) have different scales on the y -axis, demonstrating that the Bayesian approach can alleviate data scarcity problems, as claimed earlier. Similar patterns can be observed in Figs. 7(e) and 7(f).

5. Agent-Based Simulations of Networked Anagram Games and Results

In this section, we describe ABMs and the simulation process used to model the NGrAG. The uncertainty deduced in Sec. 4, derived from our experimental data, is utilized as inputs that guide the dynamic behavior of agents in our ABM simulation in this section. In this way, the variability obtained from the real-world data can be incorporated into the ABM simulations, thus ensuring that the behaviors of the simulated agents can accurately reflect the observed real-world complexities. For particular input conditions and models, we provide results for individual agents (also referred to as nodes or players) as well as all agents, under a wide range of conditions.

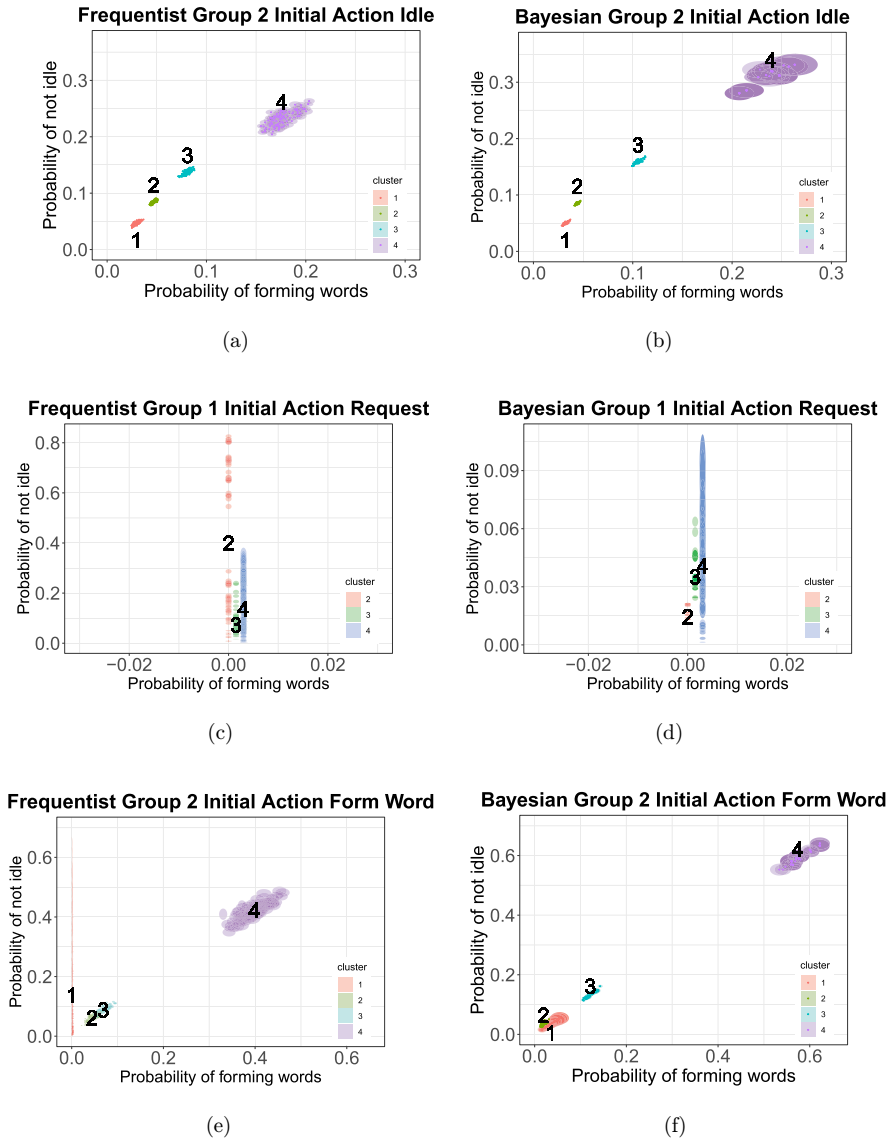


Fig. 7. (a) and (b) Bubble plots of group 2 where initial action is idle (a_1) for Frequentist approach and Bayesian approach, respectively. (c) and (d) Bubble plots of group 1 where initial action is request (a_3) for Frequentist approach and Bayesian approach, respectively. Note that in group 1 cluster 1 where initial action is request (a_3), all the players' next action is idle (a_1), so in this cluster, the probability of being non-idle is 0 and is not shown in the figures. In the other three clusters (clusters 2, 3, and 4), none of the players' next action is forming words, so the probabilities of forming words are 0. We assign a different value for bubbles in clusters 3 and 4 to avoid overlapping. (e) and (f) Bubble plots of group 2 where initial action is forming words (a_4) for Frequentist approach and Bayesian approach, respectively. While Fig. 6 provided these data for one group and cluster combination ($[g, c]$), this plot shows that data across all clusters can be compared.

5.1. Simulation process and computations

First, the initial conditions of a simulation are described, followed by a description of the simulation steps. Then the models of agent behaviors in the game are presented. Key parameters are summarized in Table 1.

The network $G(V, E)$ of Fig. 8 represents the possible interactions among the seven game players. It contains five players with two neighbors each (degree $d = 2$) and two agents with three neighbors each ($d = 3$). Each player is provided four initial letters, and the letters are purposely specified to enable players to form words, e.g. one player v_2 is given letters $\{i, l, m, n\}$ and neighboring players are given complementary letters, e.g. v_3 is assigned letters $\{o, p, r, s\}$. In this way, differences in models are highlighted (e.g. consider the opposite extreme where players are only given letters like x and z and no vowels — then agents cannot form words irrespective

Table 1. Summary of parameters and their values used in simulations of NGrAGs.

Parameter	Description
Networks $G(V, E)$.	The graph of Fig. 8.
Number n_l of owned letters.	The number of owned letters initially assigned to a player.
Initial letters L_k^{init} .	The set of initial n_l letters assigned to a player v_k .
Word corpus C_W .	The corpus of 5000 words available at www.wordfrequency.info . According to the website, these are the 5000 most frequently used words in English.
Player actions a .	The set A of actions a is given in Fig. 1: think or idle a_1 , reply to request with a letter a_2 , request a letter from a neighbor a_3 , and form a word a_4 .
Duration of NGrAG t_g .	NGrAG duration is fixed at $t_g = 300$ s.
Group, g .	There are two groups: $g = 1$ corresponds to nodes with degree $d \leq 2$ in the game network and $g = 2$ corresponds to nodes with degree $d \geq 3$.
Cluster, c .	For each group g , there are four clusters (c): $c = 1$ through 4.
Group-cluster $[g, c]$.	The group-cluster pair $[g, c]$ determines the behavior regime for each node.
Behavior classes B .	There are three behavior classes for each $[g, c]$ pair: the worst class B_W , the mean or average class B_A , and the best class B_B . These three categories correspond to the least, average, and greatest activity in terms of performing actions in a game.
Model approach R .	There are two modeling approaches: a Frequentist approach R_F and a Bayesian approach R_B . See Sec. 3. Each R has a separate model for each combination of $[g, c, B]$.
Game player behavior models M .	Each player in a NGrAG is assigned a behavior model M , which consists of the quadruple $M = [g, c, B, R]$.
Number of iterations n_{iters} .	Each simulation is composed of $n_{\text{iters}} = 100$ individual dynamics instances, where each instance starts from time $t = 0$, with initial conditions reset, and then the dynamics of the system are executed for t_g discrete time steps.

Notes: The first section contains variables that are physical entities that map directly to a NGrAG. The second section contains model parameters that prescribe node (i.e. player, agent) behaviors. The third section contains the simulation parameter. Sections are delineated by three horizontal lines.

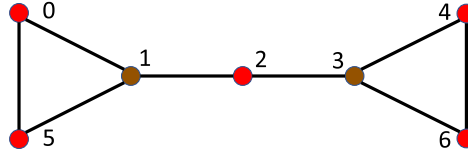


Fig. 8. (Color online) Seven node (agent) game network on which simulations are run. Red (respectively, brown) nodes are low (respectively, high) degree nodes of degree $d = 2$ (resp., $d = 3$). Therefore, red (respectively, brown) nodes are in group $g = 1$ (respectively, $g = 2$).

of behavior model so there is no way to contrast model results). Each agent may request all of its neighbors' *initial* letters (but not letters that neighbors receive from other players), as in the experiments.

One simulation is composed of 100 iterations or runs. Each iteration models one NGrAG, from time $t = 0$ through $t = 300$ s, as described in Sec. 1.1 and Fig. 1. An iteration uses discrete time steps, i.e. $t \in \mathbb{N}$ where $t \in [0, 300]$, and computations are performed at 1-s steps. Conditions at $t = 0$ s are the initial conditions above. At each $t \in (1, 2, \dots, 299, 300)$, each player $v_k \in V$ (simultaneously) computes the probability of performing each of the four actions in A . That is, given the last action a_i ($i \in \{1, 2, 3, 4\}$) and $\mathbf{z}$ for v_k at time t , the four π_{ij} ($j \in \{1, 2, 3, 4\}$) are computed using Eq. (2). The specific action is chosen based on a Bernoulli trial. If the selected action (i.e. form word a_4 , request a letter a_3 , or reply to a request a_2) is not possible, then the action for that agent and time step is changed to idle a_1 . This latter condition can arise in multiple ways, e.g. a player may choose to request a letter from a neighbor, but all letters have already been requested; a player may choose to reply to a neighbor's letter request, but may have already replied to all requests made of it thus far. Our discrete 1-s time step interval is justified by the fact that players do not take successive actions among request letter, reply to letter request, and form word within 1 s in the online experiments. Indeed, actions at time steps are mostly idle or thinking.

The difference among iterations within a simulation is the stochasticity of choosing an action for each agent at each time step via the Bernoulli trial, which over time, can lead to different action histories of agents. The initial conditions for each of the 100 iterations within a simulation are identical. Consequently, when we refer to "average" results below, we mean time point-wise averages across the 100 iterations, unless specified otherwise.

The remaining specifications for a simulation is the behavior model assigned to each agent. A behavior model M is given by the 4-tuple $M = [g, c, B, R]$. The first two parameters are the group g and cluster c assigned to an agent; these are described in Sec. 2. The five agents with $d = 2$ in Fig. 8 are in group 1 and the two agents with $d = 3$ are in group 2; the game network fixes the group of an agent. The cluster is a free input parameter; there are four clusters per group, and an agent is generally more active (i.e. forms more words, requests more letters, and replies to more letter requests) as cluster number increases from 1 to 4 (e.g. see Fig. 7). We examine all

combinations $[g, c]$ in these simulations. For each $[g, c]$ pair, there are three possible behavior classes B that are also free parameters: worst, average, and best. These three classes denote a further partitioning of each cluster into, respectively, a category of behavior that is least active B_W (in terms of taking actions request letter, reply to request, and form word), of average activity B_A , and most active B_B . See Table 1. We also evaluate all three classes in these simulations. In each simulation reported on herein, all agents are assigned the same class. Finally, the modeling approach R is either Bayesian R_B or Frequentist R_F . Thus, the behavior models of ABMs are designed and constructed from the models and results of Sec. 2 through 4.

5.2. Simulation results

In the results sections in what follows, several topics are explored: (i) variability across iterations of one simulation; (ii) variability across heterogeneous modeling classes worst, average, and best, both for individual iterations and for averages across all iterations of a simulation; (iii) action probabilities as explanations for actions taken by agents; (iv) variability in average action histories for agents with nominally the same behavior models; (v) contrasting the heterogeneity and variability in numbers of words formed in games generated by the Frequentist and Bayesian approaches; and (vi) neighbor interaction effects. We also demonstrate the utility of uncertainty visualizations in Sec. 4 for explaining simulation results. In the interest of space, and because the R_F and R_B modeling approaches produce qualitatively similar results, we focus mainly on the Bayesian approach results, i.e. the model $M = [g, c, B, R_B]$, although some results are provided for $[g, c, B, R_F]$.

5.2.1. Effect of model stochasticity and variability in results

Figure 9 provides three plots of counts of actions for agent 3 in a simulation where degree-2 and degree-3 agents are assigned, respectively, $[g, c] = [1, 3]$ and $[2, 3]$. All

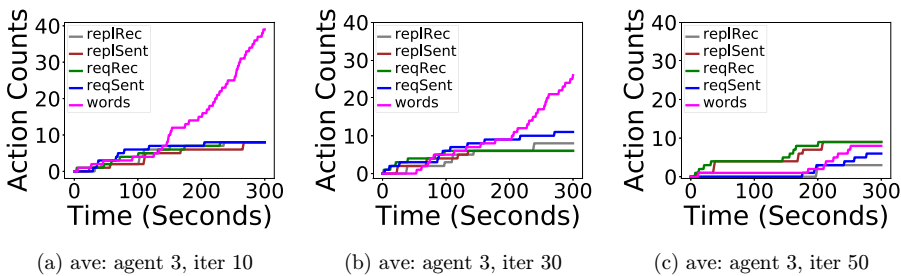


Fig. 9. Representative results for the Bayesian R_B model of anagram simulations with seven players. Results are for agent 3. Agents 1 and 3 with $d = 3$ are in $[g, c] = [2, 3]$ and the other agents with $d = 2$ are in $[g, c] = [1, 3]$. Within these clusters c , each agent is assigned the average behavior B_A . The plots are action counts for agent 3 for three of the 100 iterations or runs: (a) iteration 10, (b) iteration 30, and (c) iteration 50. Numbers of actions change across these three iterations and indeed across all iterations. All 100 iterations use the same initial conditions, so differences in results are manifestations of model stochasticity. This stochasticity in the models can be viewed in Figs. 6 and 7.

agents use B_A (the average behavior class) within their respective cluster 3. The actions are replies received (replRec), a passive action; replies sent (replSent), an active action a_2 ; requests received (reqRec), a passive action; requests sent (reqSent), an active action a_3 ; and words formed, an active action a_4 . All 100 iterations use the same properties and initial conditions, so that the differences in results among the iterations are solely due to stochasticity. Results from iterations 10, 30, and 50 are provided.

The most apparent difference among the three iterations is that the number of words formed varies from 7 (iteration 50) to 39 (iteration 10). However, requests sent (reqSent) also vary, from 6 to 10 (with three neighbors and four initial letters per agent, the maximum number of requests is 12). Iteration 50 has the least requests sent, but replies to more letter requests from its neighbors. In iteration 30, the number of requests sent is greater than the number of replies sent, and in iteration 10, they are equal. These results give a sense of the variability in results, for fixed conditions.

5.2.2. Variability across modeling class B for different groups and clusters

Figure 10 contains results for agent 5 of Fig. 8. All agents use the Bayesian R_B approach for modeling, and degree-2 agents use cluster $c = 1$ while degree-3 agents use $c = 3$. Results from multiple simulations are shown, where the classes are, from left to right B_W , B_A , and B_B (respectively, the worst, average, and best performing classes for the respective cluster). The first row of results provides simulation data for one iteration (iteration 51), while the second row provides time point-wise averages for the same actions (see legends) over all 100 iterations.

There are several interesting points. First, these are the first heterogeneous results because we are contrasting all three behavior classes. Second, the top row demonstrates that individual iterations for the worst (i.e. least active) class of behavior can show greater activity than selected iterations for the best (i.e. most active) class (formation of 10 words in Fig. 10(a), versus one word in Fig. 10(c)). However, the average results in the second row for the same three classes indicate that the number of words formed does increase in traversing worst to best classes of behavior, as one would expect. The increase in average numbers of words from worst to best is from 4.0 to 6.9, a 72% increase.

Figure 11 provides results for agent 3 (a degree-3 agent) for the same simulations reported for Fig. 10 for a degree-2 node. The collection of plots is also the same as those in the previous figure. First, a quick comparison of the two figures indicates that there is much more activity (in terms of numbers of actions) for this degree-3 node. While the degree plays a role (e.g. more neighbors to request letters from and reply to with requested letters, and more letters to form words), another key difference is the cluster. The degree-2 agent 5 in Fig. 10 has assigned cluster $c = 1$ from group $g = 1$, whereas here, agent 3 is assigned $[g, c] = [2, 3]$. From Fig. 7, it is observed that cluster 3 is generally much more active than cluster 1. This highlights the utility of uncertainty visualization to explain simulation results.

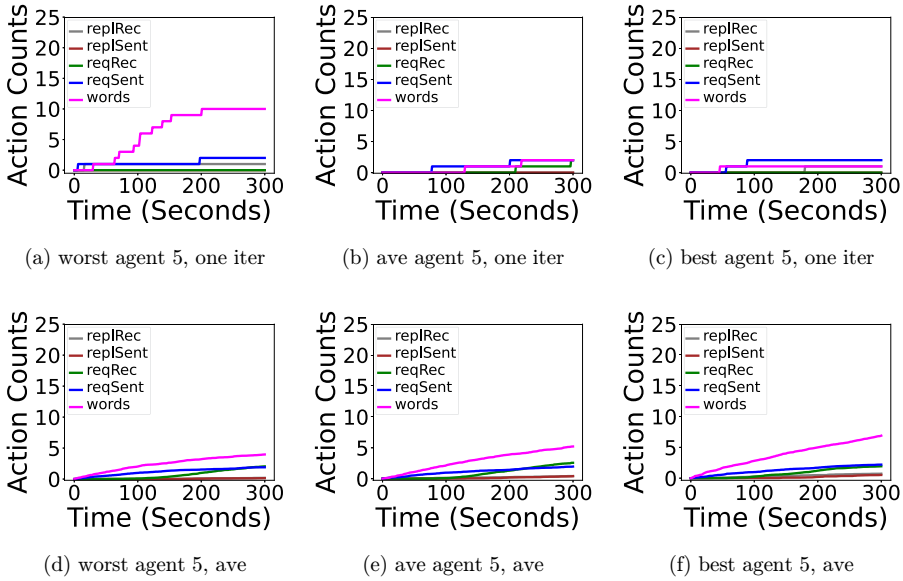


Fig. 10. Bayesian R_B model results of anagram simulations with seven players, focusing on the results for player 5 (a degree-2 agent). All players have behaviors assigned based on $[1, 1]$ for degree-2 agents and $[2, 3]$ for degree-3 agents (agent 5 is a degree-2 player). Worst performance model B_W (left column): (a) worst action histories for agent 5 in one of 100 iterations, and (d) average of worst action histories for agent 5, averaged over 100 iterations. Average performance model B_A (middle column): (b) average action histories for agent 5 in one of 100 iterations, and (e) average of average action histories for agent 5 averaged over 100 iterations. Best performance model B_B (right column): (c) best action histories for agent 5 in one of 100 iterations, and (f) average of best action histories for agent 5, averaged over 100 iterations. The main point is that when averaged over all 100 iterations, player performance (e.g. in terms of numbers of words formed) increases in going from worst to average to best *average* behavior. But within each behavior class, individual iterations can vary: the performance in (a) is better than that in (c).

Second, the first row of plots in Fig. 11 shows again that the number of formed words (“words” in the legend) does not increase monotonically in going from worst to best classes, just as in Fig. 10 — for these particular iterations. But also, the character of these curves changes: in Fig. 11(a), agent 3 forms all of its words between roughly 120 and 180 s; in Fig. 11(c), agent 3 forms most of its words before 100 s and after 200 s; and in Fig. 11(b), words are essentially formed after 100 s. The requests sent histories (reqSent in legends) are also different, even though the number of requests is the same over the entire game. In particular, the best class requests most letters late in the game (Fig. 11(c)), the average class makes requests comparatively early in the game (Fig. 11(b)), and the worst model makes requests at a more consistent pace (Fig. 11(a)). While these are individual iterations, the results do highlight the variability possible in simulation results.

The averaged results over the 100 iterations in Fig. 11(d) through Fig. 11(f) show the expected trends in numbers of words formed: they increase as the model class becomes more active. Average numbers of words formed increases, in going from

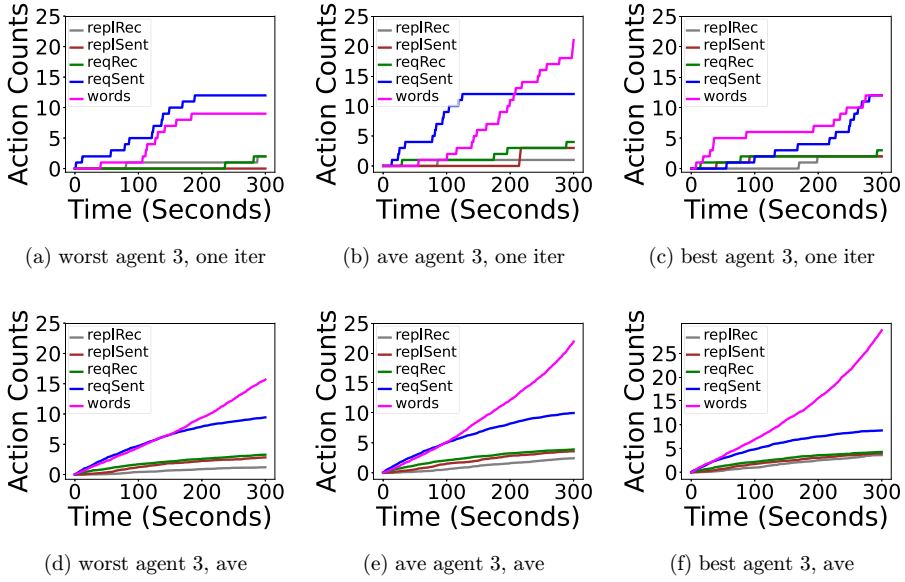


Fig. 11. Bayesian model results of anagram simulations with seven players, focusing on the results for player 3. All players have behaviors assigned based on $[1, 1]$ for degree-2 agents and $[2, 3]$ for degree-3 agents (agent 3 is a degree-3 player). Worst performance model B_W (left column): (a) worst action histories for agent 3 in one of 100 iterations, and (d) average of worst action histories for agent 3, averaged over 100 iterations. Average performance model B_A (middle column): (b) average action histories for agent 3 in one of 100 iterations, and (e) average of average action histories for agent 3 averaged over 100 iterations. Best performance model B_B (right column): (c) best action histories for agent 3 in one of 100 iterations, and (f) average of best action histories for agent 3, averaged over 100 iterations. A main point is the same as that in Fig. 10. Moreover, since node 3's behavior is from cluster 3, the counts of activities are greater than those in the previous figure.

worst B_W to best B_B behavior classes, from 15.7 to 29.9 words, a 90% increase in forming words. This demonstrates the utility of refining the behavior of $[g, c] = [2, 3]$ into these classes. However, this increase in numbers of words formed is accompanied, for the best class, by a slight decrease in numbers of letters requested.

5.2.3. Probability histories as explanations for action histories

Figure 12 provides average probability histories over the 100 iterations for agents 5 and 3, to help explain the results for agent 5 in Fig. 10(d) through Fig. 10(f), and for agent 3 in Fig. 11(d) through Fig. 11(f), respectively. Like the previous figures, this figure shows worst, average, and best classes of behavior in moving from left to right.

In all plots, the probabilities of idle (action a_1) are greatest. The probabilities of action forming words (action a_4) are most often the second greatest in magnitude, followed by requests (action a_3) and then replies (action a_2). Although it is slightly harder to discern for agent 5, because its cluster $c = 1$ is generally the least active of all clusters, the probabilities for forming words π_{i4} , for requesting letters π_{i3} , and for

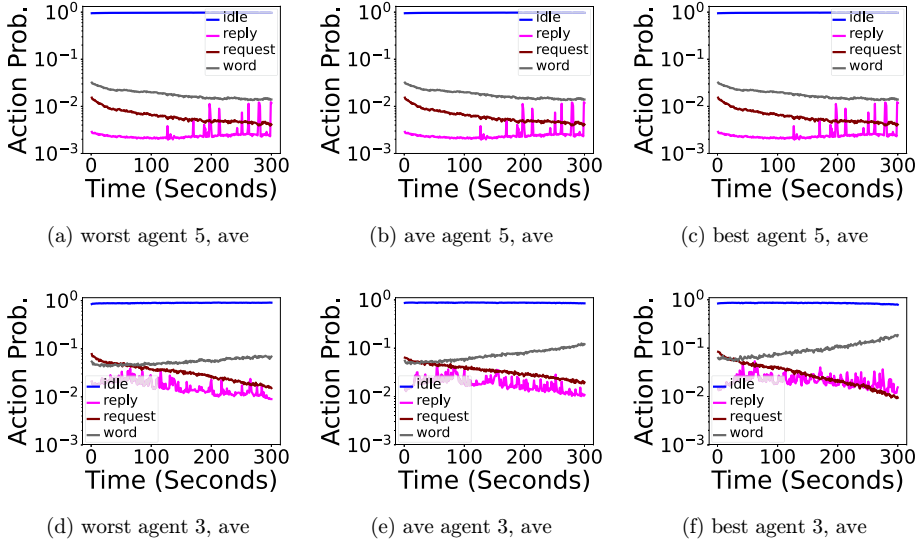


Fig. 12. Bayesian model R_B results of anagram simulations with seven players, focusing on the probability results for agent 5 (top row) and agent 3 (bottom row). All players have behaviors assigned based on $[g, c] = [1, 1]$ for degree-2 agents (agent 5 is a degree-2 player) and $[2, 3]$ for degree-3 agents (agent 3 is a degree-3 player). All probabilities are the time point-wise averages over the 100 runs of a simulation. Worst performance model B_W (left column): worst probability histories averaged over 100 iterations, (a) for agent 5, and (d) for agent 3. Average performance model B_A (middle column): average probability histories averaged over 100 iterations, (b) for agent 5, and (e) for agent 3. Best performance model B_B (right column): best probability histories averaged over 100 iterations, (c) for agent 5, and (f) for agent 3. Probabilities are greater for agent 3 (cluster 3) than for agent 5 (cluster 1), per Fig. 7, and these greater probabilities give rise to greater activity for agent 3 (compare Fig. 11 versus Fig. 10). The uncertainty visualizations help interpret and explain simulation results.

replying to letter requests π_{i2} do increase slightly in going from B_W to B_B . These trends are easier to see for agent 3 and its $c = 3$ cluster because this cluster has greater probabilities of actions (again see Fig. 7). The probabilities for forming words, in particular, show increasing values with time for $[2, 3]$, while those for $[1, 1]$ decrease (asymptotically) in time. The increased probability values π_{i2} , π_{i2} , and π_{i4} for agent 3, compared to those for agent 5, generate more actions for agent 3, as has been observed.

5.2.4. Heterogeneity of agent action histories for the same behavior models

Figure 13 contains action histories for agents 4 (top row) and 1 (bottom row). The point is to compare this top row of plots with the second row of plots in Fig. 10, as both are degree-2 nodes in the same simulations. It is observed that almost all of the actions for agent 4 are greater than those for node 5, across the three behavior classes B_W , B_A , and B_B . On the other hand, comparing the bottom row of plots in Fig. 11 with the bottom row of plots in Fig. 13, it is observed that the action histories for

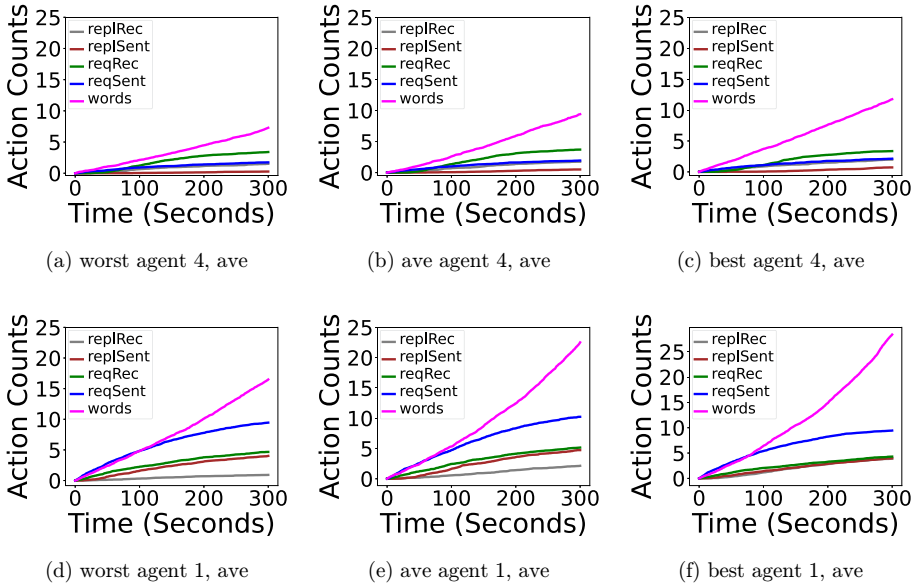


Fig. 13. Bayesian model R_B average results over 100 iterations of anagram simulations with seven players, focusing on the results for players 4 (degree-2 agent) and 1 (degree-3 player). All players have behaviors assigned based on $[1, 1]$ for degree-2 agents and $[2, 3]$ for degree-3 agents. Agent 4 (top row): (a) worst B_W , (b) average B_A , and (c) best B_B behavior classes. Agent 1 (bottom row): (d) worst, (e) average, and (f) best behavior classes. The activity counts for agent 4 are greater than those for agent 5 (both degree-2 agents); compare with Fig. 10. The activity counts for agent 1 are similar to those for agent 3 (both degree-3 agents); compare with Fig. 11. Results show that agents with the same game player behavior model M may produce the same action counts, but need not.

nodes 3 and 1 (both degree-3 agents) are quite similar. The conclusion is that nodes with the same models M may exhibit the same behavior, but they need not.

5.2.5. Heterogeneity of words formed by agents for all group and cluster combinations

Figure 14 uses the same network, the average behavior class B_A , and the Bayesian modeling approach R_B , for all combinations of $[g, c]$, i.e. models $M = [g, c, B_A, R_B]$. Since each group g can be assigned one of four clusters c , the result is 16 combinations of conditions. Each row fixes the group $g = 2$ cluster, and each column fixes the $g = 1$ cluster. In each plot, the number of words each agent generates in a simulation of 100 iterations is given as a box plot.

Data in Fig. 7 illustrate that as cluster number increases, the activity of agents increases. Hence, we would expect the box plots to move up (in increasing y -axis value) as we move left-to-right (increasing $g = 1$ cluster numbers) and as we move top-to-bottom (increasing $g = 2$ cluster numbers). This is indeed the case. Also, the rise in numbers of words formed is much greater for the $g = 2$ (i.e. degree-3) agents 1 and 3 than for the degree-2 agents.

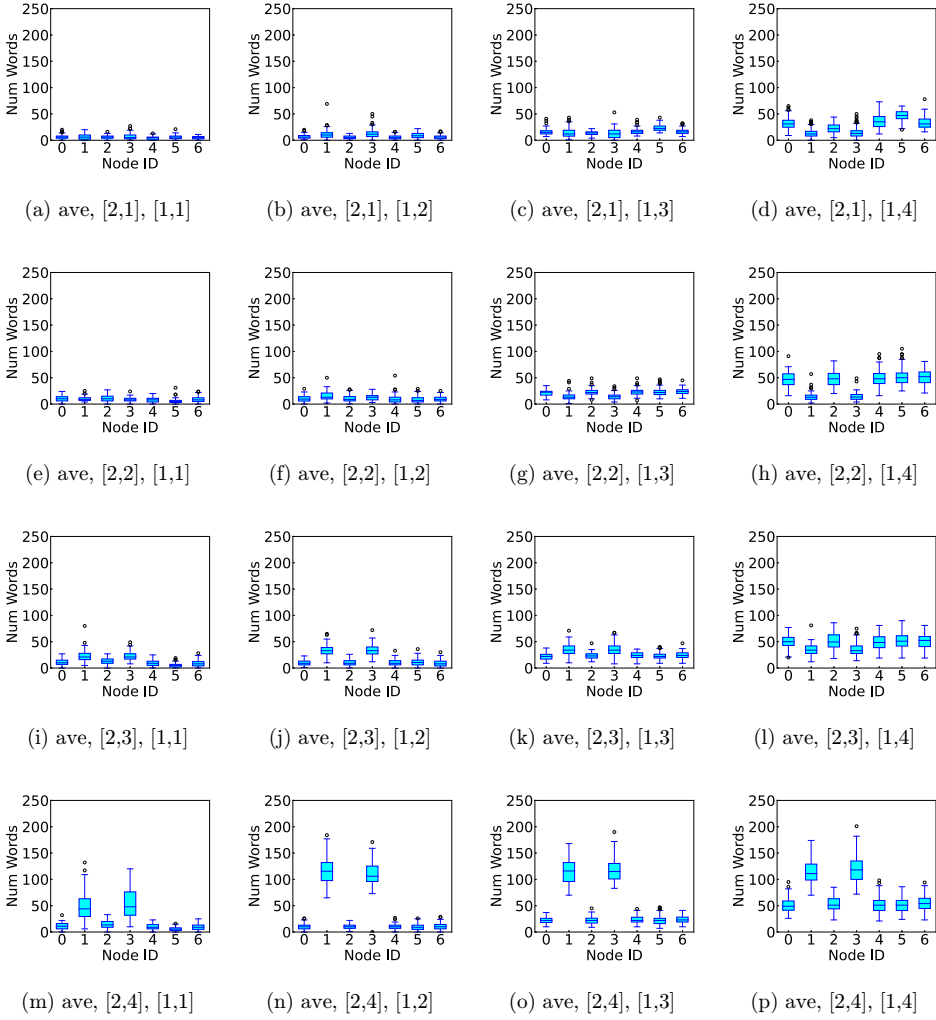


Fig. 14. Bayesian average class model $M = [g, c, B_A, R_B]$ model results over 100 iterations of anagram simulations with seven players. Each box in each plot represents the final number of words formed for each agent at the end of the game (i.e. at $t = 300$ s). Each plot represents different combinations of cluster (i.e. behavior) assignment for high and low degree nodes. For the $[g, c]$ combination under each plot, $[2, c]$ represents high degree nodes and $[1, c]$ represents low degree nodes. For each group, as cluster c increases, the number of words formed over the entire game generally increases. Changes in word counts across clusters, per group, are explained by uncertainty visualizations such as those in Fig. 7.

Figure 15 provides analogous data for the Frequentist model $M = [g, c, B_A, R_F]$. Comparing corresponding plots of the last two figures, it is observed that the Bayesian approach model produces more words than does the Frequentist model. This is suggested by Figs. 6(a) and 6(b), where the probabilities on both axes (indicating activity) are greater for the Bayesian approach

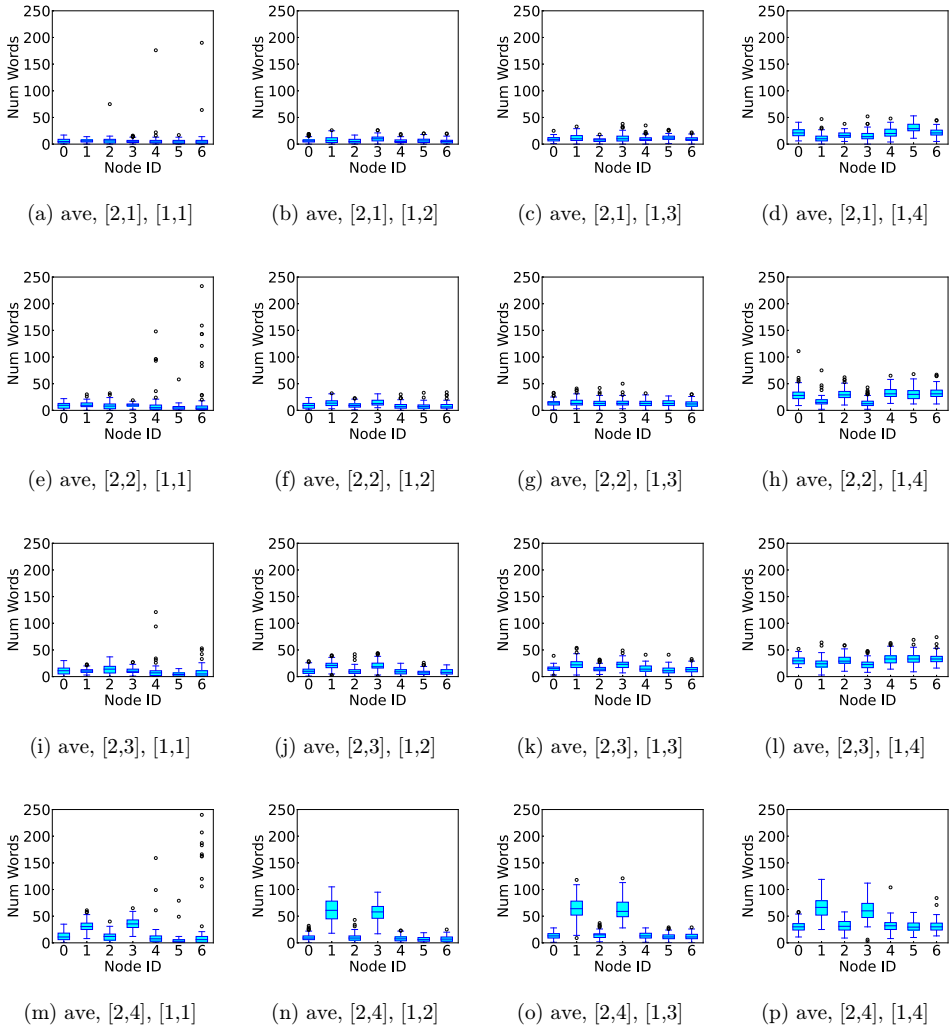


Fig. 15. Frequentist average class model $M = [g, c, B_A, R_F]$ results over 100 iterations of anagram simulations with seven players. Each box in each plot represents the final number of words formed for each agent at the end of the game (i.e. at $t = 300$ s). Each plot represents different combinations of cluster (i.e. behavior) assignment for high and low degree nodes. For the $[g, c]$ combination under each plot, $[2, c]$ represents high degree nodes and $[1, c]$ represents low degree nodes. For each group, as cluster c increases, the number of words formed over the entire game generally increases. Comparing the Frequentist and Bayesian data in Fig. 7 explains why the cluster 4 word counts for the Bayesian model in Fig. 14 are greater than those in this figure for the Frequentist model.

than are those for the Frequentist model, and by Fig. 7, where in most cases, the probabilities of activities are greater for the Bayesian approach. This is another example of the utility of uncertainty visualization in reasoning about simulation results.

5.2.6. Neighborhood effects

The edges in a game network represent interactions between pairs of players. The issue here is whether simulation results produce interaction effects. In Fig. 14, interaction effects are observed for the Bayesian approach. In going from Figs. 14(i) to 14(j), the group 1 cluster for low degree (i.e. degree-2) agents increases from 1 to 2. The high degree (i.e. degree-3) agent model is constant. Yet, it is observed that the numbers of words for the $g = 2$ agents (agents 1 and 3) increase in going from the first plot to the second one. That is, changes in the models of neighbors of agents 1 and 3 change the behaviors of agents 1 and 3. This same effect, but more pronounced, is observed in Figs. 14(m) and 14(n). Conversely, changing the $g = 2$ cluster from 1 to 2 (for agents 1 and 3) in Figs. 14(d) and 14(h) causes an increase in numbers of words formed in $g = 1$ agents, even though their model is fixed in the two simulations. Similar results are obtained from the Frequentist model of Fig. 15. Thus, changing the behaviors of an agent v_k 's neighbors, keeping v_k 's model fixed, can produce changes in the performance of v_k , thus demonstrating that agents do interact and affect each others' behaviors.

6. Related Work

Modeling of network games and data. Modeling of network games refers to the use of mathematical models to analyze and understand strategic interactions among multiple agents in a networked environment. There are multiple works [48, 59, 60] related to the proposed method. Hu *et al.* [48] proposed a versatile uncertainty quantification framework for modeling behaviors of the networked anagram game using asymptotic normal distributions for the model parameter. The paper [59] introduces a Bayesian model for analyzing human behavior in anagram games to alleviate the data scarcity problem appearing in [48], while [60] (the preliminary version of this work) proposes a Bayesian approach for visualizing uncertainty and identifying the best, average, and worst behaviors in a networked anagram game model. This work involves a comparison of the results obtained from uncertainty visualization using both Frequentist and Bayesian approaches. The goal of this comparison is to evaluate the strengths and limitations of each approach and to determine which approach is best suited for a particular modeling problem or application. By comparing the results of the two approaches, we can gain insights into the underlying assumptions and limitations of each approach and develop a better understanding of how uncertainty affects models and predictions. Other games incorporate multiple player actions over time, e.g. [53, 61, 83]. These games, like ours, use fixed networks. Other types of network models, for other phenomena, use evolving networks, e.g. [26].

Our anagram game fits within the larger context of human subject games. This research area is vast, and we do not cover it in detail here. Instead, we refer the reader to two recent large surveys of games [51, 69]. Among the topics covered in [69] are

different games that investigate behaviors that could be active in NGrAGs, including tit-for-tat, trust, cooperation, and coordination. Other behaviors, that are not part of our game, include punishment and tolerance; see [69] for details. In [51], several subjects, such as traffic flow, crime, climate change, and epidemiology are used as contexts to examine social behaviors. In both works, use of networks is prominent, since they enable one to control which subjects or entities interact, as is done here.

Estimating transition probability. A Markov chain is a frequently used mathematical model that describes a system that transitions between a finite number of states over time. In a discrete-time Markov chain (also known as a stochastic process), the system evolves in time in a series of discrete steps, and the probability of transitioning from one state to another depends only on the current state and not on any previous history. This property is known as the Markov property, and it is the defining characteristic of a Markov chain [54, 66]. The maximum likelihood estimation of the transition probability is easily derived as the sample proportion of transitions from each state to all possible states of the system. Several researchers have examined in detail the statistical properties of these estimators and developed tests for a series of hypotheses [9, 17]. Lee *et al.* [58] suggested a Bayesian estimation that used multivariate beta distribution as the prior probability density function. If a system is observed at infrequent time intervals, there may be missing data in the Markov chains. In such cases, an Expectation–Maximization (EM) algorithm can be used to estimate the transition probabilities [28, 78, 81]. In our model, we also want to explore covariates effects (e.g. players’ activity levels) on transition probabilities, so the Multinomial logistic model is employed to explore the dynamics in the NGrAGs.

Uncertainty visualization. Visualization is a powerful tool for exploring and understanding data, and it is particularly valuable in identifying any sources of uncertainty or variability that may exist in data [24, 25]. The effective visualization of uncertainty is commonly recognized as a challenging task [50]. Pang *et al.* [67] proposed a classification system for uncertainty visualization techniques and described various methods for representing uncertainty. Since their work, significant advances have been made in uncertainty visualization, and several research papers have provided a summary of the state-of-the-art techniques and challenges in the field [18, 19, 52, 71]. For the Bayesian approach, Gabry *et al.* [36] illustrated the role of visualization in exploratory data analysis in the context of a Bayesian workflow. House *et al.* [46] developed Bayesian visual analytics (BaVA) to justify Bayesian sequential update of parameters. The objective of our work is to develop effective visualizations that enable accurate identification of uncertainties present in the data, as well as the heterogeneous behaviors of players. By visualizing uncertainty, we can facilitate a more thorough understanding of the complexities and limitations of the data, thus seeking to contribute to the scientific understanding of complex systems.

7. Discussion and Conclusion

In this paper, we introduce and assess two uncertainty visualization methods for a complex system, i.e. multi-player games. We demonstrate the effectiveness of these methods through a step-by-step implementation on a networked group anagram game, where players cooperate to form words by sharing letters. The visualizations produced by our methods provide a valuable tool for evaluating model uncertainties and facilitating the interpretation of player behaviors. Additionally, we have developed software modules based on these models that enable the simulation of game conditions beyond the scope of the experiments.

The visualization of uncertainty in multi-player game data holds significant value in various aspects. On one hand, it enables game designers to gain a better understanding of the level of uncertainty present in the game data, aiding in identifying areas where the game can be improved and input spaces where more (or fewer) experiments need to be run. On the other hand, it can enhance the performance of players by providing them with a clearer comprehension of the risks and rewards associated with different actions, thus enabling them to be more engaged in the game and make more informed decisions. Overall, uncertainty visualization is an important tool for analyzing and understanding multi-player game data. It can help players make better decisions, enhance player engagement, and aid in game development and design.

It is worth to remark that the increase in population size might influence the results. Specifically, a larger population size on data points will alleviate the data imbalance issue in model estimation. Increasing the number of players in the game, restricted by experimental budget and experimental setup (e.g. number of neighbors each player has), could lead to heterogeneity of players' actions and potentially enhance the efficacy of uncertainty quantification. In the ABM simulation, a larger number of agents often implies more complex networks and could help gain deep and comprehensive understanding of the NGrAGs.

Further study includes modeling the feature-dependent Markov chains [11] and higher-order Markov chains [73]. Feature-dependent Markov chains are particularly useful in situations where the probability of transitioning between states varies depending on specific conditions or attributes of the complex system. By incorporating these features into the model, it becomes possible to more accurately predict the behavior of the complex system and deal with data that has missing values. Higher-order Markov models are valuable when the accuracy of predictions can be improved by considering a larger number of past observations.

References

- [1] Adamatzky, A., Goles, E., Martinez, G. J., Tsompanas, M.-A., Tegelaar, M. and Wosten, H. A. B., Fungal automata, *Complex Syst.* **29** (2020) 759–778.
- [2] Adams, A. M., The effects of interaction functions between two cellular automata, *Complex Syst.* **31** (2022) 203–217.

- [3] Adiga, A., Galyean, H., Kuhlman, C. J., Levet, M., Mortveit, H. S. and Wu, S., Activity in Boolean networks, *Nat. Comput.* (2016) 1–13.
- [4] Adiga, A., Kuhlman, C. J., Marathe, M. V., Mortveit, H. S., Ravi, S. S. and Vullikanti, A., Graphical dynamical systems and their applications to bio-social systems, *Int. J. Adv. Eng. Sci. Appl. Math.* **11** (2018) 1–19.
- [5] Adiga, A., Kuhlman, C. J., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J. and Stearns, R. E., Inferring local transition functions of discrete dynamical systems from observations of system behavior, *Theor. Comput. Sci.* **679** (2017) 126–144.
- [6] Adiga, A., Kuhlman, C. J., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J. and Stearns, R. E., Learning the behavior of a dynamical system via a “20 questions” approach, in *Proc. Thirty-Second AAAI Conf. Artificial Intelligence (AAAI-18)* (2018), pp. 4630–4637.
- [7] Albert, J. H. and Chib, S., Bayesian analysis of binary and polychotomous response data, *J. Amer. Stat. Assoc.* **88** (1993) 669–679.
- [8] Anderson, N. H., Group performance in an anagram task, *J. Soc. Psychol.* **55** (1961) 67–75.
- [9] Anderson, T. W. and Goodman, L. A., Statistical inference about Markov chains, *Ann. Math. Stat.* **28** (1957) 89–110.
- [10] Baas, N. A. and Helvik, T., Higher order cellular automata, *Adv. Complex Syst.* **8** (2005) 169–192.
- [11] Barratt, S. and Boyd, S., Fitting feature-dependent Markov chains, *J. Global Optimiz.* (2022) 1–18.
- [12] Barrett, C. L., III, H. B. H., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J. and Stearns, R. E., Reachability problems for sequential dynamical systems with threshold functions, *Theor. Comput. Sci.* **295** (2003) 41–64.
- [13] Barrett, C. L., III, H. B. H., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J. and Stearns, R. E., Complexity of reachability problems for finite discrete dynamical systems, *J. Comput. Syst. Sci.* **72** (2006) 1317–1345.
- [14] Barrett, C. L., III, H. B. H., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J. and Stearns, R. E., Modeling and analyzing social network dynamics using stochastic discrete graphical dynamical systems, *Theor. Comput. Sci.* **412** (2011) 3932–3946.
- [15] Barrett, C. L., III, H. B. H., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J., Stearns, R. E. and Thakur, M., Predecessor existence problems for finite discrete dynamical systems, *Theor. Comput. Sci.* **386** (2007) 3–37.
- [16] Bayes, T., LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S, *Philos. Trans. R. Soc. London* **53** (1763) 370–418.
- [17] Billingsley, P., Statistical methods in Markov chains, *Ann. Math. Stat.* **32** (1961) 12–40.
- [18] Bonneau, G.-P., Hege, H.-C., Johnson, C. R., Oliveira, M. M., Potter, K., Rheingans, P. and Schultz, T., Overview and state-of-the-art of uncertainty visualization, in *Scientific Visualization: Uncertainty, Multifield, Biomedical and Scalable Visualization* (Springer, 2014), pp. 3–27.
- [19] Brodlie, K., Allendes Osorio, R. and Lopes, A., A review of uncertainty in data visualization, *Expanding the Frontiers of Visual Analytics and Visualization* (Springer, 2012), pp. 81–109.
- [20] Cadsby, C. B., Song, F. and Tapon, F., Sorting and incentive effects of pay for performance: An experimental investigation, *Acad. Manage. J.* **50** (2007) 387–405.
- [21] Cadsby, C. B., Song, F. and Tapon, F., Are you paying your employees to cheat? an experimental investigation, *B.E. J. Econ. Anal. Policy* **10** (2010) 1–30.

- [22] Cedeno-Mieles, V., Hu, Z., Ren, Y., Deng, X., Adiga, A., Barrett, C. L., Contractor, N., Ekanayake, S., Epstein, J. M., Goode, B. J., Korkmaz, G., Kuhlman, C. J., Machi, D., Macy, M. W., Marathe, M. V., Ramakrishnan, N., Ravi, S. S., Saraf, P. and Self, N., Networked experiments and modeling for producing collective identity in a group of human subjects using an iterative abduction framework, *Soc. Netw. Anal. Min.* **10** (2020) 43.
- [23] Charness, G., Cobo-Reyes, R. and Jimenez, N., Identities, selection and contributions in a public-goods game, *Games Econ. Behav.* **87** (2014) 322–338.
- [24] Chen, C.-H., Härdle, W. K. and Unwin, A., *Handbook of Data Disualization* (Springer Science & Business Media, 2007).
- [25] Cook, D., Lee, E.-K. and Majumder, M., Data visualization and statistical graphics in big data analysis, *Annu. Rev. Stat. Appl.* **3** (2016) 133–159.
- [26] Dankulov, M. M., Melnik, R. and Tadic, B., The dynamics of meaningful social interactions and the emergence of collective knowledge, *Sci. Rep.* **5** (2015) 12197–1–12197–10.
- [27] Davis, W. L. and Davis, D. E., Internal-external control and attribution of responsibility for success and failure, *J. Personal.* **40** (1972) 123–136.
- [28] Dempster, A. P., Laird, N. M. and Rubin, D. B., Maximum likelihood from incomplete data via the EM algorithm, *J. R. Stat. Soc. Ser. B (Methodol.)* **39** (1977) 1–22.
- [29] Dominowski, R. L., Anagram solving as a function of letter moves, *J. Verb. Learn. Verbal Behav.* **5** (1966) 107–111.
- [30] Dominowski, R. L., The effect of pronunciation practice on anagram difficulty, *Psychon. Sci.* **16** (1969) 99–100.
- [31] Dreyer, P. and Roberts, F., Irreversible k -threshold processes: Graph-theoretical threshold models of the spread of disease and opinion, *Discr. Appl. Math.* **157** (2009) 1615–1627.
- [32] Feather, N. T., Attribution of responsibility and valence of success and failure in relation to initial confidence and task performance, *J. Pers. Soc. Psychol.* **13** (1969) 129–144.
- [33] Feather, N. T. and Simon, J. G., Attribution of responsibility and valence of outcome in relation to initial confidence and success and failure of self and other, *J. Pers. Soc. Psychol.* **18** (1971) 173–188.
- [34] Feather, N. T. and Simon, J. G., Causal attributions for success and failure in relation to expectation of success based upon selective or manipulative control, *J. Pers. Soc. Psychol.* **39** (1971) 527–541.
- [35] Fisher, R. A., On the mathematical foundations of theoretical statistics, *Philos. Trans. R. Soc. London A* **222** (1922) 309–368.
- [36] Gabry, J., Simpson, D., Vehtari, A., Betancourt, M. and Gelman, A., Visualization in Bayesian workflow, *J. R. Stat. Soc. Ser. A (Stat. Soc.)* **182** (2019) 389–402.
- [37] Geman, D., Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **6** (1984) 721–741.
- [38] Gilhooly, K. J. and Johnson, C. E., Effects of solution word attributes on anagram difficulty: A regression analysis, *Q. J. Exper. Psychol.* **30** (1978) 57–70.
- [39] Goldman, M., Stockbauer, J. W. and McAuliffe, T. G., Intergroup and intragroup competition and cooperation, *J. Exper. Soc. Psychol.* **13** (1977) 81–88.
- [40] Goles, E. and Martinez, S., *Neural and Automata Networks* (Kluwer Academic Publishers, Dordrecht, The Netherlands, 1990).
- [41] Goles, E. and Montealegre, P., Computational complexity of threshold automata networks under different updating schemes, *Theor. Comput. Sci.* **559** (2014) 3–19.
- [42] Griffiths, W. E., Hill, R. C. and Pope, P. J., Small sample properties of probit model estimators, *J. Amer. Stat. Assoc.* **82** (1987) 929–937.

- [43] Hastings, W., Monte Carlo sampling methods using Markov chains and their application, *Biometrika* **57** (1970) 97–109.
- [44] Hazra, S. and Dalui, M., Characterization of single length cycle two-attractor cellular automata using next-state rule minterm transition diagram, *Complex Syst.* **31** (2022) 363–388.
- [45] Hou, W., Tamura, T., Ching, W.-K. and Akutsu, T., Finding and analyzing the minimum set of driver nodes in control of Boolean networks, *Adv. Complex Syst.* **19** (2016) 1650006–1–1650006–32.
- [46] House, L., Leman, S. and Han, C., Bayesian visual analytics: Bava, *Stat. Anal. Data Min. ASA Data Sci. J.* **8** (2015) 1–13.
- [47] Hu, Z., Deng, X. and Kuhlman, C. J., An uncertainty quantification approach for agent-based modeling of human behavior in networked anagram games, in *2021 Winter Simulation Conf. (WSC)* (IEEE, 2021), pp. 1–12.
- [48] Hu, Z., Deng, X. and Kuhlman, C. J., Versatile uncertainty quantification of contrastive behaviors for modeling networked anagram games, in *Complex Networks & Their Applications X: Volume 1, Proc. Tenth Int. Conf. Complex Networks and Their Applications* (Springer, 2022), pp. 644–656.
- [49] Jeffreys, H., An invariant form for the prior probability in estimation problems, *Proc. R. Soc. London A, Math. Phys. Sci.* **186** (1946) 453–461.
- [50] Johnson, C. R. and Sanderson, A. R., A next step: Visualizing errors and uncertainty, *IEEE Comput. Graph. Appl.* **23** (2003) 6–10.
- [51] Jusup, M., Holme, P., Kanazawa, K., Takayasu, M., Romic, I., Wang, Z., Gecek, S., Lipic, T., Podobnik, B., Wang, L., Luo, W., Klanjscek, T., Fan, J., Boccaletti, S. and Perc, M., Statistical physics of human cooperation, *Soc. Phys.* **948** (2022) 1–148.
- [52] Kamal, A., Dhakal, P., Javaid, A. Y., Devabhaktuni, V. K., Kaur, D., Zaients, J. and Marinier, R., Recent advances and challenges in uncertainty visualization: A survey, *J. Vis.* **24** (2021) 861–890.
- [53] Kearns, M., Suri, S. and Montfort, N., An experimental study of the coloring problem on human subject networks, *Science* **313** (2006) 824–827.
- [54] Kemeny, J. G. and Snell, J. L., *Finite Markov Chains: With a New Appendix" Generalization of a Fundamental Matrix"* (Springer, 1976).
- [55] Kuhlman, C. J., Korkmaz, G., Ravi, S. and Vega-Redondo, F., Theoretical and computational characterizations of interaction mechanisms on facebook dynamics using a common knowledge model, *Soc. Netw. Anal. Min* **11** (2021) 116.
- [56] Kuhlman, C. J. and Mortveit, H. S., Attractor stability in nonuniform Boolean networks, *Theor. Comput. Sci.* **559** (2014) 20–33.
- [57] Kulis, B. and Jordan, M. I., Revisiting k-means: New algorithms via Bayesian non-parametrics, arXiv:1111.0352.
- [58] Lee, T. C., Judge, G. and Zellner, A., Maximum likelihood and Bayesian estimation of transition probabilities, *J. Amer. Stat. Assoc.* **63** (1968) 1162–1179.
- [59] Liu, X., Hu, Z., Deng, X. and Kuhlman, C. J., A Bayesian uncertainty quantification approach for agent-based modeling of networked anagram games, in *2022 Winter Simulation Conf. (WSC)* (IEEE, 2022), pp. 310–321.
- [60] Liu, X., Hu, Z., Deng, X. and Kuhlman, C. J., Bayesian approach to uncertainty visualization of heterogeneous behaviors in modeling networked anagram games, in *Complex Networks and Their Applications XI: Proc. Eleventh Int. Conf. Complex Networks and their Applications: Complex Networks* (Springer, 2023), pp. 595–607.
- [61] Mason, W. and Watts, D. J., Collaborative learning in networks, *PNAS* **109** (2012) 765–769.

- [62] Mayzner, M. S. and Tresselt, M. E., Anagram solution times: A function of letter order and word frequency, *J. Exp. Psychol.* **56** (1958) 376.
- [63] Meeker, W. Q. and Escobar, L. A., Teaching about approximate confidence regions based on maximum likelihood estimation, *Amer. Stat.* **49** (1995) 48–53.
- [64] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E., Equation of state calculations by fast computing machines, *J. Chem. Phys.* **21** (1953) 1087–1092.
- [65] Mortveit, H. and Reidys, C., *An Introduction to Sequential Dynamical Systems* (Springer, New York, 2007).
- [66] Norris, J. R., *Markov Chains*, Vol. 2 (Cambridge University Press, 1998).
- [67] Pang, A. T. et al., Approaches to uncertainty visualization, *Vis. Comput.* **13** (1997) 370–390.
- [68] Papsdorf, J. D., Himle, D. P., McCann, B. S. and Thyer, B. A., Anagram solution time and effects of distraction, sex differences and anxiety, *Perc. Motor Skills* **55** (1982) 215–222.
- [69] Perc, M., Jordan, J. J., Rand, D. G., Wang, Z., Boccaletti, S. and Szolnoki, A., Statistical physics of human cooperation, *Phys. Rep.* **687** (2017) 1–51.
- [70] Polletta, F. and Jasper, J. M., Collective identity and social movements, *Annu. Rev. Sociol.* **27** (2001) 283–305.
- [71] Potter, K., Rosen, P. and Johnson, C. R., From quantification to visualization: A taxonomy of uncertainty visualization approaches, in *IFIP Working Conf. Uncertainty Quantification* (Springer, 2011), pp. 226–249.
- [72] Qiu, Z., Chen, C., Marathe, M. V., Ravi, S. S., Rosenkrantz, D. J., Stearns, R. E. and Vullikanti, A., Finding nontrivial minimum fixed points in discrete dynamical systems, arXiv:2301.04090.
- [73] Raftery, A. E., A model for high-order Markov chains, *J. R. Stat. Soc. Ser. B (Methodol.)* **47** (1985) 528–539.
- [74] Rosenkrantz, D. J., Marathe, M., Ravi, S. S. and Stearns, R. E., Synchronous dynamical systems on directed acyclic graphs: Complexity and algorithms, in *Proc. AAAI Conf. Artificial Intelligence* (2021), pp. 11334–11342.
- [75] Russell, D. G. and Sarason, I. G., Test anxiety, sex and experimental conditions in relation to anagram solution, *J. Pers. Soc. Psychol.* **1** (1965) 493–496.
- [76] Sanchez, A., Group identity and charitable contributions: Experimental evidence, *J. Econ. Behav. Organ.* **194** (2022) 542–549.
- [77] Schneider, F., Gur, R. E., Alavi, A., Seligman, M. E., Mozley, L. H., Smith, R. J., Mozley, P. D. and Gur, R. C., Cerebral blood flow changes in limbic regions induced by unsolvable anagram tasks, *Amer. J. Psych.* **153** (1996) 206–212.
- [78] Sherlaw-Johnson, C., Gallivan, S. and Burrige, J., Estimating a Markov transition matrix from observational data, *J. Oper. Res. Soc.* **46** (1995) 405–410.
- [79] Stones, C. R., Self-determination and attribution of responsibility: Another look, *Psychol. Rep.* **53** (1983) 391–394.
- [80] Sweeting, T. J., Uniform asymptotic normality of the maximum likelihood estimator, *Ann. Stat.* **8** (1980) 1375–1381.
- [81] Toussaint, M. and Storkey, A., Probabilistic inference for solving discrete and continuous state Markov decision processes, in *Proc. 23rd Int. Conf. Machine Learning* (2006), pp. 945–952.
- [82] Tresselt, M. E., Reexamination of anagram problem solving, *Trans. New York Acad. Sci.* **30** (1968) 1112–1119.
- [83] Unakafov, A. M. et al., Emergence and suppression of cooperation by action visibility in transparent games, *PLoS Comput. Biol.* **16** (2020) e1007588.

- [84] Van Ravenzwaaij, D., Cassey, P. and Brown, S. D., A simple introduction to Markov chain Monte Carlo sampling, *Psych. Bull. Rev.* **25** (2018) 143–154.
- [85] Wald, A., Tests of statistical hypotheses concerning several parameters when the number of observations is large, *Trans. Amer. Math. Soc.* **54** (1943) 426–482.
- [86] Warren, M. W. and Thomson, W. J., Anagram solution as a function of transition probabilities and solution word frequency, *Psych. Sci.* **17** (1969) 333–334.
- [87] Watson, G. B., Do groups think more effectively than individuals? *J. Abnorm. Soc. Psychol.* **23** (1928) 328–336.
- [88] Zellner, A. and Rossi, P., Bayesian analysis of dichotomous quantal response models, *J. Econom.* **25** (1984) 365–393.
- [89] Zertuche, F., On the number of NK-Kauffman networks mapped into a functional graph, *Complex Syst.* **25** (2016) 329–345.