# Technical Report: Modeling Anagram Games

Vanessa Cedeno-Mieles[*‡]
Zhihao Hu[†]
Yihui Ren[§]
Xinwei Deng[†]

Department of Computer Science[*]
Department of Statistics[†]
Virginia Tech
Blacksburg, Virginia
Escuela Superior Politécnica del Litoral, ESPOL[‡]
Guayaquil, Ecuador
Brookhaven National Laboratory[§]
Upton, New York

Abhijin Adiga
Christopher Barrett
Saliya Ekanayake
Gizem Korkmaz
Chris J. Kuhlman
Dustin Machi
Madhav V. Marathe
S. S. Ravi
Biocomplexity Institute
University of Virginia
Charlottesville, Virginia

Brian J. Goode[‖]
Naren Ramakrishnan[**]
Parang Saraf[**]
Nathan Self[**]
Biocomplexity Institute[‖]
Discovery Analytics Center[**]
Virginia Tech
Blacksburg, Virginia

Noshir Contractor
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, Illinois

Joshua M. Epstein
Department of Epidemiology
New York University
New York, New York
Santa Fe Institute
Santa Fe, New Mexico

Michael W. Macy
Department of Sociology
Cornell University
Ithaca, New York

## ABSTRACT

In anagram games, players are provided with letters for forming as many words as possible over a specified time duration. Anagram games have been used in controlled experiments to study problems such as collective identity, effects of goal-setting, internal-external attributions, test anxiety, and others. The majority of work on anagram games involves individual players. Recently, work has expanded to group anagram games where players cooperate by sharing letters. In this work, we analyze experimental data from web-based, networked experiments of group anagram games. We develop mechanistic and data-driven models of human reasoning to predict detailed game player actions (e.g., what word to form next). With these results, we develop a composite agent-based model that incorporates the models from data analysis. We compare model predictions against experimental data, which enables us to provide explanations of human reasoning and behavior. Finally, we provide illustrative case studies using agent-based simulations to demonstrate the efficacy of models to provide insights that are beyond those from experiments alone.


*WebSci '19, June 30–July 3, 2019, Boston, MA*

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

## CCS CONCEPTS

• **Networks** → **Network experimentation**; • **Computing methodologies** → **Agent / discrete models**.

## KEYWORDS

web-based social experiments, mechanistic models, data-driven models, agent-based modeling, group anagrams game

**ACM Reference Format:**
Vanessa Cedeno-Mieles, Zhihao Hu, Yihui Ren, Xinwei Deng, Abhijin Adiga, Christopher Barrett, Saliya Ekanayake, Gizem Korkmaz, Chris J. Kuhlman, Dustin Machi, Madhav V. Marathe, S. S. Ravi, Brian J. Goode, Naren Ramakrishnan, Parang Saraf, Nathan Self, Noshir Contractor, Joshua M. Epstein, and Michael W. Macy. 2019. Technical Report: Modeling Anagram Games. In *WebSci '19: A ACM Conference on Web Science, June 30–03, 2019, Boston, MA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

### 1.1 Background and Motivation

In an **individual anagram game**, a player is provided with a set of alphabetical letters to form as many words as possible with those letters in a prescribed time duration. For example, from the letters "u," "b," "g," and "t," a player can form the words "bug" and "tub," among others. The performance of a player is often quantified based on the number of words formed.

In a **group anagram game**, multiple players work cooperatively/collaboratively. Each player is given letters and forms words with her own letters, and can share letters with her neighbors to enable everyone to form more words. Figure 1 provides a schematic

of a 3-player group anagram game. Each player ($v_1$, $v_2$, and $v_3$) is initially provided with $n_l = 3$ letters as shown. A player may form words, and through the communication channels in gray, may request letters and reply to letter requests.

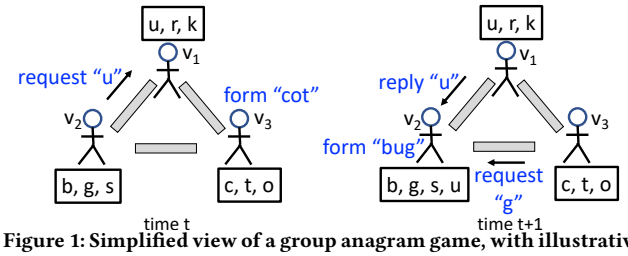


**Figure 1: Simplified view of a group anagram game, with illustrative actions among $n = 3$ players that communicate and share letters through the gray channels. Each player is initially given $n_l = 3$ letters. Letters that a player has "in-hand" to form words are shown in boxes. Player actions are shown in blue. At time $t$, $v_2$ requests a "u" from $v_1$ and $v_3$ forms the word "cot." At the next time, $v_2$ receives a "u" from $v_1$, forms the word "bug," and receives a request from $v_3$.**

Overwhelmingly, research on anagram games considers the individual setting. It has been extensively studied for more than 60 years to analyze the effects of goal-setting, compensation types, internal-external attributions, and test anxiety (e.g., [9, 11]). Other names for anagram game are *word formation game* and *word construction game*.

There are several reasons to study group anagram games. Face-to-face group anagram games have recently been played. In particular, [8] used them to study experimentally the formation of collective identity (CI), as defined in social psychology [23] as an individual's cognitive, moral, and emotional connection with a broader community, category, practice, or institution [24]. A second motivation is their relevance to other types of group dynamics, notably intergroup and intragroup cooperation and competition (e.g., [12]). A third motivation is that many of the phenomena listed above for the individual anagram game could be studied in group settings with models of group behavior.

Overall, researches involving anagram games encompass a broad range of disciplines like sociology, economics, management science, and (social) psychology [5, 9, 11]. It is clear that using anagram games is valuable in various fields of research. With all of this experimental work on anagram games, it is surprising that very little work has been done in modeling and simulating these games. In fact, the first work modeling human group anagram games—to our knowledge the only such work—was recently completed [25]. We enumerate the differences between our work and [25] in Section 2.

## 1.2 Novelty of Work

In this work, we conduct online group anagram games through a web application (web app). We analyze these data to develop individual models of player behaviors in the game. We implement and validate the models using an agent-based modeling (ABM) approach to represent and model each player in a game. We run simulations beyond preset experimental conditions to uncover interesting team behaviors.

There are several novel aspects of our work. First, we devise well-defined models of human *reasoning* that produce as output human *actions* in the game, and that are formalized in algorithms. Furthermore, our models are combinations of mechanistic and data-driven models. Mechanistic models, for our purposes here, have the following characteristics: (*i*) the models are based on first principles and are not tied to any particular domain; and (*ii*) the models can be specified and implemented without any experimental data. For example, we can build word formation and letter request agent-based models (ABMs) based solely on the mechanistic models, without any anagram game data. In addition, we employ specific mechanistic concepts like Levenshtein distance (LD) [19] and a greedy optimization procedure to represent human reasoning in mechanistic models. For example, LD has never been used in models as we do here. As explained in the contributions, these models are related to cognitive and economic theories. Note that we operationalize these theories in terms of the anagram game.

Item (*i*) above is particularly relevant in developing explanatory models because our mechanistic models are decoupled from the application, and thus our mechanistic models' structures have explanatory power [4, 16]. This ability of our models to explain human behavior in anagram games is our second novelty.

A third novelty is that our paper is an exemplar of a detailed procedure for combining mechanistic and data-driven models to form a single model of behavior. Here, data-driven models are essentially stochastic models built from data analyses of experiments; thus, it is an abductive process [13]. With these models, we develop an agent-based modeling (ABM) platform to accurately model the detailed actions of players in group anagram games. This is a fourth novelty. This enables us to investigate phenomena beyond those uncovered via our experiments.

As called for in the social sciences, (1) our focus is on model predictions and explanations [14], and (2) we provide methods and solutions for a human behavior game of wide interest [29].

## 1.3 Contributions

**1. A process for combining mechanistic and data-driven approaches to build models of human reasoning based on group anagram games experimental data.** We provide the details of our process in Section 5. We first conjecture mechanistic models, and evaluate them by comparing their predictions to experimental data. This does two things: (*i*) enables comparisons of model predictions with experimental data, and if these comparisons are favorable (which they are), then (*ii*) the structures of the models provide explanations for human reasoning [4, 16]. Second, because the mechanistic models can be improved by including data from experiments, we use data-driven modeling approaches to introduce stochasticity to account for variability across human subject game players. Hence we utilize these two modeling approaches in a well-defined process. These ideas are embodied in Figure 2.

**2. New experimental findings and explanations of player behaviors based on cognitive and economic theories.** The analyses focus on data for three types of player actions: (1) form a word; (2) request a letter; and (3) reply to a letter request. We compare these data with model predictions. The successful comparisons enable us to *explain* how players are *reasoning* and *behaving* in the games. These explanations are based on the structure of the mechanistic models and the results of data-driven modeling [4, 16]. See Figure 2. A summary of some explanations follows. A word that

a player forms is explained by considering the letters that the player has in-hand (i.e., in her possession) and LD [19]—an edit distance—between the most recently formed word and the next word to be formed from a candidate set of words (Section 5.2). This is motivated by, and consistent with, cognitive load theory [27] in that people try to reduce cognitive load during learning. Here, the closer the next word formed is to the previously formed word—as measured by LD—the lesser the cognitive load in forming a new word. We base the letter that a player requests on rational choice theory [3]. Our analyses (Section 5.3) demonstrate that the letter that a player requests from her neighbors is explained by identifying the letter that maximally increases the number of words that the player can form, when also considering the letters that the player has in-hand (greedy optimization algorithm). This behavior is consistent with rational choice theory. This is because players' earnings in games are proportional to the number of words formed, so it is rational for players to choose letters to maximize their candidate word set. It is interesting that our explanation means that players are reasoning beyond more naive approaches, such as simply requesting some "most frequently" used letter (e.g., preferring "e" over "z"). (We have modeled this second approach—results not shown here—and this model's results are not consistent with the data.) Finally, we also show that there are four types of behavior in replying to letter requests (Section 5.4)).

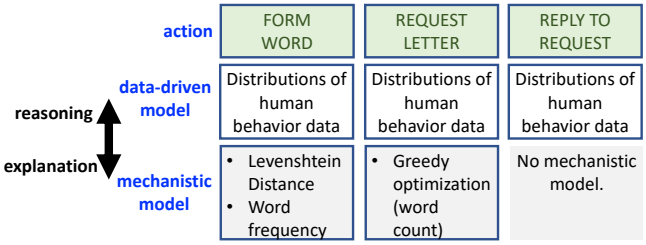


**Figure 2: Combined mechanistic and data-driven models for the three player actions in the group anagram game (see Figure 1). These are models of human reasoning, which output specific player actions in the game. The particulars of the mechanistic and data-driven models are given in the respective boxes under the actions and are detailed in Section 5.**

**3. Mechanistic and data-driven modeling of group anagram game.** We use concepts such as LD, word corpora, word proximity networks (WPNs), and a greedy optimization algorithm (defined in Section 5) to develop mechanistic models for two of the three player actions. Essentially, the models are mechanistic, but player actions within the models are also described by data-driven distributions that reflect variations in player-to-player performance (Figure 2). The models are algorithms of individual human reasoning, based on the state of the system (the anagram game), and the outputs are human behavior in the form of specific actions to take (e.g., what particular word to form).

A family of ABMs are developed, yielding a composite model, where each ABM is comprised of a distinct model for each of the three actions, with user-specified parameter values for player/agent characteristics, such as the agent's vocabulary and their aptitude, i.e., the degree to which they perform optimally. The multi-logit regression model based on [25] is adopted to determine which action each agent selects at each discrete time in a simulation (time granularity is seconds). The selected action then invokes the appropriate model developed herein. Note that there is a fourth action, a no-operation (no-op), where the agent does nothing at particular times, which represents agent thinking and requires no model.

**4. Comparisons of model results with experimental data.** Evaluations are based on individual player actions. We compare model predictions of word formation and letter requests to both experimental data and a null model (a uniformly random model). It is found that the median behavior of our models are closer to the median experimental data, compared to the null model.

**5. Novel insights from exercising agent-based models.** We build an agent-based modeling and simulation (ABMS) platform and run simulations of anagram games using our models in Contribution 3. We illustrate how different parameter settings produce interesting insights, beyond those provided by experiments (which is another characteristic of explanatory models [4, 16]). For example, we quantity how the average number of words formed per player decreases with decreasing word formation and letter request aptitudes of players. Our platform scales well beyond the numbers of agents used herein, but our goal here is to model games on par with the experiments.

**Remark.** We have performed many analyses of player behavior in the game. Due to space limitations, we focus on our best models, explanations, and algorithms. Our work, in this sense, is abductive [13, 28], as we devise models that explain the experimental data. We present the approaches described herein based on (*i*) mechanistic rather than purely statistical approaches; (*ii*) ability of models to explain observed behavior; and (*iii*) simplicity of the models. We explicitly state that our explanations and models may not be the only viable ones. This is not a short-coming of this work: more experiments can always be done to provide additional data to improve models [13, 16, 28].

## 2 RELATED WORK

**Anagram experiments.** Over 20 experiment works (e.g., [9, 11]) use *single player* anagram games. The only *face-to-face* cooperative team anagram games are reported in [8]. The game is used to foster CI among teammates. Multiple simultaneously-playing teams can change composition (4-player teams) as players vote others into and off the team. This motivated the *online* experiments in [25] where one team of fixed size plays the game through a web browser.

**Agent-based models of anagram games.** To the best of our knowledge there is only one work that models the anagram game. [25] designs, constructs, and evaluates three data-driven ABMs as part of an iterative abductive analysis. Differences between our work and [25] include the following. First, the ABMs in [25] are purely statistical; there are no mechanistic models incorporated in the work, which limits transferability across contexts. Second, in [25], if a player action is form word, and the player has letters "q," "z," and "r," then their model will form an unspecified (unrealistic) word from these letters. Their work similarly handles other player actions. This is because that work is primarily concerned with predicting the time series of *types* of player actions, and not the details of actions. In our work here, with the above letters, no word is formed because no word can be constructed from these letters (and so would not be in a word corpus). Third, the previous

difference immediately implies that if simulations are run many times and the results are averaged, then all agent model results will tend to the *same* mean behavior (e.g., the same number of letter requests) in [25]. Our ABMs account for initial conditions, letters in-hand, and particular player parameter assignments—all of which can vary among players—so results will remain distinct across agents. Consequently, we can account for differences, for example, in the quality of letters assigned to players. That is, our models are of much greater fidelity.

**Online experiments and data-driven modeling.** There are several online experiments using data-driven modeling to analyze different phenomena like CI, diffusion, etc. In [6], a statistical aggregated model is created to analyze human cooperation through social dilemmas. In [2], a statistical aggregated model provides a network-theoretic approach to empirically determine the structural signatures of online collective identity. In [7], online health forums are used to study diffusion. Apart from [25], none of these works build ABMs from networked experiments where individuals take a series of actions that may be repeated over time. Our experiments are comparable in size to these works.

**Networked experiments and modeling.** There are several other online (e.g., [20]) and in-person (e.g., [8, 18] ) experiments with interacting participants that can be represented as networks, and analyses of network populations (e.g., [21, 22]), where edges represent interaction channels.

**Explanatory modeling.** There are many works (e.g., [4, 15, 16]) that describe different definitions of explanations, different types of explanations that models provide, and procedures for arriving at explanations. We follow ideas from [4, 16] (principally, that the structure of mechanistic models that adequately predict human behavior can be used to explain behavior). As a counter example, [15] discusses the need for visual representations for simulation explanation when models are not well understood. However, it is precisely our goal in this paper to describe our modeling approach, our models, and our experimental justification for them. We present well-motivated, well-defined, formal models in Section 5 that are also used in ABM.

## 3 WEB-BASED GROUP ANAGRAMS EXPERIMENTS

We built a customized web application (web app) for a group anagrams game consisting of game instructions, a workflow, set of user screens, and data capture software (all button clicks by all players are recorded), which sits on top of an off-the-shelf infrastructure [10]. Players are provided game instructions, participate in the online anagrams game through their web browsers, and are paid based on their performance. A total of 48 experiments were performed using a total of 367 players, with numbers of players per game ranging from 3 to 17. In the following, we describe the team anagrams game/experiment (omiting details such as recruitment procedures and game instructions).

Figure 3 provides a visual description of the game setup. A game begins with $n$ players, $v_1$ through $v_n$. Each player has a degree $d$ that specifies the number of connections to other players. A connection (edge) between two players denotes a communication channel where a letter $\ell$ can be requested and sent (sending a letter is a reply). Thus, an experiment configuration is a graph $G(V, E)$ with player set $V$ and communication channels $E$. In experiments, $G$ is a $k$-regular random graph, with uniform degree $2 \leq k \leq 8$. Each player starts the game with $n_\ell$ initial letters, which they can use to form words or share among their neighbors, when requested. At the beginning of a game, a Word Corpus $C^W$ is defined with a list of words a player can form during the game. A key to the important data structures per player in a game, and their descriptions, are in Figure 3. The three major player actions in a game are now described.

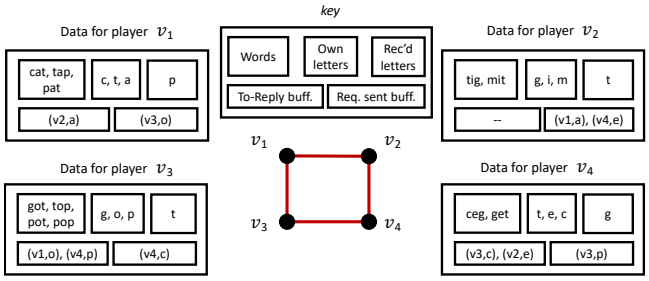


**Figure 3: Example anagrams game configuration with a $k = 2$-regular graph on $n = 4$ players and $n_l = 3$ letters *initially* assigned to each player. Requests for letters and replies with letters are sent across the links. The key indicates the types of bins and buffers that contain specific information for players. For example, the To-Reply buff. (reply buffer) of player $v_i$ contains letter requests from other players $v_j$, in the form ($v_j$, letter), to which $v_i$ has not yet replied. Analogously, the Requests sent buff. (buffer) keeps track of a player's letter requests that have yet to be filled. Own (received) letters are letters a player has originally or received from neighbors, to use in forming words.**

**Player action: forming a word.** At any point during a game, a player $v_i$ can form a word $w_i$. All letters in the word $w_i$ must come from the set of letters $v_i$ has in-hand $L_i^{ih}$ (superscript $ih$). A specific letter $\ell$ can appear any number of times in a word. For a word submission to be accepted in the game, the word has to be in the game word corpus $C^W$.

**Player action: requesting a letter.** At any point during a game, a player $v_i$ can request a letter $\ell_{ij}^{req}$ from a neighbor $v_j$'s set of $n_\ell$ *initial* letters $L_j^{init}$. The anagrams game screen shows all neighbors' *initial* letters as available for request. A letter received by $v_i$ is put into the set $L_i^{ih}$.

**Player action: replying with a letter.** At any point during a game, a player $v_i$ can reply with a letter $\ell_{ij}^{rep}$ to a neighbor $v_j$'s request ($\ell_{ij}^{rep}$ must be in $L_i^{init}$). The anagrams game screen for $v_i$ shows all of the letters requested of $v_i$.

To encourage cooperation, any letter in $L_i^{ih}$ can be used any number of times in forming words, and the letter is not lost; the letter bestows an infinite supply of use. Similarly, if $v_i$ requests a letter $\ell$ from $v_j$, and $v_j$ replies with it, $v_j$ still retains a copy of the letter and can use it. Also, earnings for the team is based on the total number of words formed, and all players receive $(1/n)$ of the total earnings. Typical player earnings are \$7 to \$10 per game.

## 4 AGENT-BASED MODEL OVERVIEW

We refer to our ABM as a **composite model** or a **model family** because it consists of an action selection module and three component modules for the actions: form word, request letter, and reply to (letter) request. See Figure 4. We use the multinomial logistic regression approach in [25] for the action selection module, which is based on four variables that account for history effects and the state of each player (see Figure 3 for an example of player states). The four variables are the size of the To-Reply buffer, the number of letter in-hand, the number of words formed, and the number of consecutive actions. See [25] for details; we discuss it no further.

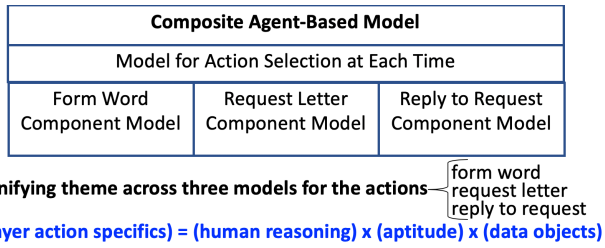


**Figure 4: Design of the composite agent-based model. At each time in a group anagrams game, a player takes one of four actions, consistent with the online game: form word, request letter, reply to letter request, or think. The selection of each action is determined by a multinomial logistic regression model from [25]. Each of the first three actions requires a component model (software module) that simulates human reasoning and outputs the *specifics* of an action. The algorithms for these three actions are described in Figure 8, Figure 11, and Figure 13. Thinking is an idling action, no model is needed. The common theme across these models is given in blue. This structure is intimately linked with the concepts in Figure 2.**

Our focus is on the three component models of Figure 4. The goal is to devise mechanistic and data-driven models for these three actions. By specifying different parameters and behaviors for each component model, we produce different ABMs; hence, our claim that our ABM is a composite model.

There is a unifying theme to these models, shown in Figure 4. For each model, the inputs are (*i*) key concepts in characterizing human reasoning that underpin the data and algorithm; (*ii*) an aptitude variable; and (*iii*) relevant data objects from the game state. The output is behavior: a *specific* human action (e.g., form the word "dog," reply to all current letter requests).

The control flow structure for simulations using ABM is provided in Figure 5. In essence, we loop over *runs* (i.e., individual game simulation instances), as multiple instances are used to assess stochasticity. For each run, we loop over time in discrete units of seconds, and for each time, we loop over all agents $v_i \in V$. Note that there are many definitions in the **Input** section of the figure that are subsequently used.

## 5 DATA ANALYSIS AND MODEL DEVELOPMENT

Each of the player actions from Figure 4 is described in turn. For each action—which is a component model of the ABM—we provide

**Input:** Graph of players and interactions $G(V, E)$ with players $v_i \in V$, $1 \le i \le n$, and communication and letter sharing channels $E$. Maximum duration of game (i.e., one run) $t_{max}$. Number of simulation runs $n_{runs}$ to execute. For each $v_i$: word corpus $C_i^W$; initial letter assignments $L_i^{init}$; words formed to this point in game $W_i^f$; aptitude in forming words $b_i^{wf}$; aptitude in requesting letters $b_i^{req}$; aptitude in replying to letter requests $b_i^{rpl}$; the initial letters of neighbors of $v_i$, $L_i'$, that $v_i$ can request. Distributions (several are families of distributions): distributions $\mathcal{D}^{d^L}(|C_i^W|, b_i^{wf}, d_{min}^L)$ of $d_{act}^L$ for a specific $d_{opt}^L$; distribution $\mathcal{D}^{wr}$ of word ranks; distributions $\mathcal{D}^{\ell r}(|C_i^W|, b_i^{req}, r_{num})$ of letter ranks; distribution $\mathcal{D}^{n^{rpl}}(b_i^{rpl})$ of the number of contiguous replies to make; distribution $\mathcal{D}^{\Delta t}(b_i^{rpl}, n^{rpl})$ for time duration over which letters are replied to.

**Output:** For each run, time, and agent $v_i$, output $v_i$'s action, payload (e.g., word formed, letter requested, letter received) to produce the entire simulation history.

**Steps:**

(1) Read and store inputs (see above).
(2) **for** $j = 1$ **to** $n_{runs}$ **do** ## *Loop over runs.*
  (a) Reset all parameters and initial conditions for a new simulation run.
  (b) Read in separate inputs that vary across runs.
  (c) **for** $t = 1$ **to** $t_{max}$ **do** ## *Loop over time.*
    (i) **for each** $v_i \in V$ **do** ## *Loop over agents.*
      (I) Determine next action $a_i$ for $v_i$ using multinomial logistic regression [25].
      (II) Based on $a_i$, call routine to execute action: Form Word (Figure 8), Request Letter (Figure 11), Reply To Request (Figure 13), or if the action is "think" (idle), then *no op*.
      (III) Output appropriate data for $v_i$.

**Figure 5: Steps of the Algorithm for ABM Simulation Control Flow. The action Form Word is described in Figure 8, the action Request Letter is described in Figure 11, and the action Reply To Request is described in Figure 13.**

(*i*) our premise for understanding players' behavior and the key concepts for this premise, (*ii*) experimental results for these key ideas that justify (i.e., give evidence for), and provide data for, the component model of the composite ABM, and (*iii*) a formal algorithm for the component model for the action in Figure 4. Note that (*a*) the steps of algorithms that we specify below are not focused on efficient implementation, but rather on conveying the steps of the algorithms as they relate to the data analyses, and (*b*) all of the data structures required for the algorithms, and more, are illustrated in Figure 3. First, we address preliminaries.

### 5.1 Preliminaries

We introduce two concepts used in data analysis and modeling. **Levenshtein distance** ($d^L$) [19], is prominent in our work and the work's novelty, and is motivated by work in linguistics and bioinformatics [17, 26]. It is a string metric that quantifies the difference in letters of two words. In starting with one word to obtain a second word, a letter substitution counts as one, as does each

of letter insertion and letter deletion. Hence, going from "had" to "hats," requires $d^L = 2$: one to substitute "t" for "d" and one for inserting an "s."

A **word proximity networks** (WPN) is a clique graph $H(V_H, E_H)$ where vertices $V_H$ are words that can be formed, according to a word corpus $C^W$, with the letters that a player currently has in-hand and $E_H$ is the set of edges between pairs of words, labeled with its $d^L$.

Each player is assigned a **word corpus** $C^W$. For this we use a list of the top 5000 words from the 450 million word Corpus of Contemporary American English, the only large and balanced corpus of American English [1].

## 5.2 Player Action: Form Word

**Basic premise and key concepts.** We seek to identify a method that explains the process of players selecting words to form. Our premise is that given the last word $w_1$ that $v_i$ has formed, the next word $w_2$ that $v_i$ will form will be one with minimal $d^L$ from $w_1$ because this requires a minimal number of letter manipulations (i.e., lesser cognitive load [27]). For the first word, $v_i$ selects the most frequent word from the corpus that can be formed with $L_i^{ih}$. We note that for each player $v_i$, there is a set $L_i^{ih}$ of letters that she has in-hand (superscript "ih") and a corresponding set $W_i^{ih} \subseteq C^W$ of words that $v_i$ can form from the entire corpus $C^W$ of words, based on the letters in $L_i^{ih}$. As $v_i$ requests and receives more letters from her neighbors, the cardinalities of $L_i^{ih}$ and $W_i^{ih}$ will (typically) increase. Also note that for a given word $w_1$ formed by $v_i$ in a game, $W_i^{ih}$ can be partitioned based on $d^L(w_1, w_2)$ for each $w_2 \in W_i^{ih}$ using the WPN. Let $W_i^{ih}(w_1, d^L) \subseteq W_i^{ih}$ be the set of words at $d^L$ from $w_1$ that $v_i$ can form.

Our data analysis is based on two central ideas, for each player $v_i$: $(i)$ we compare $d^L$ values between two consecutive words formed ($w_1$ and then $w_2$), both the actual value $d^L_{i,act}(w_1, w_2)$ measured from experiments and the optimal (i.e., minimal) value $d^L_{opt}(w_1, w^*)$ for some $w^* \in W_i^{ih}(w_1, d^L_{min})$, both of which are based on $v_i$'s set $L_i^{ih}$ [we drop the arguments when they are obvious from context]; and $(ii)$ for a given set $W_i^{ih}(w_1, d^L)$ of words, we use the popularity of words as provided by the rank (frequency of use) from [1] of the words in $W_i^{ih}(w_1, d^L) \subseteq C^W$. Here, $d^L_{min}$ is the minimum $d^L$ at which any word in $W_i^{ih}$ is from $w_1$. All of these parameters are either inputs (e.g., $C^W$), measured in experiments, or computed from experimental data. These high-level steps enable us to understand players' behavior in forming words, as described next.

**Data analysis.** First, for each player $v_i$ in the game, we consider pairs of consecutive words formed, $(w_1, w_2)$. From this, we compute $d^L_{i,act}(w_1, w_2)$, the actual $d^L$. Also from these data and from $L_i^{ih}$ at the time $w_2$ was formed, we can compute the set $W_i^{ih}(w_1, d^L_{min}) \subseteq W_i^{ih}$ of words that $v_i$ can form that are at $d^L_{min}$ from $w_1$. We compute $\Delta d^L = d^L_{i,act} - d^L_{min}$ from $(w_1, w_2)$. A value of zero means that the player is performing optimally according to our premise; a value $> 0$ means that $v_i$ is performing suboptimally—$v_i$ is making more letter edits (expending greater effort) than is required by the data. For each $v_i$, over all $n_w$ words formed by $v_i$, we compute an average $\Delta d^L_{i,ave} = (1/(n_w - 1)\Sigma_{j=1}^{n_w-1}[\Delta d^L_j]$. We rank the players $v_i \in V$ by $\Delta d^L_{i,ave}$, least to greatest, and produce five equi-sized player groups, such that the first group (i.e., the top 20% of players) performs closest to optimal and the last group (i.e., the bottom 20% of players) performs farthest from optimal. We label these five groups of players as $P_1$ through $P_5$, with $P_1$ (resp. $P_5$) corresponding to the top (resp., bottom) 20% of players. We take this player-centric approach because we want to produce agent models based on individual player and groups of players' behaviors.

Second, for each of the five groups of players, we plot all data points $(x, y) = (d^L_{min}, d^L_{i,act}(w_1, w_2))$ for each person in that group, in Figure 6. In each plot, for each $d^L_{min} (\equiv d^L_{opt})$ on the x-axis, there is a range of $d^L_{i,act}(w_1, w_2)$ for all $v_i$ in a particular 20% bin. The number of observations for the plot is given at the top of the plot and the number of observations for each box is also specified. If we break the players down into 10% bins, and confine our observations to $1 \le d^L_{min} \le 3$, it is observed that the top 30% of players perform such that the median value of $d^L_{i,act}(w_1, w_2)$ equals $d^L_{min}$. That is, in a median sense, these top 30% of players form word $w_2$ such that $d^L_{i,act}(w_1, w_2) = d^L_{min}$, and hence $w_2$ is formed optimally. Moreover, if we look at the top 80% of players, then $d^L_{min} \le d^L_{i,act}(w_1, w_2) \le d^L_{min} + 1$. These data for $|C^W| = 5000$ substantiate our premise that players form word $w_2$ based on $d^L$. Although not shown, similar results are generated for $|C^W| = 1000$, 2000, 3000, and 4000, if we take these sets as the 1000, 2000, 3000, and 4000 most frequently used words in the original corpus of 5000 words.

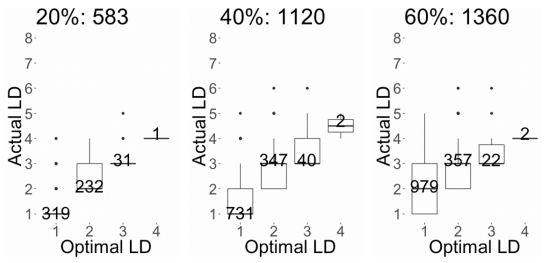


**Figure 6: Experimental data using word corpus $|C^W| = 5000$, with three plots of $d^L_{opt}$ vs. $d^L_{i,act}$. Each plot corresponds to a grouping of players by 20% bins of player performance in forming words according to $d^L$, and represents, in turn, $P_j$, $j \in \{1, 2, 3\}$. Data for $P_4$ and $P_5$ are not shown. For each $P_j$, the $d^L_{opt}$ is computed as the minimum possible $d^L_i(w_1, w_2)$, given $w_1$ in an experiment, for all $w_2 \in W_i^{ih} \subseteq C^W$. This value is plotted against the actual $d^L$, $d^L_{i,act}(w_1, w_2)$, based on the actual word $w_2$ formed in the experiments. Numbers are numbers of observations in the data. If $d^L_{i,act}(w_1, w_2) = d^L_{opt}$, then the experimental data correspond exactly with the model.**

Third, for a given $w_1$ and $d^L_{i,act}$, $W_i^{ih}(w_1, d^L_{i,act}) \subseteq W_i^{ih}$ is the candidate set of words that $v_i$ can form as $w_2$. The issue is how players extract a particular word from $W_i^{ih}(w_1, d^L_{i,act})$ as $w_2$. Figure 7 provides the answer. For each $v_i$, we rank the words in $W_i^{ih}(w_1, d^L_{i,act})$ in decreasing order of frequency of occurrence (which is obtained from the word corpus itself), such that the first ranked word is the most frequently used word. This plot shows the

number of times the chosen word $w_2$ is of a particular rank. It is clear that players select $w_2$ based on the frequency of the word's use, e.g., the top-ranked word is selected almost 700 times from the corpus. This result also holds over different corpus sizes from 1000 to 5000 words.

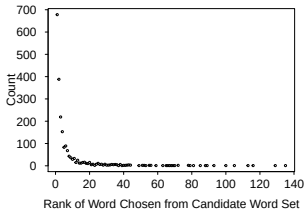


**Figure 7: Experimental data for $|C_i^W| = 5000$ showing ranks of words formed by players. Lesser rank means higher word frequency from corpus. Players most often choose words with lesser rank.**

**Remark**: These data analyses substantiate our claim that our models are explanatory. Here, the data are consistent with the explanation that humans reason about what word to form using LD and word frequency (familiarity).

**Remark**: It is emphasized that players in the experiments are not given a word corpus, frequency of letter use, $d^L$ concepts and values, etc. Our construction and procedures presented here are our representation of the mental reasoning processes that players engage in, resulting in human behavior in the form of detailed actions. In experiments, players are only given letters and the ability to share them. This remark holds for the next two models, too.

**Algorithm for form word.** See the algorithm in Figure 8, which follows the description just given. Figure 9 provides examples of $\mathcal{D}^{d^L}(|C_i^W|, b_i^{wf}, d^L_{min})$, for the top 20% of players referred to in the algorithm. It is from distributions such as these that $d^L_{i,act}$ (for an agent action in a simulation) is sampled, based on $d^L_{min}$.

## 5.3 Player Action: Request Letter

**Basic premise and key concepts.** Our goal is to uncover a process that explains how players select the next letter to request from their neighbors. Our premise is that player $v_i$ will select the next letter to request as the letter from the set of candidate neighboring letters $L_i'$ that produces the greatest increase in the number of words that a player can form with the letters in-hand. The key idea is to examine each candidate letter $\ell$ and determine the number of words $|W_i^{ih\ell}|$ that can be formed with existing letters in $L_i^{ih}$ and the requested letter combined (this word set is $W_i^{ih\ell}$), rank these letters in decreasing order of $|W_i^{ih\ell}|$, and select the letter to request based on this ranking. This greedy process—in the sense of selecting the best letter, one at a time—gives a player the best opportunity to form new words. Note that as more letters have been requested and received, $|L_i'|$ decreases. We now provide the evidence for behavior that is aligned with this premise.

**Data analysis.** First, we rank all players by their performance in requesting letters, as follows. For each player $v_i$, and for each candidate letter $\ell \in L_i'$ requested, there is accompanying $L_i^{ih\ell}$, and we compute $|W_i^{ih\ell}|$. We rank these letters $\ell$ in decreasing size of $|W_i^{ih\ell}|$, so the top-ranked letter generates the most new words. With this ranking of letters, we identify the rank $r_{i,act}$ of the letter $\ell_{i,act}$ that was actually chosen in the experiment. For each $v_i$, we then compute an average letter rank $r_{i,ave} = (1/n_{rh})\Sigma_{j=1}^{n_{rh}}[r_{i,act}]$

**Input:** Agent $v_i \in V$. Agent word-forming aptitude $b_i^{wf}$. Word corpus or vocabulary $C_i^W$ for $v_i$. Letters in-hand $L_i^{ih}$. Most recent word formed by $v_i$, $w_1$. Words $W_i^f$ formed up to now by $v_i$. Distribution $\mathcal{D}^{d^L}(|C_i^W|, b_i^{wf}, d^L_{min})$ of $d^L_{i,act}$ for a specific $d^L_{opt}$. Distribution $\mathcal{D}^{wr}$ of word ranks.

**Output:** Next word $w_2$ that $v_i$ forms, if any.

**Steps:**

(1) From $L_i^{ih}$ and $C_i^W$, construct the set $W_i^{ih}$ of words that $v_i$ can form. Let $V_H \equiv W_i^{ih}$.
(2) Remove from $V_H$ (= $W_i^{ih}$) all words in $W_i^f$ except $w_1$; cannot form same word twice.
(3) **If** $W_i^{ih} = \emptyset$, return no word. Terminate this algorithm.
(4) Form a word proximity network (WPN). Specifically, form a clique graph $H(V_H, E_H)$. Let $E_H = \{\{w_i, w_j\} \mid \forall w_i, w_j \in V_H, i \neq j\}$ be the undirected edge set. Each edge $\{w_i, w_j\}$ is labelled with Levenshtein distance $d^L(w_i, w_j)$.
(5) Construct the word set $W_{v_i}$ consisting of words $w_j$ for $w_i = w_1$ where $\{w_1, w_j\} \in E_H$.
(6) Partition $W_{v_i}$ into subsets of words, where each subset of words has the same $d^L$ from $w_1$. Let these subsets be $W^k$, with $1 \leq k \leq q$, where $q$ is the maximum $d^L$ label of any edge incident on $w_1$.
(7) Let $d^L_{min}$ be the minimum value of $k$ such that $W^k \neq \emptyset$.
(8) With the three-tuple $(|C_i^W|, b_i^{wf}, d^L_{min})$, identify the correct distribution $\mathcal{D}^{d^L}(|C_i^W|, b_i^{wf}, d^L_{min})$ from which to draw an actual $d^L$, $d^L_{i,act}$, for $v_i$. Sample $d^L_{i,act}$. See Figure 9.
(9) From the subset (partition) of words $W^{d^L_{i,act}} \subseteq W_{v_i}$, order the words in decreasing order of popularity or use, obtained directly from the input corpus $C_i^W$. This ranks the words.
(10) From the distribution $\mathcal{D}^{wr}$ of word ranks (formed from data such as those in Figure 7), draw a rank $r_i$ of a word.
(11) The word to form is $w_2 \in W^{d^L_{i,act}}$ that has rank $r_i$. Add $w_2$ to $L_i^{ih}$.
(12) Return $w_2$.

**Figure 8: Steps of the Algorithm Form Word, invoked from Figure 5. This algorithm returns a word that an agent forms.**

where $n_{rh}$ is 1/2 of the total number of letter requests. We use only the first 1/2 of requests because as $|L_i'|$ decreases, the selected rank and the top-ranked letters will be more closely aligned because there are so few letters left; hence, in order to not bias the results, we use only the first 1/2 of letter requests. We rank the players $v_i$ from smallest to largest value of $r_{i,ave}$, and divide the players into five equi-sized groups, $P_1$ through $P_5$, where $P_1$ (resp., $P_5$) is the set of players with the least (resp. greatest) average rank. This partitioning is to ensure a sufficient number of observations for each bin. This means that the players in $P_1$ perform most closely to the idealized model directly above. Again, we partition based on player behaviors because we want to develop agent behaviors based on player behavior.

Second, we analyze each $P_j$, $j \in \{1, 2, 3, 4, 5\}$, separately, as follows. We take each $v_i \in P_j$, note each $r_{i,act}$ corresponding to each letter request in the first 1/2 of requests, and count the number of occurrences of the ranks of each requested letter, and summing

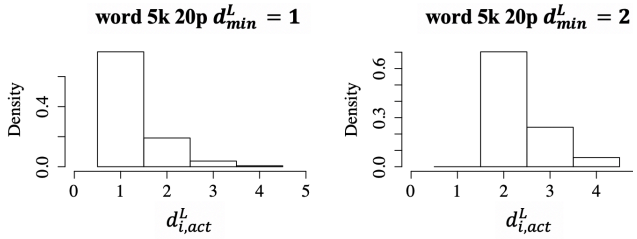


**Figure 9: For $|C_i^W| = 5000$, and for $b_i^{wf} = P_1$, there are two plots for the distribution $\mathcal{D}^{d^L}(|C_i^W|, b_i^{wf}, d^L_{min})$ of $d^L_{i,act}$ for a fixed $d^L_{opt} = d^L_{min} = 1$ (Left) and $d^L_{opt} = d^L_{min} = 2$ (Right). Values for 3 and $> 3$ are not shown. For a given $d^L_{opt}$ computed in Figure 8, the appropriate distribution is sampled to obtain $d^L_{i,act}$ for $v_i$. These distributions are formed from the data in Figure 6.**

the counts over all players. Results are shown in Figure 10. Note that for each 20% grouping, the number of occurrences of a selected rank generally increases as the rank decreases, though the effect is sometime less pronounced for some cases. We claim that the data support our premise, i.e., our model explains the data. That is, players select letters to request that generate the greatest increase in the number of words that they can form.

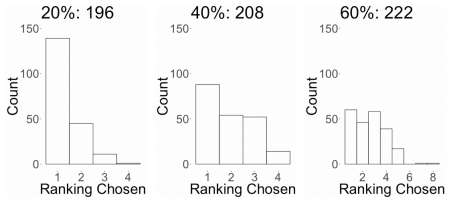


**Figure 10: Experimental data for the 5000-word corpus, with one plot for each value of the aptitude for letter request $b_i^{req}$ of $P_j$, $j \in \{1, 2, 3\}$, $P_4$ and $P_5$ are not shown. For each aptitude, the frequency of the rank of the chosen letter is plotted. Letter ranks are generated in decreasing order of the number of words that can be formed when a candidate letter is joined with $L_i^{ih}$. These data show that players most often choose letters with lower rank in all five bins, meaning that they choose letters that can form relatively more words.**

**Algorithm for request letter.** See the algorithm in Figure 11. Figure 12 provides a set of distributions for $\mathcal{D}^{\ell r}(|C_i^W|, b_i^{req}, r_{num})$, used in the algorithm, for the specific case of $|C_i^W| = 5000$ words.

## 5.4 Player Action: Reply to Letter Requests

Unlike the previous two models, this model is purely data-driven because there is no clear mechanistic model for this action.

**Basic premise and key ideas.** The goal is to produce a model that explains how players respond to letter requests from their neighbors. The basic premise is that players can be partitioned into categories of behavior, such as those players that always reply to all current letter requests in an uninterrupted series of replies, once they choose to reply to letter requests. The key ideas are that for each category, we need to determine how many replies to letter

**Input:** Agent $v_i \in V$. Agent letter requesting aptitude $b_i^{req}$. Word corpus or vocabulary $C_i^W$ for $v_i$. Letters in-hand $L_i^{ih}$. The set $L_i'$ of letters that $v_i$'s neighbors were initially assigned that $v_i$ has not yet requested; this is the candidate set of letters to request. The request number $r_{num}$ for $v_i$. Distributions $\mathcal{D}^{\ell r}(|C_i^W|, b_i^{req}, r_{num})$ of letter ranks.

**Output:** Next letter $\ell^*$ that $v_i$ requests, if any.

**Steps:**

(1) Instantiate a map $m(\ell, W_i^{ih\ell})$.
(2) **for each** $\ell \in L_i'$ **do**
  (a) Form letter set $L_i^{ih\ell} = L_i^{ih} \cup \{\ell\}$.
  (b) Form all words $W_i^{ih\ell}$ from corpus $C_i^W$ with all letters in $L_i^{ih\ell}$.
  (c) Add $(\ell, W_i^{ih\ell})$ to $m(\ell, W_i^{ih\ell})$.
(3) **If** each word set $W_i^{ih\ell} = \emptyset$, return no letter; terminate this algorithm.
(4) Rank the letters in $\ell \in L_i'$ according to decreasing values of $|W_i^{ih\ell}|$ using $m(\ell, W_i^{ih\ell})$. Let $r(\ell)$ be the rank of $\ell$.
(5) *Comment*: The rank in absolute numbers of words is the same as the rank for the increase in numbers of new words that can be formed.
(6) Determine the rank $r_{i,act}$ of the letter to select for requesting, as follows. Form the tuple $(|C_i^W|, b_i^{req}, r_{num})$, and select a distribution $\mathcal{D}^{\ell r}(|C_i^W|, b_i^{req}, r_{num})$ that is formed from data such as those in Figure 12. Sample from this distribution and call the sample $r_{i,act}$.
(7) Select the letter $\ell^*$ such that $r(\ell^*) = r_{i,act}$. Remove $\ell^*$ from $L_i'$. Return $\ell^*$.

**Figure 11: Steps of the Algorithm Request Letter, invoked from Figure 5. This algorithm returns a letter an agent requests.**

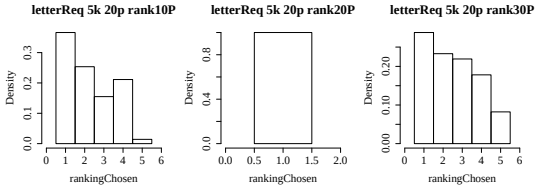


**Figure 12: For $|C_i^W| = 5000$, $b_i^{req} = P_1$, and for a particular request number $r_{num}$, there is a plot of the distribution $\mathcal{D}^{\ell r}(|C_i^W|, b_i^{req}, r_{num})$ of $r_{i,act}$. The three plots are for $r_{num} \in \{1, 2, 3\}$, in turn. $r_{num} \in \{4, 5\}$ are not shown. These distributions are formed from the data in Figure 10 and used in the algorithm of Figure 11 to obtain $r_{i,act}$ for $v_i$. There are analogous plots for each $P_j$, $j \in \{2, 3, 4, 5\}$ and for different vocabularies.**

requests are made uninterrupted (i.e., contiguously), and for each number of letter replies, the time duration over which these letter replies are made.

**Data analysis.** There are four observed player behaviors in responding to letter requests: (1) a player replies to all letter requests in its reply buffer when she takes the reply action (FB case); (2) a player always replies to less than the full reply buffer (LTFB case); (3) a player sometimes replies as a player in (1) and sometimes as a player in (2) (Mixed case); and (4) a player never responds to any letter requests (NR case). See Table 1. For each category, the

fractions of players over all games that exhibit these behaviors are provided. The last two rows indicate whether we need to analyze the number of contiguous replies and whether we need to analyze the time over which these replies take place.

**Table 1: The four categories of player behavior for taking the action reply to letter requests of neighbors. See text for category names.**

| **Player Category** | FB | LTFB | Mixed | NR |
|---|---|---|---|---|
| **Frac. Players** | 0.664 | 0.0636 | 0.176 | 0.0964 |
| **Need Reply Size** | no | yes | yes | no |
| **Need Reply Time** | yes | yes | yes | no |

First, we address the number letter requests that are replied to for the action reply to letter requests. We require data for only LTFB and Mixed categories because FB means that all letter requests in the buffer are replied to and NR means players never reply to any letter requests (see Table 1).

Second, we must know how long it takes for players to contiguously reply to more than one request. As Table 1 (last row) denotes, we must obtain these data for categories FB, LTFB, and Mixed. For each of these categories, we break the data into the times to reply to 2, 3, 4, 5, and $\geq 6$ letter requests. Since there is far more data for FB than for LTFB and Mixed, we combine the LTFB and Mixed data for one distribution. Sample distributions are not provided owing to space limitations.

Third, we evaluate how multiple replies are made. We find that for the FB category, the individual letter replies are roughly evenly spaced in time, whereas for LTFB and Mixed categories, the reply times to requests are more random.

**Algorithm for reply to (letter) request.** See the algorithm in Figure 13.

**Remark**: In these various algorithms, elements of sets are returned, or a distribution corresponding to particular inputs is sampled. In some cases, there are no data for some condition. For these types of situations, we implement a recursive search technique to sample from the distribution or set with the closest set of inputs.

**Input:** Agent $v_i \in V$. Agent reply to (letter) request aptitude $b_i^{rpl}$. Word corpus or vocabulary $C^W$ for $v_i$. Graph of all players and comm. channels in simulation $G(V, E)$. Letters in hand $L_i^{ih}$. The set $L_i'$ of letters that $v_i$'s neighbors were initially assigned that $v_i$ has not yet requested; this is the candidate set of letters to request. Most recent word formed by $v_i$, $w_1$. For reply aptitude $b_i^{rpl}$ =LTFB or Mixed, the distribution for sampling the number of requests to contiguously reply to $n^{rpl}$, $\mathcal{D}^{n^{rpl}}(b_i^{rpl})$. For aptitudes $b_i^{rpl}$ =FB, LTFB, or Mixed, the distributions for time duration over which the $n^{rpl}$ are made, $\mathcal{D}^{\Delta t}(b_i^{rpl}, n^{rpl})$.

**Output:** The number $n^{rpl}$ of contiguous replies to make, and the time at which each letter is replied. (Letters are replied to in FIFO order.)

**Steps:**

(1) The ordering of multiple replies to letter requests (i.e., for $n^{rpl} > 1$) are in FIFO (first in, first out) order.
(2) Determine the number $n^{rpl}$ of replies to letter reqeusts that will be replied to contiguously by $v_i$, as follows. **If** $b_i^{rpl}$ =FB, then the $n^{rpl}$ is the size of the reply buffer. **Else If** $b_i^{rpl}$ =LTFB or Mixed, then the $n^{rpl}$ is determined by sampling $\mathcal{D}^{n^{rpl}}(b_i^{rpl})$. **Else If** $b_i^{rpl}$ =NR, then the $n^{rpl} = 0$ because this category of reply behavior does not reply to any request; terminate this algorithm.
(3) Determine the total time duration $\Delta t$ over which all $n^{rpl}$ are made, by sampling $\mathcal{D}^{\Delta t}(b_i^{rpl}, n^{rpl})$.
(4) Determine how the $n^{rpl}$ replies to (letter requests) are spaced out over the interval $\Delta t$, as follows. **If** $b_i^{rpl}$ =FB, then the $n^{rpl}$ replies are spaced uniformly in time interval $[t, t + \Delta t]$, where the current time is $t$. **Else If** $b_i^{rpl}$ =LTFB or Mixed, then the $n^{rpl}$ replies are spaced uniformly at random in time interval $[t, t + \Delta t]$.
(5) With FIFO ordering, these steps provide an ordered list $R$ of (time, reply to letter request) elements.
(6) Execute these letter replies at the specified times, with "thinking" or "idle" action at intermittent times.
(7) Return this ordered list $R$.

**Figure 13: Steps of the Algorithm Reply To Request, invoked from Figure 5. This algorithm returns a time-ordered list of letter replies to execute.**

## 6 MODEL EVALUATIONS

**Evaluation Procedure.** We evaluate our composite ABM by comparing experimental data with predictions from the component models of Figure 4 for form words and request letters.

For each experiment (game) and each player in each game, we have a time history trace of each of their actions, so that we have the complete state of each player in time. For each player in each game, when one of the three above actions is taken, we take the state of the player and execute the appropriate algorithm among Figures 8, 11, and 13. Moreover, for each of the three component models, we implemented a *null* model, in turn, where a word is selected uniformly at random from $W_i^{ih}$, a letter to request is selected uniformly at random from $L_i'$, or the number $n^{rpl}$ of replies is chosen uniformly at random. Thus, we have for each $v_i$ and each action type, three *kinds* of values for that action: (*i*) the experimentally measured value, (*ii*) the value generated according to the appropriate algorithm of Section 5, and (*iii*) the value from the appropriate null model.

Our measures are as follows, and we compute each of these for each kind of data. For action "form word," we compute $d_{i,act}^L(w_1, w_2)$ for successive words. For action "request letter," we compute the rank $r$ of the letter chosen. For action "reply to (letter) request," we compute $n^{rpl}$.

**Evaluation Results.** For each of these kinds of data, we generate sequence histories of values, with error bars, for comparisons, and we compute $\alpha$ values that average these results. For example, for $v_i$, and word formation, we have $\alpha_i^{wf} = 1/(n^{wf} - 1)\, \Sigma d_i^L(w_1, w_2)$, for successive words $w_1$ and $w_2$, for each of the three kinds of data. (Note: $w_1$ is the same for all three types of data.) Similarly for letter requests, we calculate $v_i$'s average rank of requested letters as $\alpha_i^{req} = 1/(n_i^{req})\, \Sigma_{j=1}^{n_i^{req}} r_j$ and for replies to letter requests, we compute the average number of letter replies per reply action as $\alpha_i^{req} = 1/(n_i^{rpl})\, \Sigma_{j=1}^{n_i^{rpl}} n^{rpl}$. We then compute the differences in corresponding $\alpha$ values between the model (labeled "opt" for optimal)

and the experimental data, and the differences in $\alpha$ values between the null model and the experimental data. Figure 14 shows box plots of these differences for words formed and or letter requests. We see that the box plots for the optimal or true model have lesser variance and lesser median values than those for the null model, indicating greater conformance with experiments.

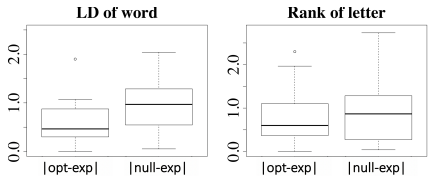


**Figure 14: (Left) Box plots for the LDs of formed words, comparing the model of Section 5.2 to a null model. (Right) Box plots for the ranks of letters requested, comparing the model of Section 5.3 with the null model. See text for details of data generation. The models of Section 5 are in better agreement with experimental data than null models.**

## 7 CASE STUDIES

We study the effects of input parameters on group anagram games simulation results. We use a game configuration $G(V, E)$ consisting of six players that form a circle, with each player having two neighbors. We run $n_{runs} = 30$ runs or simulation instances, with $t_{max} = 300$ seconds, as in the experiments. We use the 5000-word corpus $C^W$, with initial letter assignments given in Table 2. We systematically vary the aptitudes of players in forming words $b_i^{wf}$, in requesting letters $b_i^{req}$, and in replying to letter requests $b_i^{rpl}$. See Table 3. Recall that these aptitudes correspond to the bins $P_1$ through $P_5$ into which we partitioned the game players in experiments. Note that these are purely simulation studies and are not tied to the experiments. The goal is to demonstrate that the models alone provide insights into human behavior and game results.

**Table 2: Initial letter assignments in simulations for six players arranged as 2-regular graph.**

| **Player #:** | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Init. Ltrs:** | b, a, t | m, e, n | l, u, t | s, o, p | h, u, g | r, i, e |

**Table 3: Parameters that are systematically varied in the simulations. These aptitude ($b_i^{wf}$, $b_i^{req}$, $b_i^{rpl}$) settings are the same for all agents in a simulation.**

| **Sim. No.** | $b_i^{wf}$ | $b_i^{req}$ | $b_i^{rpl}$ | **Sim. No.** | $b_i^{wf}$ | $b_i^{req}$ | $b_i^{rpl}$ |
|---|---|---|---|---|---|---|---|
| 1 | $P_1$ | $P_1$ | FB | 6 | $P_5$ | $P_5$ | LTFB |
| 2 | $P_2$ | $P_2$ | FB | 7 | $P_5$ | $P_5$ | NR |
| 3 | $P_3$ | $P_3$ | FB | 8 | $P_4$ | $P_5$ | FB |
| 4 | $P_4$ | $P_4$ | FB | 9 | $P_5$ | $P_4$ | FB |
| 5 | $P_5$ | $P_5$ | FB | – | – | – | – |

Figure 15 (left) shows the average number of interactions (requests sent, replies received, requests received, replies sent) and the average number of words formed per player for the first five simulation numbers (sim. no.) of Table 3. There is a drop-off in performance in going from $b_i^{wf} = b_i^{req} = P_1$ to $P_5$, for fixed $b_i^{rpl}$ =FB. We observe that decreasing the letter request aptitude $b_i^{req}$ and the word formation aptitude $b_i^{wf}$ decreases the quality of letters requested and hence the number of words that can be formed.

To determine how $b_i^{rpl}$ affects performance, we plot in Figure 15 (right) results from simulation numbers 5, 6, and 7 of Table 3. Using $b_i^{wf} = b_i^{req} = P_5$ as a reference, there is a large decrease in reply interactions going from $b_i^{rpl}$ =FB to $b_i^{rpl}$ =LTFB. Surprisingly, $b_i^{rpl}$ =LTFB shows a behavior comparable to $b_i^{rpl}$ =NR.

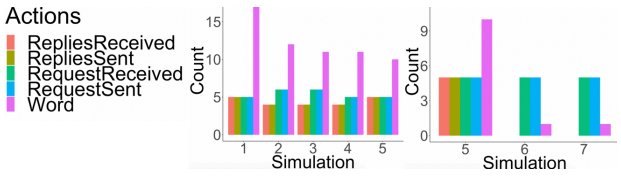


**Figure 15: (Left) Simulation results for Sim. nos. 1 through 5 of Table 3. The average number of words formed per player drops in going from $b_i^{wf} = b_i^{req} = P_1$ to $P_5$, for fixed $b_i^{rpl}$ =FB. (Right) Simulation results for Sim. nos. 5, 6, and 7 of Table 3. Using $b_i^{wf} = b_i^{req} = P_5$ as a baseline, these results show a precipitous drop-off in replies to letter requests, and to words formed, in going from $b_i^{rpl}$ =FB to $b_i^{rpl}$ =LTFB. $b_i^{rpl}$ =LTFB shows a behavior comparable to $b_i^{rpl}$ =NR.**

Although plots are not shown here, comparisons among rows 4, 8, and 9 show that decreasing the letter request aptitude $b_i^{req}$ from $P_4$ to $P_5$ (row 4 vs. row 8), is sufficient to decrease average words formed. However, decreasing word formation aptitude $b_i^{wf}$ from $P_5$ to $P_4$ (row 4 vs. row 9) does not decrease average words formed. These results demonstrate that the letter request aptitude is more important for increasing the number of words formed.

## 8 SUMMARY AND LIMITATIONS

We have presented mechanistic and data-driven models for representing the reasoning and actions of humans in networked group anagram games. Our contributions are in Section 1.3. We would like to (a) conduct more experiments at greater $k$ and (b) control the quality of letters given to players in the game.

## REFERENCES

[1] [n. d.]. Word frequency data: Corpus of Contemporary American English. https://www.wordfrequency.info/free.asp. Accessed: 2018-11-16.
[2] Robert Ackland and Mathieu O'Neil. 2011. Online collective identity: The case of the environmental movement. *Social Networks* 33 (2011), 177–190.
[3] Gary S. Becker. 1976. *The economic approach to human behavior.* University of Chicago Press Chicago.
[4] Alisa Bokulich. 2011. How Scientific Models Can Explain. *Synthese* 180 (2011), 33–45.
[5] C. Bram Cadsby et al. 2007. Sorting and Incentive Effects of Pay for Performance: An Experimental Investigation. *Acad. of Mgmt. Jour.* 50 (2007), 387–405.
[6] Valerio Capraro. 2013. A Model of Human Cooperation in Social Dilemmas. *PLoS One* 8 (2013), e72427–1–e72427–6.
[7] Damon Centola. 2011. An Experimental Study of Homophily in the Adoption of Health Behavior. *Science* 334 (2011), 1269–1272.
[8] Gary Charness, Ramon Cobo-Reyes, et al. 2014. Identities, selection, and contributions in a public-goods game. *Games and Economic Behavior* (2014), 322–338.
[9] Gary Charness, Peter Kuhn, and Marie Claire Villeval. 2011. Competition and the Ratchet Effect. *Journal of Labor Economics* 29 (2011), 513–547.
[10] D. L. Chen et al. 2016. oTree–An open-source platform for laboratory, online and field experiments. *J. Beh. & Exp. Fin.* 9 (2016), 88–97.
[11] W. L. Davis and D. E. Davis. 1972. Internal-external control and attribution of responsibility for success and failure. *Journal of Personality* 40 (1972), 123–136.
[12] Morton Goldman, Joseph W Stockbauer, et al. 1977. Intergroup and intragroup competition and cooperation. *Journal of Experimental Social Psychology* (1977).

[13] Brian D. Haig. 2005. An Abductive Theory of Scientific Method. *Psychological Methods* 10 (2005), 371–388.
[14] Jake M. Hofman, Amit Sharma, and Duncan J. Watts. 2017. Prediction and explanation in social systems. *Science* 355 (2017), 486–488.
[15] Julie Jebeile. 2018. Explaining with Simulations: Why Visual Representations Matter. *Perspectives on Science* 26 (2018), 213–238.
[16] Julie Jebeile and Ashley Graham Kennedy. 2015. Explaining with Models: The Role of Idealization. *Int. Studies in the Philosophy of Science* 29 (2015), 383–392.
[17] Daugelaite Jurate et al. 2013. An Overview of Multiple Sequence Alignments and Cloud Computing in Bioinformatics. *ISRN Biomathematics* 2013 (2013), 14.
[18] Michael Kearns, Stephen Judd, et al. 2012. Behavioral Experiments on a Network Formation Game. In *Economics and Computation (EC)*. 690–704.
[19] V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10 (Feb. 1966), 707.
[20] Winter Mason and Duncan J. Watts. 2012. Collaborative learning in networks. *Proceedings of the National Academy of Sciences* 109, 3 (2012), 764–769.
[21] Nam P. Nguyen, Guanhua Yan, et al. 2012. Containment of Misinformation Spread in Online Social Networks. In *Web Science Conference*. 213–222.
[22] Derek O'Callaghan et al. 2013. Uncovering the Wider Structure of Extreme Right Communities Spanning Popular Online Networks *(WebSci '13)*. ACM, 276–285.
[23] Timothy J. Owens. 2006. Self and Identity. In *Handbook of Social Psychology*. 205–232.
[24] Francesca Polletta and James M. Jasper. 2001. Collective Identity and Social Movements. *Annual Review of Sociology* 27 (2001), 283–305.
[25] Y. Ren, V. Cedeno-Mieles, et al. 2018. Generative Modeling of Human Behavior and Social Interactions Using Abductive Analysis. In *ASONAM*. 413–420.
[26] Samuel Spaulding, Huili Chen, et al. 2018. A Social Robot System for Modeling Children's Word Pronunciation: Socially Interactive Agents Track. In *AAMAS*.
[27] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science* 12 (1988), 257–285.
[28] Iddo Tavory and Stefan Timmermans. 2014. *Abductive Analysis: Theorizing Qualitative Research*. University of Chicago Press, Chicago and London.
[29] Duncan J. Watts. 2017. Should social science be more solution-oriented? *Nat. Hum. Behav.* (2017), 1–5.