

ORIGINAL ARTICLE

Robust coordination in adversarial social networks: From human behavior to agent-based modeling

Chen Hajaj^{1,2*} , Zlatko Joveski³ , Sixie Yu⁴ and Yevgeniy Vorobeychik⁴ 

¹Department of Industrial Engineering and Management, Ariel University, Ariel, Israel, ²Cyber Innovation Center, Ariel University, Ariel, Israel, ³Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235, USA (e-mail: jovzlatko@gmail.com) and ⁴Department of Computer Science and Engineering, Washington University, St. Louis, MO 63130, USA (e-mails: sixie.yu@wustl.edu, yvorobeychik@wustl.edu)

*Corresponding author. Email: chenha@ariel.ac.il

Action Editor: Fernando Vega-Redondo

Abstract

Decentralized coordination is one of the fundamental challenges for societies and organizations. While extensively explored from a variety of perspectives, one issue that has received limited attention is human coordination in the presence of adversarial agents. We study this problem by situating human subjects as nodes on a network, and endowing each with a role, either regular (with the goal of achieving consensus among all regular players), or adversarial (aiming to prevent consensus among regular players). We show that adversarial nodes are, indeed, quite successful in preventing consensus. However, we demonstrate that having the ability to communicate among network neighbors can considerably improve coordination success, as well as resilience to adversarial nodes. Our analysis of communication suggests that adversarial nodes attempt to exploit this capability for their ends, but do so in a somewhat limited way, perhaps to prevent regular nodes from recognizing their intent. In addition, we show that the presence of trusted nodes generally has limited value, but does help when many adversarial nodes are present, and players can communicate. Finally, we use experimental data to develop computational models of human behavior and explore additional parametric variations: features of network topologies and densities, and placement, all using the resulting data-driven agent-based (DDAB) model.

Keywords: coordination; social networks; adversaries; agent-based modeling

1. Introduction

Coordination is one of the fundamental problems faced by teams, organizations, and societies. Such coordination problems are often decentralized and involve limited local information and interaction, with locality naturally captured by a network structure. A prominent example for the special case of consensus is blockchain, which enables verifiable decentralized transactions (Narayanan et al., 2016).

Considerable prior research has been devoted to understanding and modeling human behavior in networked coordination settings such as networked consensus (Kearns et al., 2009; Judd et al., 2010; Kearns, 2012; Vorobeychik et al., 2017), coloring (Matthew et al., 2009; Judd et al., 2010), bargaining (Chakraborty et al., 2010), and social dilemma games (Gracia-Lázaro et al., 2012; Leibbrandt et al., 2015), among others. However, decentralized coordination problems often take place in adversarial predicaments. For example, organizations attempting to coordinate on a strategy may also compete with other organizations (legal and illegal), and coordination in combat mission planning and execution inherently faces adversarial entities in the form of enemy

combatants. Moreover, adversaries often attempt to exert their influence covertly such as by bribing insiders, taking control of network nodes through cyberattacks, and spreading malicious influence tacitly through social networks, for example, by means of fake news (Alon et al., 2015). Consequently, an important consideration in decentralized coordination is resilience to adversarial tampering with the process. While much prior research has been devoted to the study of robust coordination protocols, these rely on simple stylized models of individual behavior (Abbas et al., 2014; Bracha & Toueg, 1983; LeBlanc & Koutsoukos, 2012; LeBlanc et al., 2013). However, many settings feature humans in the loop who play an important role in reaching consensus. Surprisingly, the question of human behavior in adversarial coordination settings has received little prior attention.

We investigate the problem of decentralized consensus on networks in the presence of adversarial nodes, first using human subject experiments with 556 participants, and subsequently through the *data-driven agent-based modeling (DDABM)* methodology (Zhang et al., 2016). Our experiments focus on two design factors: allowing neighboring nodes to communicate and embedding a small set of trusted nodes in the network. While communication has been a major subject of inquiry in prior research (Demichelis & Weibull, 2008; Ellingsen & Ostling, 2010; Miller & Moser, 2004; Cooper et al., 1992), recent research suggests that communicating solely among network neighbors has limited value in facilitating consensus (Vorobeychik et al., 2017). On the other hand, much prior research, using stylized models of individual behavior, has argued that the presence of trusted nodes can significantly facilitate decentralized coordination (Abbas et al., 2014, 2017; Usevitch & Panagou, 2018). Our results run counter to both of these observations. First, we demonstrate that communication helps a great deal, especially as we increase the number of adversarial nodes, even though adversaries often send messages that are deliberately misleading. Second, we show that the presence of trusted nodes does not, in the aggregate, help, reinforcing the need to develop better models of individual and collective behavior in such settings. A surprising feature of adversarial behavior is that their manipulation attempts are relatively subdued: their tendency to choose colors opposing local majority is relatively weak, and they rarely communicate in a way that blatantly disagrees with their objective local state. We conjecture that this behavior is also partly strategic: since the identity of adversarial nodes is unobserved, remaining covert necessitates limiting the extent of malicious activity.

Next, we develop a data-driven agent-based model of adversarial decentralized consensus on networks, following the DDABM methodology (Zhang et al., 2016; Zhang & Vorobeychik, 2019). In DDABM, individual agent models are derived from data, and are then instantiated in an agent-based framework via features that capture behavioral interdependencies among network neighbors. For us, these serve three purposes. First, they provide further insight into individual behavior. For example, we observe that adversarial nodes clearly engage in deliberate attempts to manipulate outcomes. Second, the resulting agent-based model effectively captures our experimental observations *at the macro level*, and is quite robust to small errors in the individual agent models. Third, we demonstrate the usefulness of the derived computational platform as a means for further simulation-based investigation of the adversarial consensus problem by studying the impact of optimized network location of both trusted and adversarial nodes. We find that optimizing location is particularly beneficial for adversarial nodes, even when the placement of trusted players is similarly optimized *before* we choose where to place adversaries (i.e. in a Stackelberg fashion). Consequently, and counter to prior observations with stylized behavioral models, trusted nodes appear to have only a limited value in facilitating decentralized human consensus in adversarial settings.

Our simulation experiments consider four additional analyses: (1) optimizing behavior models, through limited change to parameters, to maximize consensus rate, (2) optimizing location of trusted and adversarial nodes within the network, and (3) systematically considering the impact of parameters of network topology, such as density, clustering, and disparity in degree distribution,

on consensus rates. Overall, we observe that small changes in model parameters have little impact on consensus rates, and optimizing location is particularly beneficial for adversarial nodes, even when they do so following a similarly optimized placement of trusted players. In addition, we find that increasing network density improves consensus rate, with and without adversaries, but also increases the value of trusted nodes. In contrast, clustering and disparity in degrees have limited impact, particularly when adversarial nodes are present.

2. Related work

Reaching coordination among a group of entities is a long-lasting problem, including some of the most researched problems: the Tragedy of the Commons (Hardin, 1968), and the Prisoner's Dilemma (Rapoport et al., 1965). Our study of networked coordination follows a number of prior efforts that investigate a variety of decentralized coordination problems on networks using human subjects methodology (Kearns et al., 2006, 2009; Chakraborty et al., 2010; Judd et al., 2010; Kearns, 2012; Matthew et al., 2009; Vorobeychik et al., 2017). The impact of communication on human coordination and cooperation has extensive, parallel literature, using both human subjects (Szamado, 2011; Richerson & Boyd, 2010; Olmstead et al., 2009) and theoretical methods (Farrell, 1987, 1988; Demichelis & Weibull, 2008; Ellingsen & Ostling, 2010; Miller & Moser, 2004). However, in most of this literature, communication is grafted on as a distinct pre-play stage; moreover, much of this literature study simple, two-player games. A recent exception is the work of Vorobeychik et al. (2017), combining both threads, but investigating only non-adversarial settings. Regarding human behavior, Coviello et al. (2012) took a more algorithmic approach to look at the matching behavior of a human in social networks. While using the same experimental design as ours, the authors focus on the case where players have to divide into pairs, when the structure of the network is unknown, with a collective goal of maximizing the number of teams. Still, similar to our work, the authors use the experimental data to produce an algorithmic model and analyze its properties by simulations. Mao et al. (2017) showed that different incentives and actions can be due to different understating of the world, or as side effect of not knowing the truth. Similar as in our world, where entities have only partial information (e.g. the state of their neighbors but not of the entire entities in the network), they can act counter to their desired goal of coordination, but still in good faith.

Robust coordination has been analyzed by several efforts, but theoretically and in simulations, using highly stylized behavior models (LeBlanc & Koutsoukos, 2012; LeBlanc et al., 2013; Zeng & Chow, 2014; Gvirts & Dery, 2021). Specifically, (LeBlanc & Koutsoukos, 2012; LeBlanc et al., 2013) focus on design of a consensus protocol that is resilient to worst-case security breaches assuming the compromised nodes have full knowledge of the network and the intentions of the other nodes. Similar to our work, Banikova et al. (2021) study Consensus under a deadline, where a group is required to reach a joint decision under a tight deadline. Still, the authors chose a different definition for the consensus as they defined it as a time-bounded iterative voting process and provide convergence guarantees as well as an extensive user study. In this work, we provide a behavioral analysis using extensive human subject experiments using a well-known crowdsourcing platform. Furthermore, we relax the assumption of full knowledge and knowledge about the intentions of different nodes in the network. Several prior efforts study the importance of trusted nodes in such settings (Abbas et al., 2014, 2017; Usevitch & Panagou, 2018). Our results suggest that stylized models used in these efforts may be limited in evaluating the efficacy of trusted nodes. An interesting result was published by Arenas et al. (2011) who created a game-theoretic model of cooperation in a social dilemma game which shows that the introduction of rare, malicious agents performing exclusively destructive actions on the other agents can induce bursts of cooperation. In contrast with their work, similar to many others, we empirically showed that an increase in the number of adversaries results in lower probability of coordination.

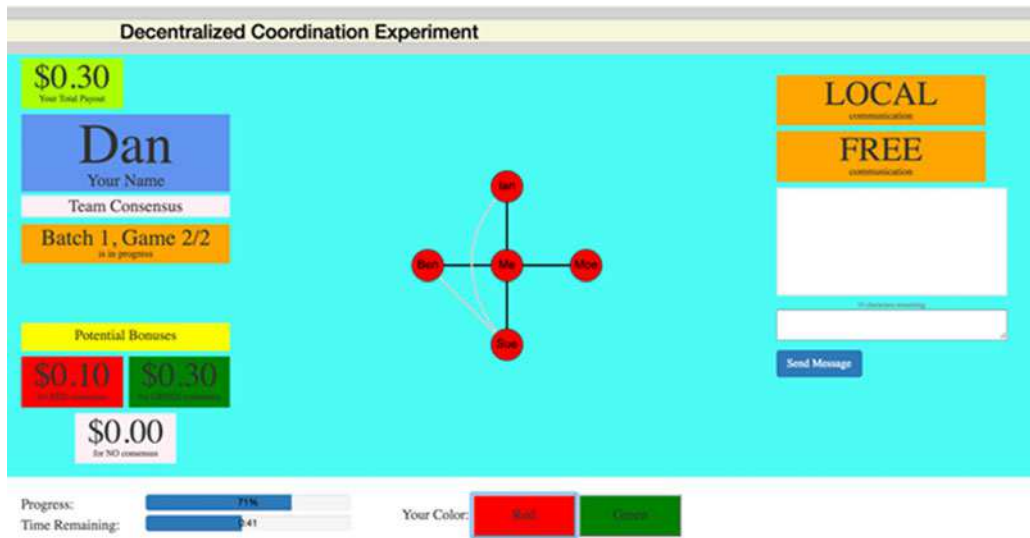


Figure 1. Top: an example graphical interface from the point of view of an experimental subject, who is represented by a node in the network. Bottom: example instances of networks, where darker colors indicate higher node degrees.

Finally, data-driven or empirical agent-based modeling has been proposed as a means of performing simulations that reliably reflect actual behavior data (Wunder et al., 2013; Zhang et al., 2016; Nay & Vorobeychik, 2016; Zhang & Vorobeychik, 2019). Shirado & Christakis (2017) performed experiments involving a networked color coordination game in which groups of humans interacted with autonomous software agents (“bots”). Similar to our work, subjects were embedded in networks of 20 nodes. In contrast to our work, in which adversarial entities are added to the network, Shirado and Christakis added three bots to their network. Our simulation-based analysis follows in the spirit of these efforts. Specifically, the authors showed that bots acting with small levels of random noise and placed in central locations meaningfully improve the collective performance of human groups, accelerating the median solution time by 55.6%.

3. Experimental methodology

3.1 General setup

We designed a human subject experiment to study *adversarial coordination on social networks*. Specifically, the experiment builds on networked consensus games (Judd et al., 2010; Kearns et al., 2012), in which a collection of players (human subjects) act as nodes on an exogenously specified graph, choosing between two colors: RED and GREEN. These games proceed for 60 seconds, with individuals able to make changes to their color choice in essentially real time. Each player has an egocentric view of the game illustrated in Figure 1, where their node is displayed at the center, and their network neighbors are shown surrounding the “Me” node, along with their color choices, as well network connections among them. Any node is displayed as white prior to actively choosing a color. The display screen also shows the time remaining in the game. Each player receives a base payment for each game played (\$0.15), as well as a bonus of \$0.20 if a global consensus on either color is reached (i.e. all nodes have the same color, at the same time). The game ends as soon as consensus is reached.¹

The game description so far replicates features from all prior experiments in networked consensus. A new feature, introduced by Vorobeychik et al. (2017), allows network neighbors to communicate through an instant message-style interface, shown on the right in Figure 1. To

facilitate such communication, when allowed, each player is assigned a three-letter name at the beginning of each game, and this name serves as their unique identifier in communicating with others. Specifically, when a player sends a message through this interface, all their immediate network neighbors receive the message (this mode of communication was termed *local* communication by Vorobeychik et al., 2017).

We made one change to this general setup, which turns out to be quite consequential. In all prior experiments, the interface featured a *progress bar*, which shows how close the overall state is to global consensus (measured by the number of nodes disagreeing with majority color). In our setting, however, such a progress bar communicates too much information, particularly when adversaries are present, and we consequently removed it (particularly since it doesn't have a clear motivation and was just a design artifact of prior experiments). As we observe below, removing the progress bar increases the importance of communication, relative to findings reported by Vorobeychik et al. (2017).

3.2 Design of adversarial consensus games

Starting with the basic experimental framework described above, we augment the experimental platform with several features in order to study how adversarial nodes impact the ability of the rest (i.e. the non-adversarial sub-network) to reach global consensus. For this purpose, we divide players into two teams: a *consensus* team and a *no-consensus* team (in our parlance, these are *adversaries*). The goal of the *consensus* team is to reach global consensus *among members of this team only* (i.e. get to a point of time in which all the members of the consensus team choose the exact same color), captured by the bonus payment structure described above. The goal of the *no-consensus* team is to prevent consensus among members of the *consensus* team, which we incentivize by paying a \$0.40 bonus to members of this team if and only if consensus fails. At the beginning of the game, each player is assigned to one of these teams, and this assignment is indicated in their view of the game (see left part of Figure 1).

We fixed the number of *consensus* players in each game to 20 to control the baseline difficulty of the task (the underlying consensus problem on networks becomes more difficult as the network size grows, other things being equal). In addition, we introduced in each game *a* *no-consensus* players, where $a \in \{0, 2, 5\}$. The value of *a* was not disclosed to the players at the beginning of a game; although an omniscient observer can infer it from the size of the network (which is $20 + a$), no player could, in fact, do this, since players could only observe their direct neighbors, and we limited the maximum degree to 15 to facilitate effective visualization.

A crucial part of our design was the invisibility of adversaries (no-consensus nodes) to others, including other adversaries, and vice versa. On the other hand, it is often possible to have a small number of known *reliable* or *trusted* nodes on the network, for example, nodes which are particularly difficult to compromise due to a high amount of investment in their security, and conventional wisdom is that such nodes can greatly facilitate consensus (Abbas et al., 2014). To allow for this, we vary the number of *visible* members of the *consensus* team (henceforth, *visible nodes*), $v \in \{0, 1, 2, 5\}$.² However, these nodes are visible only to their immediate network neighbors, highlighted by an orange circle around the corresponding nodes, as in Figure 1 for the player with an assigned name “Moe”.

3.3 Network topologies

For each game, we exogenously specify a network topology, stochastically generated from one of the three random graph models: two variations of Erdos–Renyi (ER) graphs (Erdos & Rényi, 1960), and a Barabasi–Albert (BA, also known as preferential attachment) model (Barabasi & Albert, 1999). The two variations of the ER model differ in network density: one we term ER-dense, and the other ER-sparse. The 20-node version of the ER-dense model has average

degree of 5.1, while the ER-sparse networks have an average degree of 2.6. BA networks have an average degree of 5.1 (same as ER-dense). Average degrees slightly increase when we add adversarial nodes. Figure 1 shows example networks for each of the three network generative models.

3.4 Recruiting and scheduling

We recruited subjects for the experiment using the Amazon Mechanical Turk (AMT) platform (Paolacci et al., 2010; Mason & Suri, 2012), now in common use for economic experiments with human subjects (Mason & Suri, 2012; Peled et al., 2015; Hajaj et al., 2015, 2017; Elmalech et al., 2016). Recruited subjects were directed to read detailed experiment instructions and consent to participate in the experiment (which was collected online). Once we had a large enough pool of consented subjects, we scheduled experiment sessions. An experiment session (a series of 5 practice games, followed by 50–65 actual games) was scheduled to start at a particular time. Recruited subjects were informed of the starting time at least a day in advance. We considered a subject “inactive” during an individual 60-second game if they did not make a color choice. An individual game was considered “invalid” (and removed from consideration in the subsequent analysis) if it had at least one inactive subject. A subject was considered a “dropout” after a second game of being inactive. At that point, subjects were removed from the rest of the experiment session (earlier valid games in which they were active were considered for analysis). The fact that for each experiment session, we recruited 5–10 subjects more than needed for individual games meant that we could handle dropouts, with only a fraction of games in the experiment session being invalid for analysis and without having to replace missing players with bots. Subjects participating in a given game were randomly assigned to the corresponding network instance nodes. Further, for each game, a separate network instance was generated using the corresponding model. For example, for each game with a BA network of size 20, a separate network instance was generated using the Barabasi–Albert model. While this process may occasionally lead to two identical network instances being generated, this is highly unlikely when the number of such instances is relatively small (120 BA networks of size 20 in our experiments). Subjects were not explicitly limited to playing on networks of a single type. The subset and order of network types covered in an experiment session were randomized, as was the participation queue of recruited subjects showed up in the experiment session. In practice, this meant that some player A might have played in more games with ER-dense networks and fewer games with BA networks than another player B. However, the type of the underlying network was not known by the subjects.

We systematically varied four experimental variables:

1. Number of adversaries (*no-consensus* players): $a \in \{0, 2, 5\}$.
2. Number of visible nodes (within the *consensus* team): $v \in \{0, 1, 2, 5\}$.
3. Network topology: ER-dense, ER-sparse, and BA.
4. Communication: allowed or not allowed.

The full study protocol was approved by the university’s IRB. We recruited a total of 556 participants who jointly played 1,080 games.

4. Experimental results

We now analyze the results of the experiments. Throughout, we focus on consensus rate, or proportion of games reaching global consensus on a single color among the *consensus* players, as a measure of coordination success.

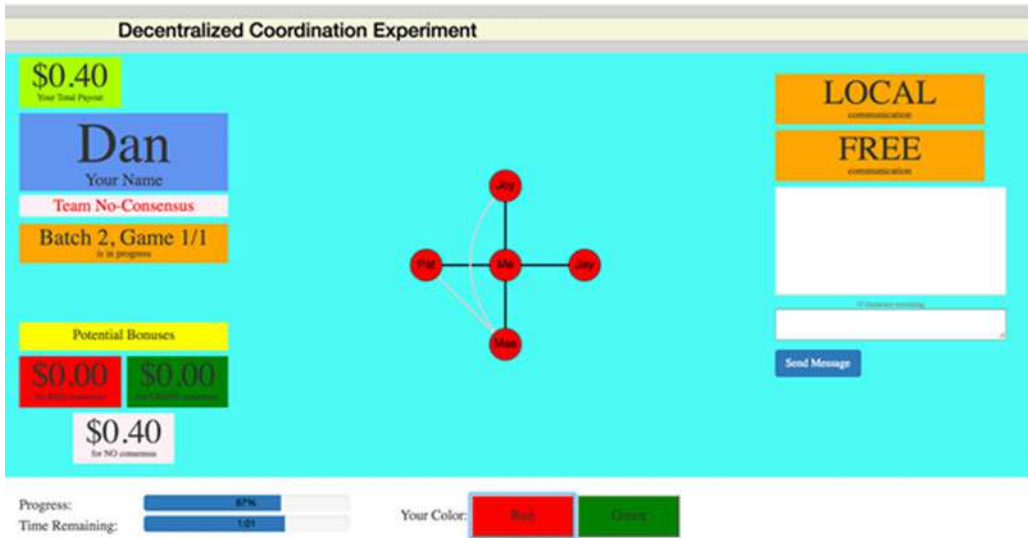


Figure 2. Impact of adversaries on the consensus rate. Left: overall consensus rate, as a function of the number of adversaries. Right: for each network distance, proportion of pairs of nodes with this distance between them who agree on a color at the end of the game.

4.1 The impact of adversarial players on consensus rate

One would naturally expect that having adversarial players participate in the game would have a deleterious impact on the consensus rate. This intuition is readily confirmed in Figure 2 (left), with all differences statistically significant ($p < 0.01$). However, this observation obscures a crucial distinction between two kinds of impact adversaries can have in our setting:

1. *Structural impact*: the adversarial nodes change network structure—in the extreme case, disconnecting the network among the *consensus* team members and
2. *Behavioral impact*: behavior of adversarial nodes limits the ability of the nodes on the consensus team to reach consensus.

There is a clear structural impact: 16% of games with 2 adversaries, and 34% of games with 5 adversaries become disconnected if we were to remove adversarial nodes. In the cases in which adversarial nodes disconnect the graph,³ consensus rate drops to 14%–15%, roughly what one would expect by random chance (if we only have two connected components, and use the consensus rate of 58% which obtains with no adversaries for each component, the expected consensus rate is 17%). Of course, it is worth remembering that the network is not, in fact, disconnected, and adversarial nodes need to deliberately prevent the information about network state from spreading through them. Indeed, not only do adversaries do so, the resulting consensus rates are slightly below expected, suggesting that adversarial behavior itself has an additional deleterious impact on the ability of nodes to coordinate.

To isolate the behavioral impact, in Figure 2 (right) we plot the proportion of times a pair which is k network hops apart agrees on a color at the end of the game, as a function of network distance k (we only include k with at least 100 instances), where network distance is defined as the number of nodes between a pair. Here, we can still see a systematic decrease in coordination success, as a function of the number of adversaries, no matter how far apart nodes are. For example, even network neighbors (i.e. $k = 1$) are finding it increasingly more difficult to agree on a color, on average, as we increase the number of adversaries.

Table 1. Average number of color changes per player in each game

Adversaries	Mean	Standard deviation
0	2.05	3.10
2	2.43	3.16
5	2.99	3.96

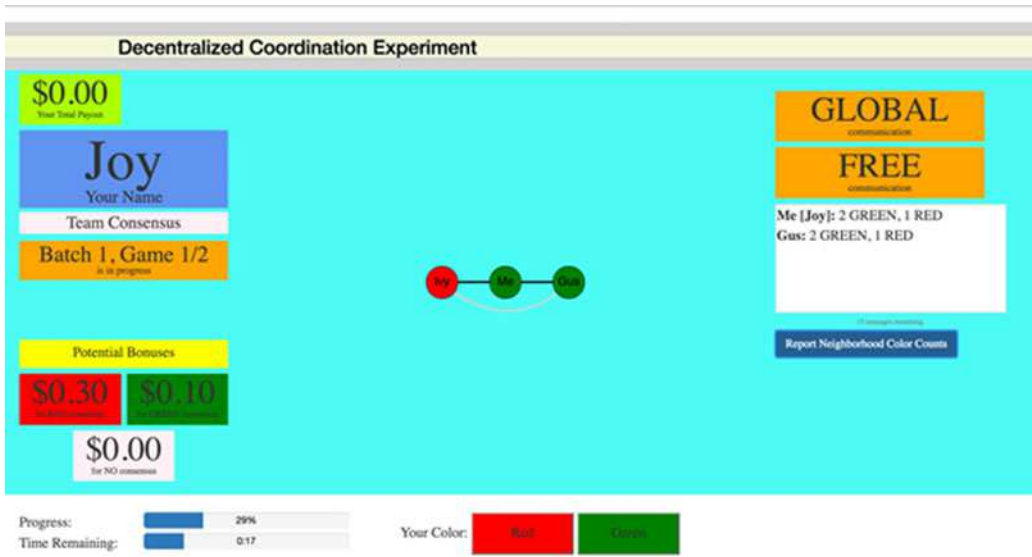


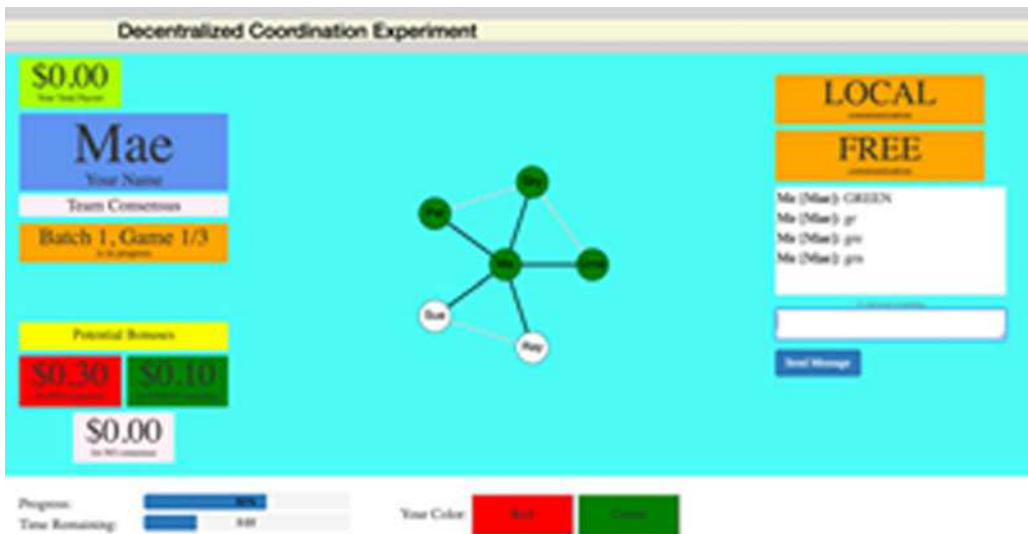
Figure 3. Impact of adversaries on the progress of consensus. Top: average portion of consensus team members in agreement. Bottom: histogram of time to consensus, as function of the number of adversaries.

Figure 3 captures the average progress of the consensus as a function of adversarial nodes in the game. Specifically, to imitate the process of a consensus, we extract the portion of consensus team members who chose the color green and the portion who chose the color red and report the maximal value for each game and on every time step. As one may expect, as the number of adversaries grows, the consensus team finds it harder to agree on the same action (color), starting from the first 10 s of the game. To make our analysis whole, we provide a histogram of the times different games reached consensus (or did not) on the right side of Figure 3.

Last, we present the average number of color changes per player in each game as a function of the number of adversaries in Table 1. Observe that as the number of adversaries increases, the number of color changes increases as well. To further understand this, we break results up by player type in Table 2. We observe that the presence of adversaries increase the number of color changes for regular nodes, from 2.06 with no adversaries to 2.9 when there are 5 adversaries in the network, an increase of 50%. This pattern repeats for the visible and even for adversarial nodes, showing that the presence of adversaries induces all nodes to be more active. It is also noteworthy that adversaries make significantly more color changes (3.39), than the regular and visible players, who make 2.45 and 2.23 changes, respectively. One may hypothesize that adversaries use the color changes to actively mislead or confuse the consensus team, reducing the likelihood of local convergence to consensus. In contrast, visible nodes make fewer color changes than others, presumably to create greater stability in their neighborhood.

Table 2. Average number of color changes per player in each game, by type

Type	Adversaries	Mean	Standard deviation
Regular	0	2.0	3.16
	2	2.38	3.18
	5	2.90	4.07
Visible	0	1.94	2.50
	2	2.16	2.18
	5	2.58	2.59
Adversary	2	3.14	3.64
	5	3.50	3.96

**Figure 4.** The impact of communication on consensus rate. (a) Disconnected networks. (b) Connected networks.

4.2 Communication improves resilience

Next, we consider the impact that allowing players to communicate with their network neighbors has on their ability to coordinate successfully. Figure 4 shows that communication makes a clear impact (pooling broken and unbroken networks, all results are significant with $p < 0.01$). In the aggregate, the value of communication increases with the number of adversaries: when no adversaries are present, communication increases consensus rate by 23.5%, with two adversaries improvement rises to 35.1%, and with five adversaries games that feature communication are 54.5% more likely to reach consensus than those that do not. Moreover, Figure 4 breaks these results into two plots: one when networks are disconnected if we were to remove adversarial nodes (a) and one for the remaining connected networks (b). One would have expected that with disconnected networks consensus occurs largely by chance, and consequently, communication should have no impact. We can observe that this is not so: even when networks are disconnected by adversaries, communication increases consensus rate, nearly doubling it when there are five adversaries. To understand this result, observe that with no communication, consensus rates in disconnected networks are well below what it should be by *random chance*, whereas communication raises them to approximate parity with random chance. In other words, in this setting communication successfully parries the *behavioral* impact of adversaries.

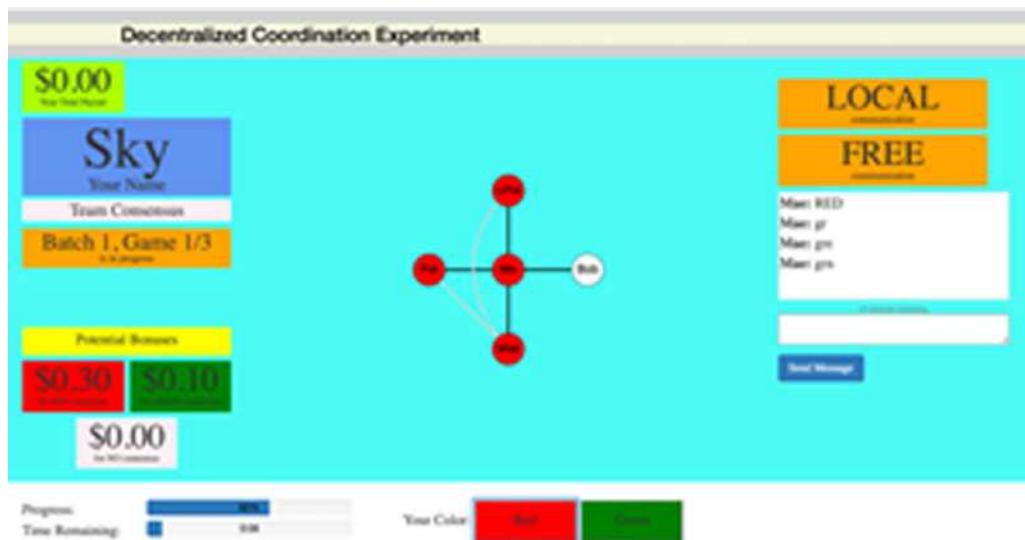


Figure 5. The impact of communication on pairs of nodes agreeing in color choice, by node distance. (a) 0 adversaries. (b) 2 adversaries. (c) 5 adversaries.

It is noteworthy that communication helps even when there are no adversaries, in contrast with prior results (Vorobeychik et al., 2017). The key distinction in our setting is the absence of the progress bar: now that this source of global information is missing, communication becomes considerably more informative.

Figure 5 unpacks the analysis of the impact of communication further by isolating, again, the behavioral impact of the adversaries, and the result is generally consistent, with communication increasing the likelihood of a given pair of nodes agrees on a color at the end of the game, particularly when they are relatively close to each other in the network.

We provide with Figure 6 that captures the progress of the consensus as a function of the ability to communicate. For each time step, we provide with the portion of the consensus team that agree on the color chose by most of its members. As depicted in the figure, toward most of the game, the ability to communicate help the consensus team to move toward unanimous choice of a single color. To make our analysis whole, we provide with an histogram of the times different games reached consensus (or did not), on the right side of Figure 6.

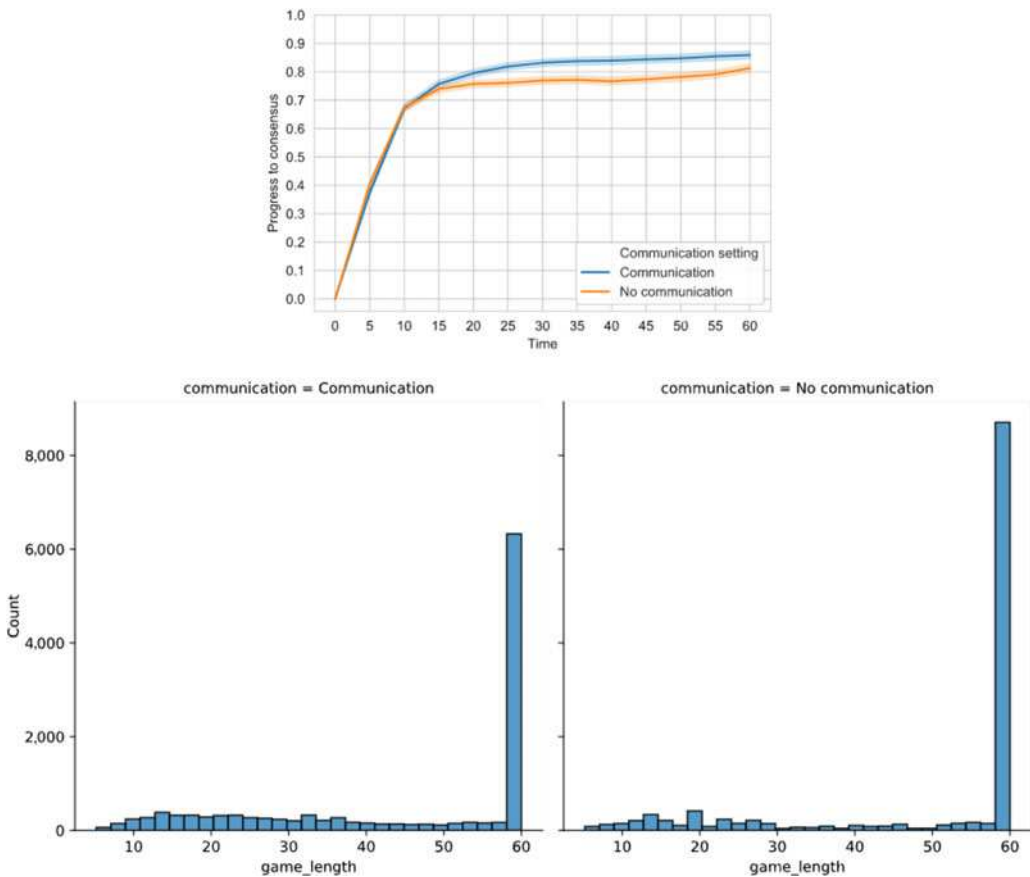
Finally, we present the average number of color changes per player in each game, broken up by player type, as shown in Table 3. As we can observe, when communication among players is not allowed, player of all types (regular, visible, and adversarial) make considerably more color changes across the game. In part this is due to communication serving as a coordination mechanism outside of the particular choices of color by the players. However, this could also be evidence that color changes themselves serve as a form of communication for players, as suggested in prior studies of networked consensus (Kearns et al., 2006; Judd et al., 2010).

4.3 The impact of network structure

Next, we consider what impact the network structure has on the ability of players to reach consensus with and without adversaries aiming to sabotage coordination. Figure 7 shows the results, broken up by network (BA, ER-dense, and ER-sparse), number of adversaries, and whether or not communication was allowed. Perhaps the most dramatic impact that communication has is on BA networks: when communication is enabled, two adversaries are unable to significantly impact

Table 3. Average number of color changes per player in each game, by type

Type	Communication	Mean	Standard deviation
Regular	Communication	1.93	1.88
	No communication	2.96	4.53
Visible	Communication	1.75	1.30
	No communication	2.71	3.13
Adversary	Communication	2.60	2.64
	No communication	4.18	4.66

**Figure 6.** Impact of communication on the progress of consensus rate. Top: average portion of consensus team members in agreement. Bottom: histogram of time to consensus, as function of the ability to communicate.

consensus rate, in contrast with games with no communication, where consensus rates of BA networks drop by over 30%. This suggests that with few adversarial nodes, the ability to communicate endows scale-free networks with resilience *even in the face of behavioral manipulation* by adversaries (which we observe to have a significant overall effect otherwise). This finding complements the already well-known resilience of BA networks to random node removal (Albert et al., 2000).

We turn to visualize how the agreement among members of the consensus team progresses over time as a function of the network topology. As depicted in Figure 8, denser topology (i.e.

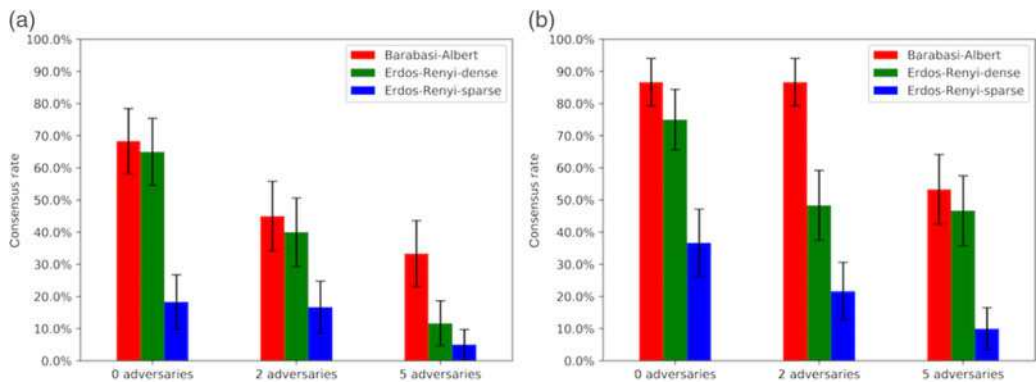


Figure 7. The effect of adversary players and network type on the consensus rate. (a) No communication. (b) Communication allowed.

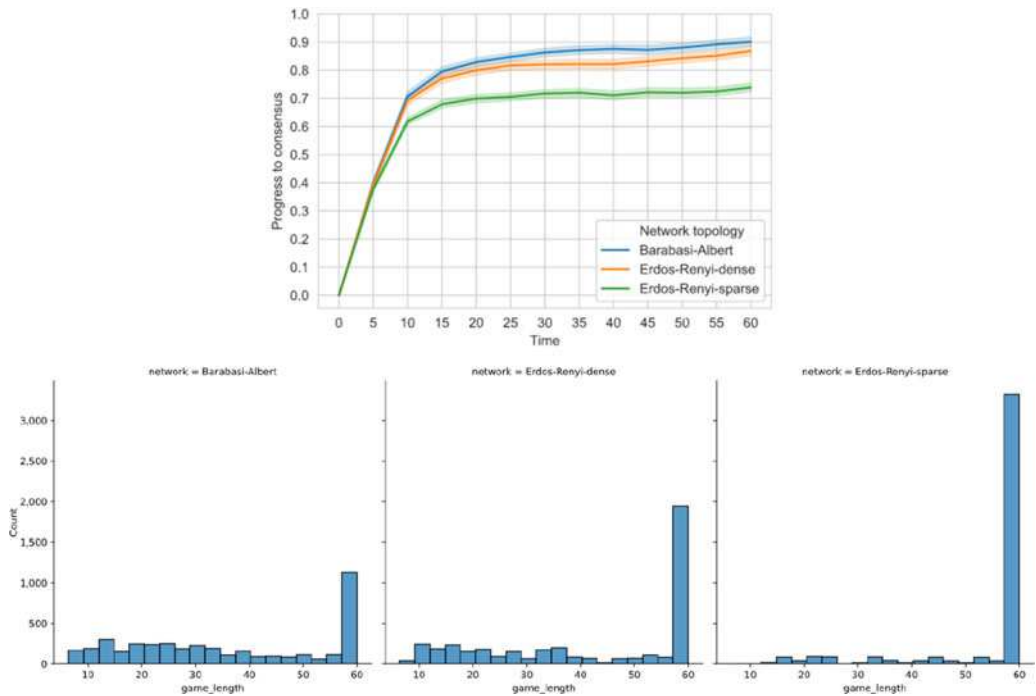


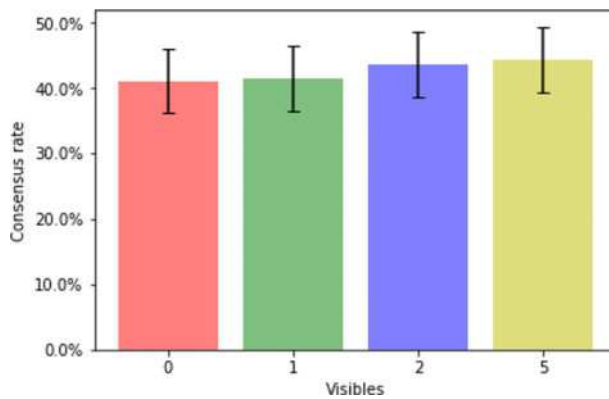
Figure 8. Impact of network type on the progress of consensus rate. Top: average portion of consensus team members in agreement. Bottom: histogram of time to consensus, as function of the network type.

Barabasi–Albert) makes it easier for the consensus team to progress toward a consensus, starting from the first 5 s of the game. Interestingly, while an average game on the Erdos–Renyi-sparse network got to a steady state of about 70% agreement starting from the 20th second of the game, an average game on the Erdos–Renyi-dense network, got closer to a consensus as time progresses. To make our analysis whole, we provide with an histogram of the times different games reached consensus (or did not), on the right side of Figure 8.

Finally, as depicted in Table 4, as the network becomes denser (i.e. moving from Erdos–Renyi-sparse to Erdos–Renyi-dense, and Barabasi–Albert) the average number of color changes increases.

Table 4. Average number of color changes per player in each game

Type	Topology	Mean	Standard deviation
Regular	Barabasi–Albert	2.12	2.89
	Erdos–Renyi-dense	2.25	2.48
	Erdos–Renyi-sparse	2.96	4.71
Visible	Barabasi–Albert	1.92	1.61
	Erdos–Renyi-dense	2.00	1.98
	Erdos–Renyi-sparse	2.76	3.33
Adversary	Barabasi–Albert	2.87	3.06
	Erdos–Renyi-dense	3.09	3.45
	Erdos–Renyi-sparse	4.23	4.76

**Figure 9.** The effect of visible players on the consensus rate.

4.4 The value of “Trusted” nodes

Lastly, we look at the value of “trusted” or visible nodes, that is, nodes whose intention of achieving coordination is visible. Prior research using stylized models of node behavior demonstrated that the presence of trusted nodes in a network can significantly improve resilience to attacks (Abbas et al., 2014, 2017; Usevitch & Panagou, 2018). It is thus natural to hypothesize that nodes that are visible on the *consensus* team (we can view these as trusted nodes, in the sense that they are known not to be adversarial) would significantly facilitate consensus. Remarkably, Figure 9 shows that this is not the case: as we increase the number of visible nodes, the impact on consensus rates is almost undetectable. The reason for the difference is that typical models assume that trusted nodes cannot be attacked. In our case, trusted nodes (as any other node) have no information about who the adversaries are, and, consequently, can also be influenced by the attackers, albeit indirectly.

To understand the impact of visible (trusted) nodes in greater depth, we unpack the results in Figure 10 by the number of visible nodes, the number of adversaries, and whether or not communication is allowed. With 0 or 2 adversaries, it is difficult to see any systematic improvement in performance as we increase the number of trusted nodes. However, with five adversaries and communication, having visible nodes constitutes a clear improvement over having none ($p < 0.05$). Thus, merely having trusted nodes is of dubious value, but allowing players (as well as the trusted node) to communicate can improve resilience when there are many adversarial nodes.

Next, we analyze how the agreement among the consensus team members progresses as a function of the number of visible nodes. As depicted in Figure 11, there is no significant difference in

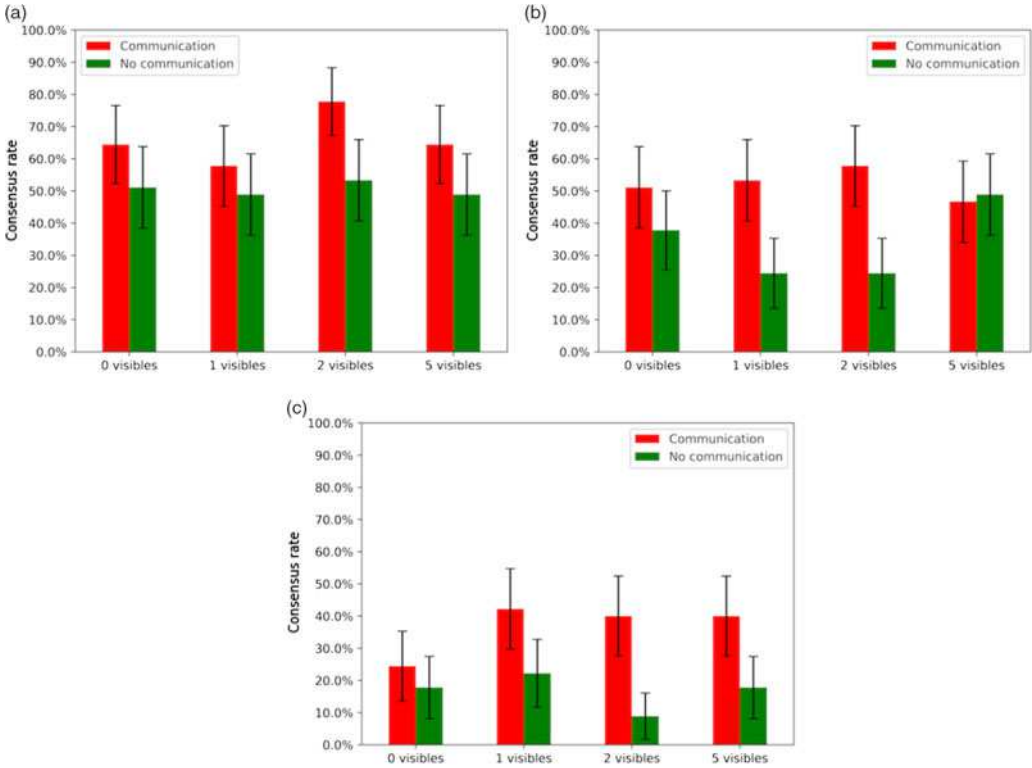


Figure 10. The effect of visible and adversarial players given the type of communication on the consensus rate. (a) 0 adversaries. (b) 2 adversaries. (c) 5 adversaries.

the way consensus was formed given a different number of visible nodes. To make our analysis whole, we provide a histogram of the times different games reached consensus (or did not), on the right side of Figure 11.

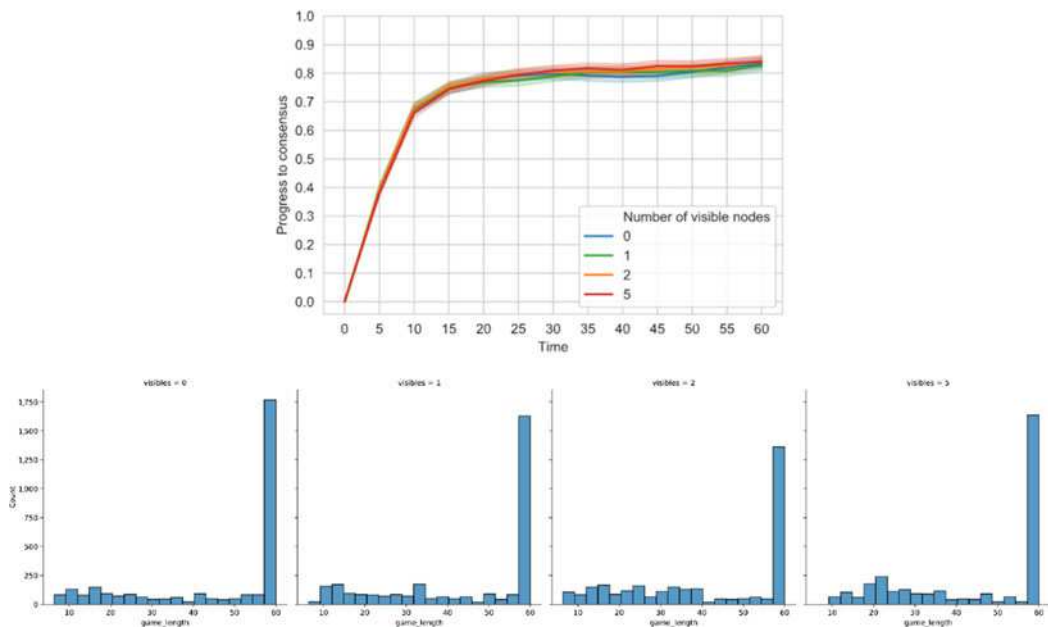
Last, we present the average number of color changes per player in each game as a function of the number of visible nodes in Table 5. It appears that the number of visible nodes does not have much impact on this.

5. Analysis of communication behavior

The previous section showed that allowing communication between the different nodes in a network improves the network's robustness. In this section, we focus on the way players communicate and the communication patterns of the regular, visible, and adversarial players. First, we observe that, no matter what the role of the player, the largest single class of messages attempt to stimulate coordination by naming a specific color (45% of all messages). Examples of this include messages that state a color (e.g. "GREEN"), or suggest that everyone use a particular color (e.g. "go for green", or "all green"). We term all messages of this kind *coordination* messages. Another common form of communication is what we call *information* messages (12% of all messages), whereby players attempt to inform their network neighbors of their local state; an example of such a message would be "5/5 red", suggesting that five out of five neighbors of the node are choosing *red*, or "3r2g", which communicates that three of the node's neighbors are choosing *red* and two are choosing *green*.

Table 5. Average number of color changes per player in each game

Visibles	Mean	Standard deviation
0	2.48	3.09
1	2.63	3.75
2	2.68	4.1
5	2.32	2.84

**Figure 11.** Impact of visible nodes on the progress of consensus rate. Top: average portion of consensus team members in agreement. Bottom: histogram of time to consensus, as function of the number of visible nodes.

On average, a typical player sends quite a few messages, although this number varies dramatically depending on the player's role. For example, a regular (non-visible) member of the *consensus* team sends on average 1.11 messages each game. On the other hand, a visible node sends 1.27 messages per game, and an adversarial node only 0.78 messages. The fact that adversarial nodes make such limited use of the communication interface to prevent consensus is especially interesting—clearly, players taking on the role of adversaries are relatively unaggressive in this role. In any case, when they do communicate, what do they write?

One thing we observe is that adversaries send considerably more coordination and information messages than *consensus* players: 53% and 15%, respectively. Thus, while adversarial nodes engage in considerably less communication, they appear to be more deliberate about it. Next, we explore precisely how adversarial nodes use each of these two categories of messages toward their ends.

Table 6 summarizes the number of messages sent by each type of player, standard deviation is given in brackets. The second column from the left depicts the average number of messages sent, while the third and fourth break this analysis to games that ended with consensus and those who do not, respectively. As depicted from the table, visible nodes deliver most of the messages. Interestingly, it seems that games that did not reach a consensus result in many more messages than those that ended with a consensus. To complete our analysis, we provide with Table 7 that summarize the average portion of player of each type that sent messages across all games. The

Table 6. Number of messages sent by different type of players

Type	msg/node	msg/node (consensus)	msg/node (no consensus)
Adversary	0.78 (1.38)	0.54 (0.91)	0.95 (1.61)
Regular	1.11 (1.38)	0.82 (1.06)	1.43 (1.59)
Visible	1.27 (1.47)	0.92 (1.07)	1.66 (1.73)

Table 7. Portion of players that communicated, by type

Type	All games	Consensus games	No consensus games
Adversary	0.42 (0.49)	0.37 (0.48)	0.46 (0.49)
Regular	0.60 (0.49)	0.53 (0.5)	0.67 (0.46)
Visible	0.66 (0.47)	0.60 (0.49)	0.72 (0.44)

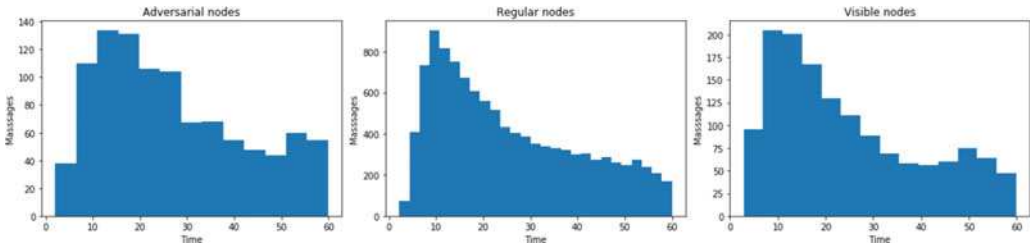


Figure 12. Number of messages sent by different types of nodes.

second column from the left depicts the average portions of players that communicated, while the third and fourth break this analysis to games that ended with consensus and those who do not, respectively. Note that the pattern is consistent with the number of messages sent. Most of the visible nodes chose to communicate (66%, on average) compared to the minority of adversaries (42%, on average). Another interesting observation, is that in game that ended with a consensus, a lower portion of the players chose to communicate. In games that did not end in a consensus, the portion of players that communicated increased by 25%.

Next, we extract the histogram of messages sent by each type of player as a function of time. As depicted by Figure 12, the behavior of all types is quite similar. Interestingly, adversarial and visible nodes start broadcasting messages from the beginning of the game while regular nodes seem to wait some time for the network to converge before messaging in high quantities. A closer observation of the adversarial nodes behavior, reveals that these nodes send 12.4% of their messages during the last 10 s of the games. One should remember that consensus games do not last the entire 60 s (the median time for a consensus game is 22 s). When looking at the portion of messages sent in non-consensus games, the portion of messages sent during the last 10 s is 16.3%.

While observing the behavioral pattern of visible nodes, we found that in games with one visible node, this node only communicates in 68.3% of the games, while in games with three and five visible nodes, these nodes will communicate in 83.9% and 98.6% of the games, respectively.

A natural strategy for an adversarial node in our setting is to send messages that are deliberately misleading. We now explore the extent to which they do so for the two types of messages we identified above: *coordination* and *information* messages.

First, consider the coordination messages. Presumably, a misleading coordination message attempts to coordinate neighbors on a color that differs from that chosen by most of their neighbors. However, such messages need not be *deliberately* misleading. With so many messages sent, there is bound to be a certain amount of noise in the nature of the messages. Moreover, players may have a perception of what the likely consensus outcome is regardless of the particular current

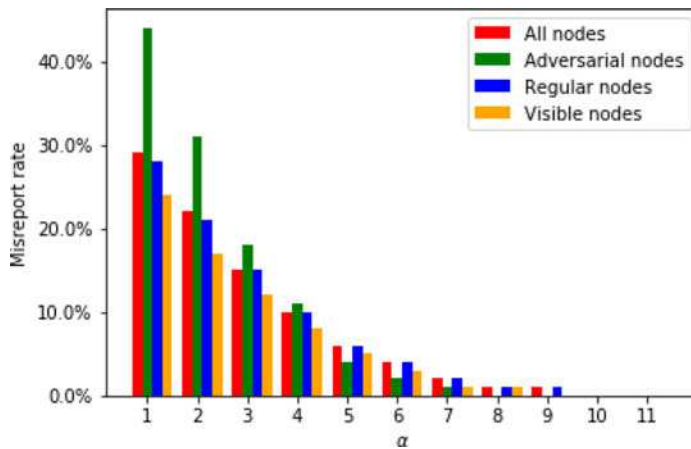


Figure 13. α -majority misreport rate.

state of their local network (for example, when it appears that most of their neighbors are green most of the time, even if the majority of them happen to be choosing red at a particular point in time). Consequently, what is most important is the *relative* rate with which such misleading messages are sent by adversaries, in comparison to other players. These results are shown in Figure 13, as a function of the relative size of the majority. Specifically, the α on the x-axis of the plot represents an α -majority when at least α more players are choosing one color, and the coordination message is sent attempting to coordinate on another. As we would expect, the fraction of such misreports (the misreport rate on the y-axis) drops quickly with increasing α . What is interesting is that, indeed, adversarial nodes are distinctly more misleading in this way than other nodes—a clear indication of such misleading messages being a part of a deliberate strategy. However, no less interesting is the fact that once $\alpha > 3$, there is no longer a meaningful difference between adversarial players and others. In other words, adversarial nodes attempt to be misleading, but only when it is not blatant.

Considering now information messages, we make a similar qualitative observation. For such messages, we can quantify “lying” as incorrectly reporting local state. Of course, we again must account for noise, in this case, erroneous reporting, which is not deliberately a lie; thus, the focus is on the relative difference between reported and true state, in comparison with non-adversarial players. We find that adversaries send information messages which are, indeed, more inaccurate on average than others. Specifically, when the difference between reported and true state is normalized by the number of neighbors, adversaries are off by 0.5, in comparison with non-visible nodes, which are off by 0.3, and visible nodes, which are off by 0.4. What we find, again, is that we see evidence that adversarial nodes deliberately lie about their state, but such lies are rarely egregious.

One may wonder if the lack of aggressiveness on the part of adversaries is a sign of human cognitive limitation, or perhaps social consciousness (unwillingness to act in a way that causes harm to many others). However, there is another natural explanation. Recall that the identity of nodes (adversarial or not) is largely invisible. An overly aggressive adversary may well reveal themselves as adversarial to all neighbors, who subsequently merely ignore them. Thus, pulling punches may be a way to remain undetected, and may thereby be a sound strategy.

Next, we analyzed how communication is affected as the number of adversaries, and visible nodes change. Figure 14 shows that as the number of adversaries increases, the number of average messages sent increase as well. We noticed a similar pattern when breaking up our results by the network topology. As the network become sparser, the number of messages increases. One

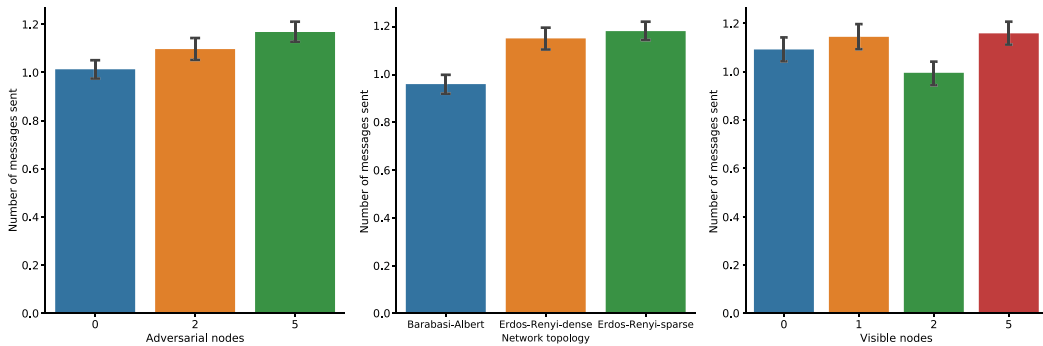


Figure 14. Average number of messages sent by each player.

may hypothesize that this increase in communication is aimed to overcome that lack of connectivity. We did not find any pattern for the number of messages as a function of the number of visible nodes. When there were none, the average number of messages was 1.09 per player, which increases to 1.14 when there was one visible node, but decreased to 0.99 when there were two, and increases again to 1.16 when there were five visible nodes in the consensus team.

6. Data-driven agent-based modeling

Our observations of collective behavior in adversarial consensus games provide a starting point for the next step: the development of a data-driven agent-based model (DDABM) of this scenario. The DDABM methodology builds agent-based models from data ground up: first, data of individual *human* behavior is used to learn computational models of this behavior, and second, such models are tied together in an agent-based simulation through variables that take as input observed behavior by other agents (in our case, network neighbors and visible nodes) (Zhang et al., 2016). Crucially, model validation must be performed at both the individual and aggregate levels.

6.1 Modeling and analysis of individual behavior

We start by using the data generated in our experiments to develop computational individual agent models that will give rise to a credible agent-based simulation model with more predictive power than the conventional stylized models. An additional benefit of these models is that they will provide qualitative insight into human behavior in adversarial networked consensus. While we found communication as an important factor in our analysis of the experiments, it is not clear how to model it in simulation. Therefore, we focus on the setting with no communication and defer the issue of modeling communication to future work.

Given that the players in our game only choose between two colors, the modeling task before us may seem simple at first glance. This simplicity, however, is quite misleading. In particular, there are several complications in modeling human behavior in our settings. The first is the fact that individuals may have three distinct roles:

- (1) Adversarial node: a member of the *no-consensus* team, whose goal is to prevent consensus among the “good” nodes (i.e. nodes on the *consensus* team),
- (2) Visible (“trusted”) node: a member of the *consensus* team who is visibly a member of this team (i.e. all neighbors can see that this node is on the *consensus* team), and
- (3) Regular node: all other members of the *consensus* team.

It is intuitive that adversarial nodes behave differently from others. For example, adversarial nodes change color more often than others: 2.9 times per game, in comparison with visible consensus

team players, who make only 2.1 changes in a game, and non-visible nodes, who change their color only twice a game, on average. Below, we observe that visible nodes also behave differently from regular nodes. The second challenge is that nodes in *any* of these roles may behave differently depending on whether they see visible nodes among their neighbors. The third is the fundamental challenge of how we should model real-time color choices by the players.

We address the third challenge by discretize time into 1-second intervals, so that there are (up to) 60 decision points in any game (as a game lasts 60 s).

To address the first two challenges, we created distinct behavioral models for the three roles, and distinct models for the situations when they have a visible node as a neighbor, and when they do not (thus, six individual agent models altogether).

Each of these cases raises an additional complication: agents make two kinds of decisions during the span of a game. First, as they start as “white” (non-committed), they must choose an initial color, and subsequently, choose whether to switch their color. Consequently, we split the decision model into two parts: (1) choosing the initial color and (2) switching their color. The rationale is that the initial decision is a deliberate choice of a particular color, and includes both the timing of changing from the initial default “white” color to either red or green, as well as the particular choice between these two. In contrast, once a color is chosen, players exhibit a considerable amount of inertia: they change color less than once every 20 s on average. Thus, modeling the decision to switch (or, effectively, the *timing* of a color switch) naturally captures such inertia, and also cleanly captures the inherent symmetry of their decision at this point, since players do not have a preference for one color over the other beyond reaching consensus.

Finally, the initial decision was itself split into two models: the first modeling the timing of the initial color choice, and the second modeling which color is actually chosen. Consequently, altogether we learned 18 different behavior models, or 3 models for each of the 6 roles and neighborhood assignments. Next, we describe these 3 models (which are qualitatively the same for each of the role $\times$ neighborhood predicaments): *timing of initial color choice*, *choosing the initial color*, and *timing of color change*. We briefly note that all models below are highly effective: either they exhibit high accuracy (90%–95%), or large likelihood improvement over a frequency-based baseline (50%–100% improvement).

Timing of Initial Color Choice. Our first set of models predicts the timing of the initial choice of color, or, more precisely, the probability that the initial color is chosen in a discrete-time unit. For these models, the features are: D_{inv} , the absolute difference between the fraction of a player’s non-visible neighbors that picked *red* and the fraction that picked *green*; D_{vis} , the absolute difference between the fraction of a player’s visible neighbors that picked *red* and the fraction of those who picked *green* (if the player has visible neighbors); N_{vis} , the number of a player’s neighbors that are visible, and N_{inv} , the number of a player’s neighbors whose are non-visible (note that $N_{vis} + N_{inv}$ is the total number of neighbors the player has). The decision model is represented by a logistic regression with these features, the parameters (coefficients) of which we learned from experimental data. We added l_1 (sparse) regularization to control for overfitting, with regularization parameter tuned using cross-validation. In all models, VN is a boolean feature indicating if a node has a visible neighbor. All feature were normalized.

The learned model coefficients for both the model with and without visible neighbors are given in Table 8. The results offer several interesting insights. First, we can see that disagreement among neighbors stimulates a player to make an initial color choice earlier. This is somewhat surprising, as we may expect players to wait until their neighbors had come to a near-consensus before making an initial move. Second, disagreement among visible nodes has a more significant, positive impact on the likelihood of choosing a color at a particular time point. Third, the behavior of adversarial nodes is broadly consistent with the first observation, but not with the second: such players appear to be more stimulated by disagreement among non-visible than among visible (trusted) neighbors.

Table 8. Color-picking model, P(pick a color)

Type	VN	Intercept	D_{inv}	D_{vis}	N_{inv}	N_{vis}
Reg	No	−1.952	1.29			
	Yes	−2.21	0.548	0.933	0.002	0.016
Vis	No	−2.045	1.742		0.04	
	Yes	−1.734	0.579	0.84	−0.061	0.048
Adv	No	−2.284	1.25		0.011	
	Yes	−2.744	0.802	0.662	0.025	0.155

Table 9. Red picking model, P(red| pick a color)

Type	VN	Intercept	G_{local}^{inv}	G_{local}^{vis}	R_{local}^{inv}	R_{local}^{vis}
Reg	No	0	−4.863		5.032	
	Yes	−0.066	−2.855	−2.022	3.453	1.733
Vis	No	0.109	−4.411		4.202	
	Yes	0.188	−3.215	−1.599	2.395	1.996
Adv	No	−0.023	0.817		−0.649	
	Yes	−0.286	0.172	0.732	−0.204	

Choosing the Initial Color. Conditional on deciding to choose the initial color in a particular discrete-time unit (per our previous models), the next decision we model is which of the two colors the player chooses. We again use l_1 -regularized logistic regression, where we predict the probability that a player chooses “red” as their initial color (conditional on choosing *some* initial color). As before, we use cross-validation to tune the regularization coefficient. For these models, the features are: G_{local}^{inv} , the fraction of a player’s non-visible neighbors choosing *green*; G_{local}^{vis} , the fraction of a player’s visible neighbors choosing *green*; R_{local}^{inv} , the fraction of a player’s non-visible neighbors choosing *red*; and R_{local}^{vis} , the fraction of a player’s visible neighbors choosing *red*. Note that $G_{local}^{inv} + R_{local}^{inv}$ and $G_{local}^{vis} + R_{local}^{vis}$ are not necessarily 1, since some of the neighbors may not have yet chosen a color. As before, all of the features were normalized.

The coefficients of the learned models are presented in Table 9. The results closely follow expectations here: the more neighbors (visible and not) are choosing *red* as opposed to *green*, the more likely the *consensus* team player to choose *red* as the initial color. On the other hand, adversarial players tend to act in opposition to their neighbors, with *red* prevalence in their local neighborhood generally leading them to choose *green*.

However, with regard to the adversarial players, we make a few noteworthy observations. First, note that adversaries are much more influenced by visible nodes than non-visible neighbors (acting more strongly in opposition to these), whereas regular players tend to be less swayed by the behavior of visible neighbors as compared to others in their neighborhood. Presumably, the adversaries are deliberately trying to counter the presumed influence of the visible nodes, which they appear to overestimate. Second, adversarial nodes act *relatively unaggressively*: the negative relationship between neighbor choices and their own initial color choice is relatively slight, in comparison with the magnitude of the positive relationships for the regular nodes (remember that features are normalized, so this comparison is meaningful). This observation that adversarial nodes are less aggressive in their activities aimed at thwarting consensus is surprising. We will return to it below, as we make a similar observation in the case of player decisions about when to change their previously chosen color.

Table 10. Color-changing model

Type	VN	Intercept	O_l^i	O_l^v	C_l^i	C_l^v	N_i	N_v
Reg	No	−3.98	2.65		−0.33		−0.01	
	Yes	−3.79	1.1	1.48	−0.87	0.09	0	−0.03
Vis	No	−4.11	2.7		−0.1		−0.01	
	Yes	−3.53	1.07	1.27	−0.33	−0.29	−0.06	0
Adv	No	−2.8	−1.13		1.19		0	
	Yes	−2.72	−0.6	−0.37	0.95	0.30	0	−0.2

Timing of Color Change. Our last set of models determine the timing of a color change by a player. More precisely, we again learn l_1 -regularized logistic regression models which represent the probability that a player switches to the other color (either from *red* to *green*, or vice versa) at a given discrete-time unit. For these models, the features are: O_l^i , the fraction of a player's non-visible neighbors choosing the opposite color from the one chosen by the player; O_l^v , the fraction of a player's visible neighbors choosing the opposite color from the one chosen by the player; C_l^i , the fraction of a player's non-visible neighbors choosing the same color as the player; C_l^v , the fraction of a player's visible neighbors choosing the same color as the player; N_v , the number of a player's neighbors who are visible; and N_i , the number of a player's neighbors that are non-visible players.

The model coefficients are presented in Table 10. The broad results are again intuitive: as we would expect, when the local color choices oppose that of a player, a regular player tends to switch, whereas the adversary tends to stay with their current color choice. However, unlike their choice of the first color, here the adversaries respond less aggressively to visible node decisions as compared to those for their remaining neighbors.

Interestingly, as we had observed above, adversarial nodes appear to be somewhat less aggressive in acting *against* the neighborhood trends, as compared to *consensus* players in their decisions to switch to be better aligned with these. This is at first glance unexpected: why would adversaries hold back, rather than aggressively opposing an emerging consensus in their neighborhood? We conjecture that the explanation is that they are concerned about being covert. If adversarial nodes act in a way that opposes neighborhood choices too aggressively, they run the risk of being discovered by their neighbors as such, at which point their behavioral influence would, presumably, be minimized. Consequently, adversarial nodes likely attempt to achieve their disruptive goals without being overly obvious to their non-adversarial neighbors.

6.2 Agent-based modeling

Given the computational models of human behavior described above, it is direct to construct an agent-based model (ABM): one simply instantiates each agent as a node on an exogenously specified network, with roles assigned randomly according to an exogenously specified model. In our case, we use the same random assignment model as in the human subjects experiments.

6.2.1 Model validation

While statistical and face validity are essential steps in confirming that our individual behavior models are reasonable, we now add another dimension: validation in terms of *aggregate outcomes* of agent-based simulations. Specifically, we simulate identical environments as in our experiments using our constructed ABM, but now using artificial agents and in discrete time, for 60 iterations (since each time step in our models is equivalent to 1 s in the experiments). Finally, we compare both qualitative trends, and quantitative outcomes, to those reported in the experimental results section above. Quantitatively, the agreement is reasonable, with the largest deviation between

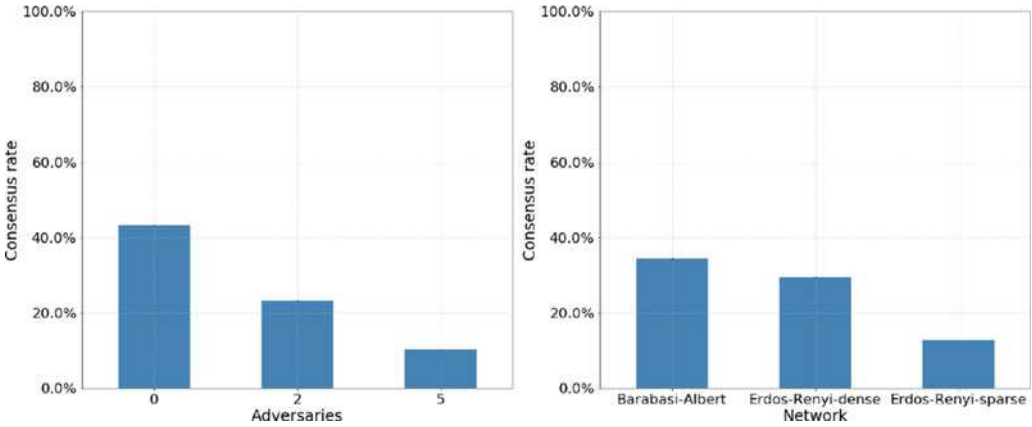


Figure 15. Coordination ratios as a function of single variable.

simulation outcomes, and the experimental consensus rates are within 0.14. The qualitative agreement is even stronger, as we illustrate in Figure 15, which shows predicted consensus rates (using simulations) as a function of the number of adversaries (left plot) and network topology (right plot). Comparing to corresponding results from the human subject experiments, we can observe broad qualitative agreement. Note that the agreement between simulated and experimental results we achieve for games at this scale (at least 20 players, with considerable interdependencies in behavior) compares quite favorably with similar efforts for devising artificial agents to model coordination in prior literature (Judd et al., 2010).⁴ The degree of consistency between simulations and experiments is particularly noteworthy in our case, if one considers that we had to construct 18 distinct behavior models to capture human behavior.

Despite strong agreement with experimental findings, it is still natural to wonder whether our models are robust to small changes in parameters. Such robustness is crucial if we are to trust the models to remain predictive as we significantly change the setup of the experiment, as we do below. We now show that our model is, indeed, robust to *worst-case* perturbations in the parameters of regular players (as these dominate the simulations).

Recall that for each non-adversarial player, we have two models: the first when a player has at least one visible neighbor and the second when they do not. Since we have two types of non-adversarial actors (visible and non-visible nodes), we optimize coefficients of the four associated models with the objective of maximizing consensus rate, with the constraint that the l_1 norm of the modification does not exceed an exogenously specified ϵ . We approximately solve this problem using *Coordinate Greedy (CG)* local search, which iteratively chooses a parameter to optimize, and attempts to find the best improvement of this parameter. To abide by the l_1 norm constraint, we subsequently project the result into the feasible space.

The expected consensus rates as a function of ϵ are provided in Figure 16, where the red dashed lines represent the consensus rate when simulating with the original $\mathbf{w}$.

Based on this analysis, we conclude that even for relatively large ϵ , the impact is surprisingly small: it appears that incremental changes in behavior of individuals has little impact on ability to successfully coordinate (the impact is generally $< 5\%$ even for ϵ as large as 0.2). This observation is especially clear in the adversarial setting. We find that the impact of small changes in parameters is surprisingly small, increasing consensus rate by only a few percentage points even for relatively substantial values of ϵ .

Our last aspect of model validation compares the mean time to consensus between actual experiments and their simulated counterparts. The results, broken up by the number of adversaries,

Table 11. Mean time to consensus (in seconds). Parentheses include standard deviation

		Experiments	Model
Adversaries	0	42.2 (19.26)	49.24 (3.91)
	2	45.98 (18.77)	54.35 (2.28)
	5	53.03 (13.93)	57.48 (1.19)
Visibles	0	47.79 (18.16)	54.44 (3.8)
	1	47.6 (18.04)	53.72 (4.29)
	2	46.51 (18.03)	53.41 (4.55)
	5	46.37 (17.96)	53.19 (4.54)
Topology	Barabasi–Albert	40.36 (19.14)	51.61 (4.33)
	Erdos–Renyi-dense	44.97 (19.07)	53.47 (4.22)
	Erdos–Renyi-sparse	55.88 (10.97)	56.9 (1.8)

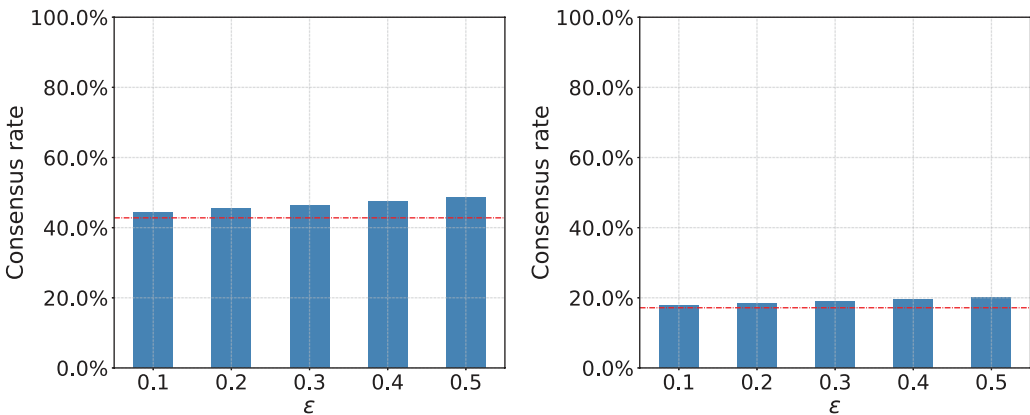


Figure 16. L_1 norm constraint. Left: No adversaries setting. Right: With adversaries setting.

visible nodes, and network topology, are given in Table 11. Note that our comparison in seconds is meaningful, since our agent-based model was learned from discrete-time data in which each iteration corresponded to 1 s. Consequently, the number of simulated iterations corresponds to the number of seconds. Throughout the table, we can observe a reasonable correspondence in average times—within a standard deviation in each case. Much more important, however, is that we observe a close *relative* correspondence: the relative ranking of all the times is consistent between simulations and experiments. This is the crucial feature we must preserve if we are to draw generalizable conclusions from simulation-based studies.

7. Model-based analysis

The human subjects methodology is inherently limited in the number of experiments one can run and, consequently, the space of alternative configurations we can consider. One of the many benefits of our developed agent-based models is the ability to simulate the expected consensus rates of different networks, differing in one (or more) of their properties. In this section, we engage in a further investigation of the problem of adversarial coordination using simulation experiments within an agent-based modeling framework. For this purpose, we make use of individual

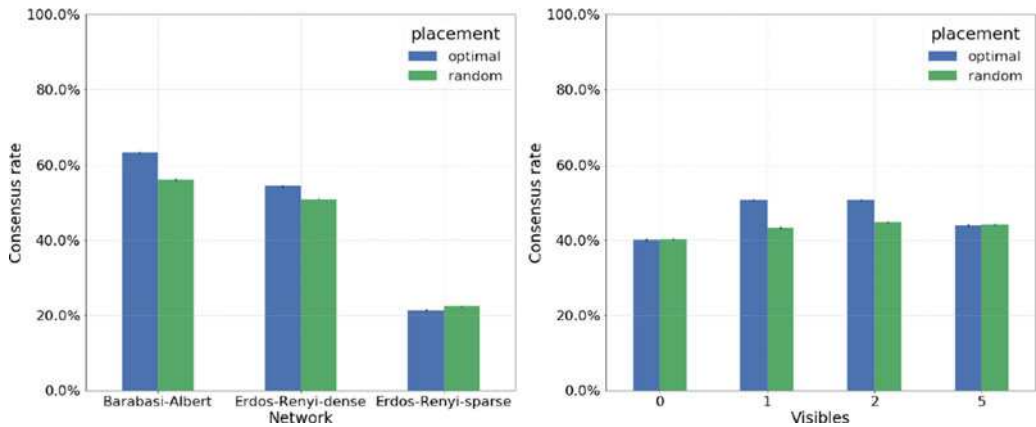


Figure 17. Consensus rate as a function of placement of visible nodes when no adversaries are present. Left: for different network topologies. Right: different number of visible nodes.

agent models developed above, and combine them into an agent-based model in which such artificial models are interacting on the exogenously specified networks. Specifically, we use the set of trained models to simulate the resulted consensus rate given: intelligent placement of visible (and adversarial) nodes, networks of different sizes, typologies, and visible nodes' strategies. Note that in order to provide with these insights, if these models were not available, thousands of participants would have had to be recruited (and being paid). Still, based on the model validation in the previous section, we can provide reliable results which mimic human behavior without these logistic and economic overheads.

7.1 Optimizing placement of trusted and adversarial players

In our experiments, we randomly assigned trusted and adversarial players to nodes within the network. We now explore the alternative possibility where the assignment of these is more deliberate. To study the problem systematically, we consider the decision of where to place trusted (visible) and adversarial nodes as a Stackelberg game with two players, the coordinator (the Stackelberg leader) and the adversary (the follower). The coordinator first places the trusted nodes on the network, and, fixing this placement, the adversary places adversarial nodes. The goal of the coordinator is to maximize consensus rate, which the adversary aims to minimize. In order to avoid time-consuming simulations in the optimization phase for both the coordinator and the adversary, we use a proxy objective of choosing a set of nodes maximizing the number of *unique* neighbors; we call this *optimal* for either player. Since the game in our case is relatively small, we solve for optimality by exhaustive search. In addition, we create three baselines for comparison: first, when both players choose nodes randomly (as in our experiments), whereas in the second and third baseline, one player chooses nodes randomly, whereas the other optimizes.

We first consider settings with no adversaries, and explore the impact of having an optimal placement of visible nodes, as compared with random placement. The results are presented in Figure 17, for different network topologies (left), and different numbers of visible nodes (right). The broad trend is that while optimal placement of visible nodes is typically helpful, the impact it has on consensus rate is quite muted, further bolstering our experimental observation that the value of having trusted nodes in this setting can be limited.

Figure 18 presents the results of considering the two placement strategies (random and optimal) for visible and adversarial nodes. From this figure, we can make several noteworthy observations. First, *adversarial players are highly effective with optimal placement*: consider blue

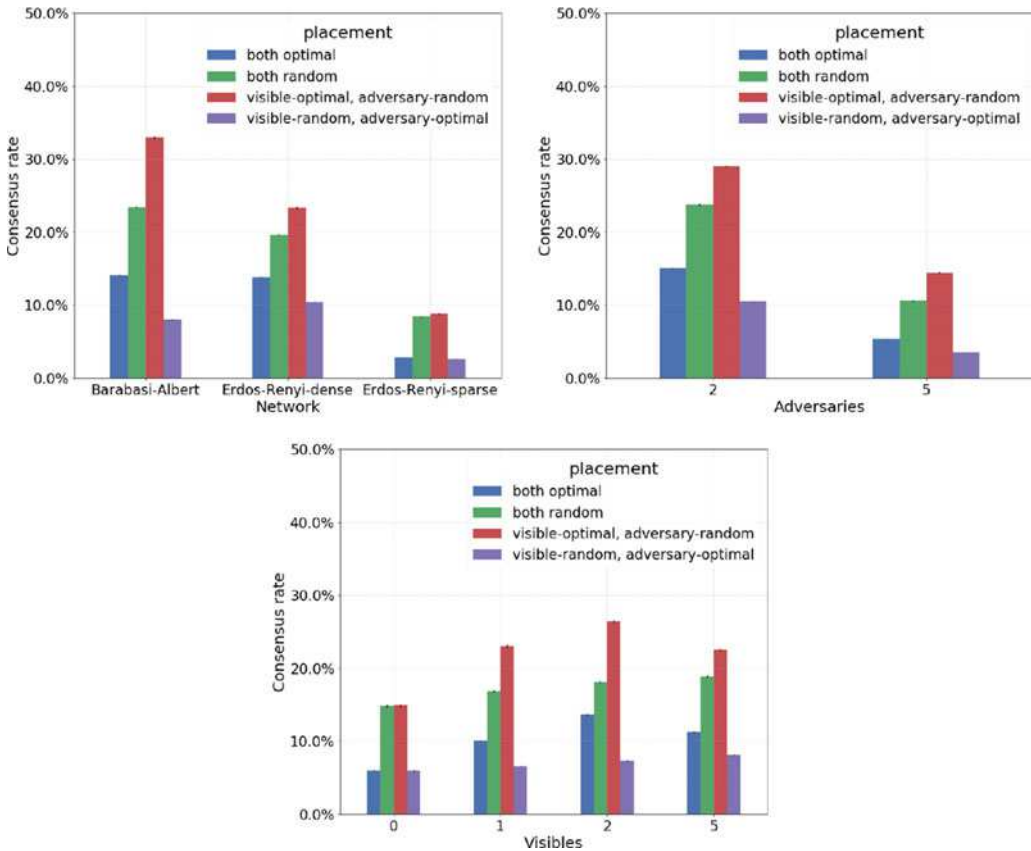


Figure 18. Consensus rate for different strategies of placing visible and adversarial nodes, as a function of: (Left) network topologies; (Right) the number of adversaries; and (Bottom) the number of visible nodes.

and purple (first and last) bars in the plots, which correspond to adversaries placed optimally. In both cases, consensus rates are quite low, for all network topologies, even with only two adversaries. This is especially surprising when we also consider the optimal placement of visible nodes, *which are placed before adversaries*, and can thereby ensure that networks remain connected even after adversarial nodes are added. While optimal placement of visible nodes clearly helps, the impact is smaller than we would have expected. Second, *optimally placing visible nodes helps*: consider the red bars (tallest in all plots), which correspond to the optimal placement of visible nodes, followed by random placement of adversaries. In this situation, we can observe a clear value of visible nodes, particularly for the scale-free (BA) topology. On the other hand, we can see that having 2 visible nodes is actually better than 5, which we conjecture is due to the increased potential for miscoordination among visible nodes themselves in the latter case.

7.2 Impact of network topology

In this section, we systematically explore the impact of network topological characteristics on the consensus rate. For BA networks, we consider two parameters: m , the number of connections we add to each node entering the network, which controls density, and γ , where the probability of connecting to a node with degree d is proportional to d^γ , which determines how heavy the tail of the distribution is. For ER networks, we vary the probability p that a pair of nodes are connected,

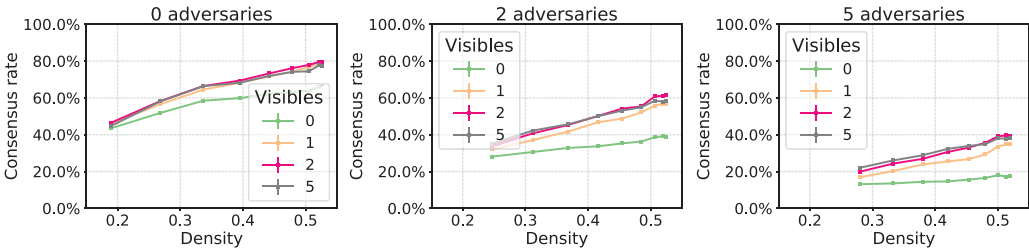


Figure 19. Consensus rate in BA networks as a function of network density, broken down by the number of adversaries and the number of visible nodes. Left: 0 adversaries. Middle: 2 adversaries. Right: 5 adversaries.

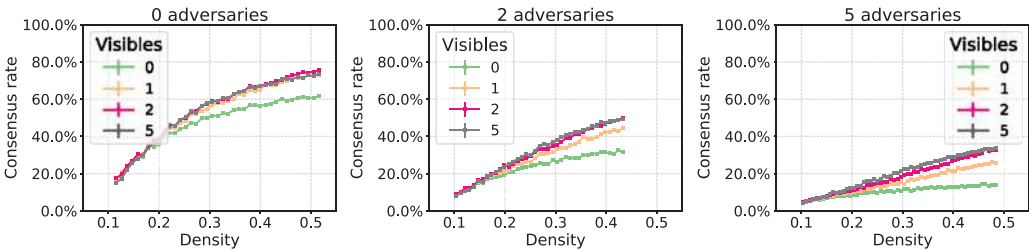


Figure 20. Consensus rate in ER networks as a function of network density, broken down by the number of adversaries and the number of visible nodes. Left: 0 adversaries; Middle: 2 adversaries; Right: 5 adversaries.

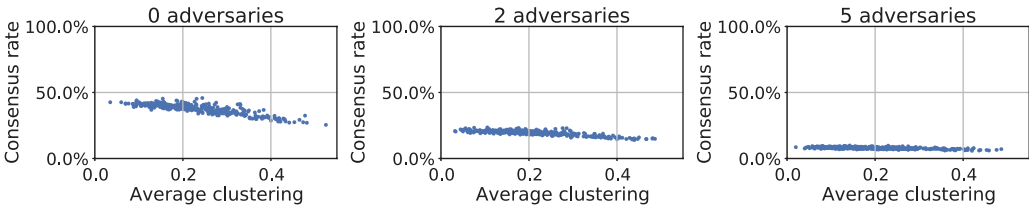


Figure 21. Consensus rate as a function of average clustering coefficient, broken down by the number of adversaries. Left: 0 adversaries. Middle: 2 adversaries. Right: 5 adversaries.

which is also directly related to density. Finally, we consider small-world networks (Watts & Strogatz, 1998), and vary the clustering coefficient.

Figure 19 shows the trends in consensus rates for different numbers of adversaries and visible nodes, as a function of network density, for the BA topology. As for the ER topologies, we provide with Figure 20 where there is little qualitative difference. Overall, increased density tends to improve consensus rates, with and without adversaries. More interestingly, the presence of visible nodes becomes more valuable with increased density as well, albeit 1 such node generally seems to suffice.

Figure 21 shows the impact of increasing the clustering coefficient (keeping density fixed). Here, we see that the trend is that higher clustering tends to hurt coordination, a finding that echoes previously reported results (Kearns et al., 2009). However, the trend becomes flatter as we add adversaries. We found that adding visible nodes, in this case, has no tangible impact on consensus rates.

Figure 22 shows consensus rate as a function of γ (higher implies greater disparity in degrees). In general, degree distributions with a heavier tail yield higher consensus rates, as long as there are only a few adversaries; the relationship becomes essentially flat with 5 adversaries. The reason is that heavy-tail distributions have fewer central actors with more neighbors, and as long as these

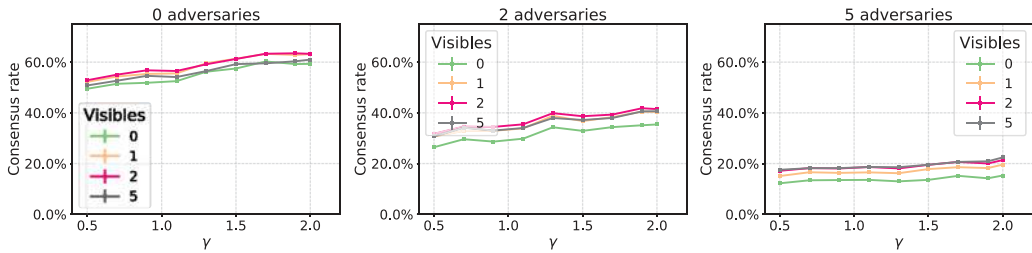


Figure 22. Consensus rate as a function of γ , broken down by the number of adversaries. Left: 0 adversaries. Middle: 2 adversaries. Right: 5 adversaries.

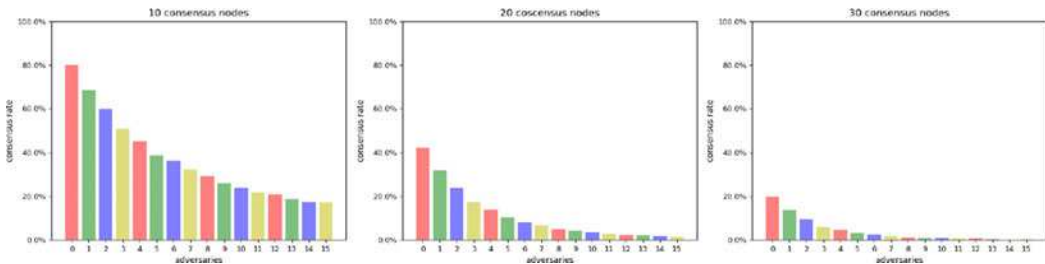


Figure 23. Consensus rate as a function of number of adversaries. Left: 10 consensus nodes. Middle: 20 consensus nodes. Right: 30 consensus nodes.

are not adversarial, they can considerably facilitate consensus. Since we assign adversarial nodes randomly in these experiments, it is unlikely that any such high-degree nodes are adversarial if there are only two adversaries, but it becomes far more likely with five adversaries.

7.3 Impact of network size

In this section, we turn our focus to the networks' size. One can assume that smaller networks will converge to a consensus with a higher probability, as fewer nodes have to agree on one state, and large networks may have a harder time coordinating. Our analysis provides concrete evidence to these assumptions, with an extensive analysis of vast of configurations varying the network size, and additional networks' parameters (i.e. adversaries, visibles, and network type). Each configuration was simulated 10,000 times, which totals in 81,000,000 simulations (15 different number of adversaries, various number of visible nodes, 3 network topologies, and 3 different sizes for the consensus group).

The first part of our analysis looks at the consensus rate as a function of the number of adversaries, for networks with 10, 20, and 30 consensus nodes. Figure 23 shows that as one may hypothesize, as the number of consensus nodes increases, the consensus ratio decreases, given the same number of adversaries. An interesting insight resulted from this analysis appears when we change the fixed parameter to be the number of visible nodes in the network. Recall that for networks with 20 consensus nodes, there was no significant change in the consensus rate as we varied the number of visible players (Figure 9). Still, as we show in Figure 24, as we increase the size of the consensus team to 30 nodes, a higher number of visible nodes resulted in a much higher consensus rate. On the other hand, when the consensus team is relatively small (i.e. 10 players), there is a threshold above which increases the number of visible nodes reduce the consensus rate rather than increasing it (similar to our results on node placement Figure 18).

We find this result to be valuable and decided to dig deeper. Since our evaluation is based on the DDABM, we can simulate any number of visible nodes in the network (up to the size of the

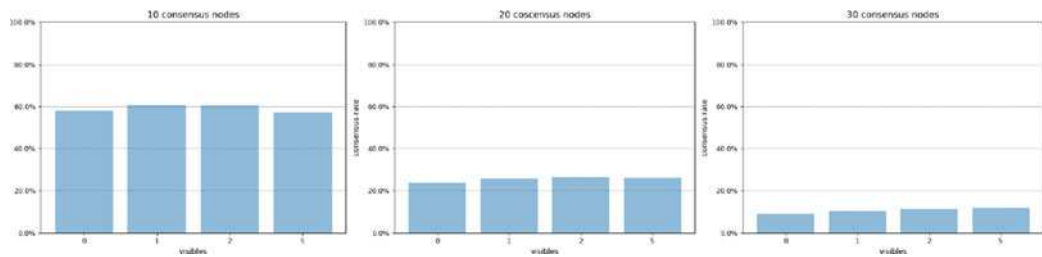


Figure 24. Consensus rate as a function of number of visible nodes. Left: 10 consensus nodes. Middle: 20 consensus nodes. Right: 30 consensus nodes.

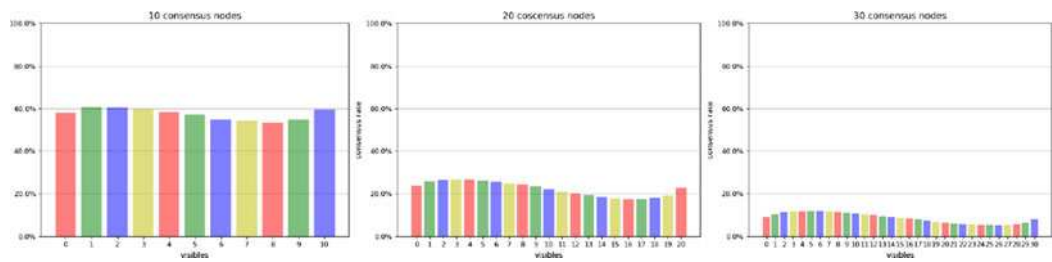


Figure 25. Consensus rate as a function of number of visible nodes. Left: 10 consensus nodes. Middle: 20 consensus nodes. Right: 30 consensus nodes.

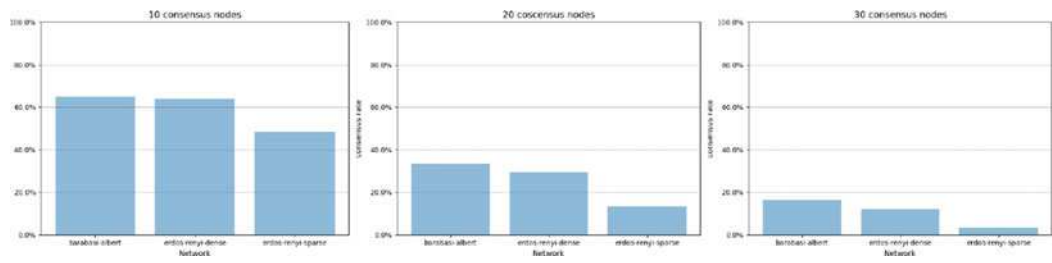


Figure 26. Consensus rate as a function of number of network's topology. Left: 10 consensus nodes. Middle: 20 consensus nodes. Right: 30 consensus nodes.

consensus team). Figure 25 shows the results of this extensive evaluation. As depicted in the figure, for every size of the consensus team, there is a threshold after which adding more visible nodes decreases the consensus rate. This result is very important as it shed new light on our evaluation regarding the effectiveness of visible nodes. This is one of the scenarios which exemplifies the importance and value of agent-based modeling. We note that we also find that populating the network solely by visible nodes is somewhat better. Still, this scenario does not seem very realistic, as if this is the case, the identity of the adversaries will be common knowledge as well.

Next, we analyze the effect of the network's size by breaking down our results according to the network's topology. As depicted in Figure 26, as the network grows (i.e. populated with more consensus nodes), the difference in the consensus rate between the BA networks to the ER ones, increases.

Finally, we analyze the time it takes the network to coordinate (i.e. when all the consensus nodes agree on the same color). As discussed before, an increase in the number of adversaries results in a decrease in the consensus rate. Our analysis revels another interesting fact on the coordination

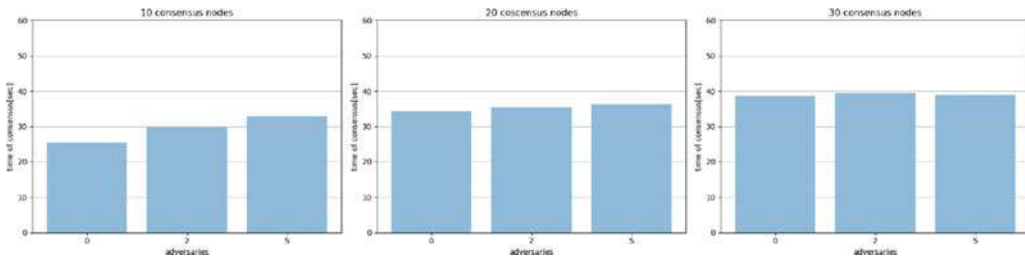


Figure 27. Consensus rate as a function of number of network's topology. Left: 10 consensus nodes. Middle: 20 consensus nodes. Right: 30 consensus nodes.

process. As depicted in Figure 27, as the number of adversaries increases, the time it takes the network to coordinate (if succeeded) increases. This new insight teaches us that adversaries are not just efficient in avoiding coordination, they also have the power to delay it. Lastly, as the number of consensus nodes increases, there are more nodes that need to coordinate and hence, as one may expect, the coordination time increases as well.

8. Conclusion

We consider the problem of adversarial consensus on social networks, both using human subjects and agent-based modeling methodologies. The overall goal of the subjects is to reach a global consensus on a particular color, despite adversarial nodes who attempt to prevent consensus. We find that while the ability to communicate can significantly improve coordination success despite adversarial presence, embedding trusted nodes within the network is of limited value. We observe several strategies used by adversarial players to subvert coordination, such as choosing a color that opposes local majority. However, we also note that these malicious activities are used in a somewhat subdued manner, suggesting perhaps an attempt of adversarial players to remain covert. We use experimental data to construct and validate an agent-based model of adversarial consensus. Extensive simulations using an agent-based model created based on experimental data additionally show that the importance does increase when their network location is optimized, but this improvement is often small, particularly when adversarial nodes are also optimizing location, and even though adversaries do so *after* we choose where to place trusted nodes. Furthermore, we explore the impact of network topological characteristics on the consensus rate. We show that for both BA and ER topologies, density has a first-order effect (i.e. increased density tends to improve consensus rate regardless of the existence of adversaries). We also show that when density is fixed, higher clustering tends to hurt coordination. As for the impact of network size on the consensus rate, We validate that larger networks are less likely to coordinate on a single decision. More importantly, we show that the effect of visible nodes depends on the size of the network. While 5 visible nodes may decrease the consensus rate with 10 consensus nodes, it is shown to be beneficial for a network with 30 consensus nodes.

Acknowledgments. A preliminary version of this paper appeared in the Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS-2019) Hajaj et al., (2019). We would like to thank the reviewers of AAMAS-2019 for the helpful comments on the earlier version of this paper. This research was partially supported by the National Science Foundation (IIS-1905558, IIS-1903207, and IIS-1939677), Office of Naval Research (N00014-15-1-2621), Army Research Office (W911NF1810208, W911NF1910241), Ariel University (grant number RA190000060), and the Ariel Cyber Innovation Center in conjunction with the Israel National Cyber directorate in the Prime Minister's Office.

Conflict of interest. None.

Notes

- 1 The full instructions given to the participants are available as an appendix for this paper.
- 2 Recall that there are always 20 nodes in the consensus team. Thus, when 2 members are visible, there are 18 regular nodes in this team.
- 3 A graph is disconnected if it is composed of more than a single connected component after removing the adversarial nodes.
- 4 Wunder et al. (2013) is noteworthy as well. However, they consider a public goods game, and aim to predict average contribution. Predicting the probability of consensus using such data-driven agent-based simulations appears to be a more challenging problem.

References

- Abbas, W., Laszka, A., & Koutsoukos, X. (2017). Improving network connectivity and robustness using trusted nodes with application to resilient consensus. *IEEE Transactions on Control of Network Systems*, 5(4), 2036–2048.
- Abbas, W., Vorobeychik, Y., & Koutsoukos, X. (2014). Resilient consensus protocol in the presence of trusted nodes. In *International symposium on resilient control systems* (pp. 1–7).
- Albert, R., Jeong, H., & Barabasi, A.-L. (2000). Error and attack tolerance of complex networks. *Nature*, 406, 378–482.
- Alon, N., Feldman, M., Lev, O., & Tennenholtz, M. (2015). How robust is the wisdom of the crowds? In *IJCAI* (pp. 2055–2061).
- Arenas, A., Camacho, J., Cuesta, J. A., & Requejo, R. J. (2011). The joker effect: Cooperation driven by destructive agents. *Journal of Theoretical Biology*, 279(1), 113–119.
- Bannikova, M., Dery, L., Obratsova, S., Rabinovich, Z., & Rosenschein, J. S. (2021). Reaching consensus under a deadline. *Autonomous Agents and Multi-Agent Systems*, 35(1), 1–42.
- Barabasi, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509–512.
- Bracha, G., & Toueg, S. (1983). Resilient consensus protocols. In *ACM symposium on principles of distributed computing* (pp. 12–26).
- Chakraborty, T., Judd, S., Kearns, M., & Tan, J. (2010). A behavioral study of bargaining in social networks. In *Proceedings of the 11th ACM conference on electronic commerce* (pp. 243–252).
- Cooper, R., DeJong, D. V., Forsythe, R., & Ross, T. W. (1992). Communication in coordination games. *Quarterly Journal of Economics*, 107(2), 739–771.
- Coviello, L., Franceschetti, M., McCubbins, M., Paturi, R., & Vattani, A. (2012). Human matching behavior in social networks: An algorithmic perspective. *Plos One*, 7(8), e41900.
- Demichelis, S., & Weibull, J. W. (2008). Language, meaning, and games: A model of communication, coordination, and evolution. *American Economic Review*, 98(4), 1292–1311.
- Ellingsen, T., & Ostling, R. (2010). When does communication improve coordination? *American Economic Review*, 100, 1695–1724.
- Elmalech, A., Sarne, D., David, E., & Hajaj, C. (2016). Extending workers' attention span through dummy events. In *Fourth AAAI conference on human computation and crowdsourcing*.
- Erdos, P., & Rényi, A. (1960). On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5(1), 17–60.
- Farrell, J. (1987). Cheap talk, coordination, and entry. *Rand Journal of Economics*, 18(1), 34–39.
- Farrell, J. (1988). Communication, coordination and Nash equilibrium. *Economic Letters*, 27, 209–214.
- Gracia-Lázaro, C., Ferrer, A., Ruiz, G., Tarancón, A., Cuesta, J. A., Sánchez, A., & Moreno, Y. (2012). Heterogeneous networks do not promote cooperation when humans play a prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(32), 12922–12926.
- Gvirts, H. Z., & Dery, L. (2021). Alexithymia and reaching group consensus. *Cognition and Emotion*, 35(3), 510–523. doi: 10.1080/02699931.2019.1675600.
- Hajaj, C., Hazon, N., & Sarne, D. (2015). Improving comparison shopping agents' competence through selective price disclosure. *Electronic Commerce Research and Applications*, 14(6), 563–581.
- Hajaj, C., Hazon, N., & Sarne, D. (2017). Enhancing comparison shopping agents through ordering and gradual information disclosure. *Autonomous Agents and Multi-Agent Systems*, 31(3), 696–714.
- Hajaj, C., Yu, S., Joveski, Z., Guo, Y., & Vorobeychik, Y. (2019). Adversarial coordination on social networks. In *Proceedings of the 18th international conference on autonomous agents and multiagent systems* (pp. 1515–1523).
- Hardin, G. (1968). The tragedy of the commons. *Science*, 162(3859), 1243–1248.
- Judd, S., Kearns, M., & Vorobeychik, Y. (2010). Behavioral dynamics and influence in networked coloring and consensus. *Proceedings of the National Academy of Sciences*, 107(34), 14978–14982.
- Kearns, M. (2012). Experiments in social computation. *Communications of the ACM*, 55(10), 56–67.
- Kearns, M., Judd, S., Tan, J., & Wortman, J. (2009). Behavioral experiments in biased voting in networks. *Proceedings of the National Academy of Sciences*, 106(5), 1347–1352.

- Kearns, M., Judd, S., & Vorobeychik, Y. (2012). Behavioral experiments on a network formation game. In *Proceedings of the 13th ACM conference on electronic commerce* (pp. 690–704). ACM.
- Kearns, M., Suri, S., & Montfort, N. (2006). An experimental study of the coloring problem on human subject networks. *Science*, 313(5788), 824–827.
- LeBlanc, H. J., & Koutsoukos, X. D. (2012). Low complexity resilient consensus in networked multi-agent systems with adversaries. In *Proceedings of the 15th ACM international conference on hybrid systems: Computation and control*. HSCC'12 (pp. 5–14). New York, NY, USA: ACM.
- LeBlanc, H. J., Zhang, H., Koutsoukos, X., & Sundaram, S. (2013). Resilient asymptotic consensus in robust networks. *IEEE Journal on Selected Areas in Communications*, 31(4), 766–781.
- Leibbrandt, A., Ramalingam, A., Saaksvuori, L., & Walker, J. M. (2015). Incomplete punishment networks in public goods games: Experimental evidence. *Experimental Economics*, 18(1), 15–37.
- Mao, A., Dworkin, L., Suri, S., & Watts, D. J. (2017). Resilient cooperators stabilize long-run cooperation in the finitely repeated prisoner's dilemma. *Nature Communications*, 8, 13800.
- Mason, W., & Suri, S. (2012). Conducting behavioral research on amazon's mechanical turk. *Behavior Research Methods*, 44(1), 1–23.
- Matthew, M., Ramamohan, P., & Nicholas, W. (2009). Networked coordination: Effect of network structure on human subjects' ability to solve coordination problem. *American Politics Research*, 37, 899–920.
- Miller, J. H., & Moser, S. (2004). Communication and coordination. *Complexity*, 9(5), 31–40.
- Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies: A comprehensive introduction*. Princeton, NJ: Princeton University Press.
- Nay, J. J., & Vorobeychik, Y. (2016). Predicting human cooperation. *Plos One*, 11(5), e0155656.
- Olmstead, A. J., Viswanathan, N., Aicher, K. A., & Fowler, C. A. (2009). Sentence comprehension affects the dynamics of bimanual coordination: Implications for embodied cognition. *Quarterly Journal of Experimental Psychology*, 62(12), 2409–2417.
- Paolacci, G., Chandler, J., & Ipeirotis, P. G. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision Making*, 5(5), 411–419.
- Peled, N., Gal, Y., & Kraus, S. (2015). A study of computational and human strategies in revelation games. *Autonomous Agents and Multi-Agent Systems*, 29(1), 73–97.
- Rapoport, A., Chammah, A. M., & Orwant, C. J. (1965). *Prisoner's dilemma: A study in conflict and cooperation*, Vol. 165. United States of America: University of Michigan Press.
- Richerson, P. J., & Boyd, R. (2010). Why possibly language evolved. *Biolinguistics*, 4(2–3), 289–306.
- Shirado, H., & Christakis, N. A. (2017). Locally noisy autonomous agents improve global human coordination in network experiments. *Nature*, 545(7654), 370.
- Szamado, S. (2011). Pre-hunt communication provides context for the evolution of early human language. *Biological Theory*, 5(4), 366–382.
- Usevitch, J., & Panagou, D. (2018). Resilient leader-follower consensus to arbitrary reference values. In *Annual american control conference* (pp. 1292–1298).
- Vorobeychik, Y., Joveski, Z., & Yu, S. (2017). Does communication help people coordinate? *Plos One*, 12(2), 1–19.
- Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 440.
- Wunder, M., Suri, S., & Watts, D. J. (2013). Empirical agent based models of cooperation in public goods games. In *Proceedings of the fourteenth ACM conference on electronic commerce* (pp. 891–908). ACM.
- Zeng, W., & Chow, M.-Y. (2014). Resilient distributed control in the presence of misbehaving agents in networked control systems. *IEEE Transactions on Cybernetics*, 44(11), 2038–2049.
- Zhang, H., & Vorobeychik, Y. (2019). Empirically grounded agent-based models of innovation diffusion: A critical review. *Artificial Intelligence Review*, 52(1), 707–741.
- Zhang, H., Vorobeychik, Y., Letchford, J., & Lakkaraju, K. (2016). Data-driven agent-based modeling, with application to rooftop solar adoption. *Journal of Autonomous Agents and Multiagent Systems*, 30(6), 1023–1049.

Appendix: Game & Experiment Description

The Decentralized Coordination Game

In this experiment you will be playing a sequence of multiplayer coordination games. In each game, you will be randomly assigned to a node of a network - this will be the central node appearing on your screen, labeled **Me**. You will also be assigned a name (shown in the left upper part of the screen). In addition, you will see a few other nodes on the screen - these are the nodes of your neighbors in the network. A link between two nodes indicates that the corresponding players are neighbors. Neighbors will see each other's nodes on their screens and non-neighbors will not. For example, in Figure 1, you and Moe are neighbors and Moe can see your node, labeled Dan, on his screen. But Moe cannot see Sue's node as Sue is not a neighbor of Moe.

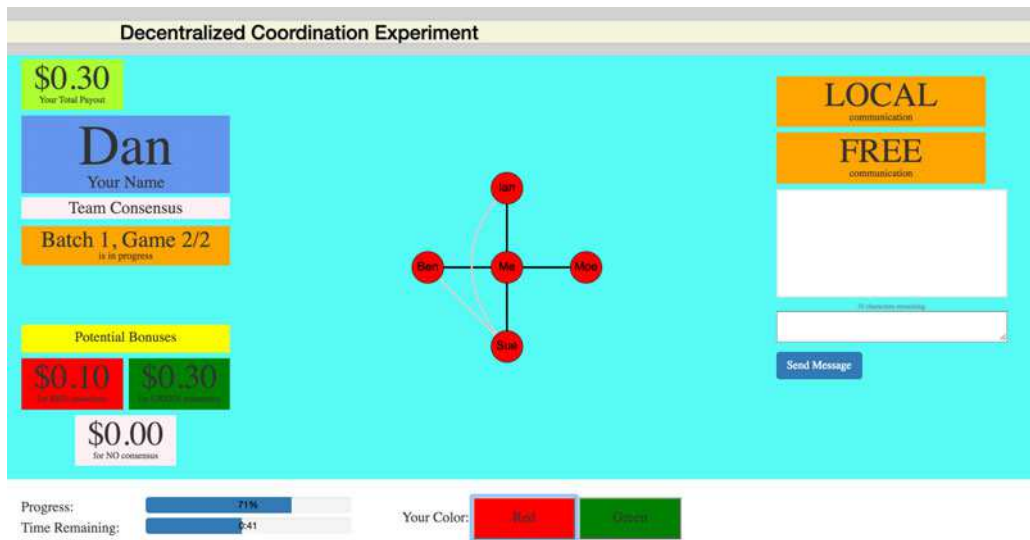


Figure 1. A view of the game screen. The name assigned to the player for this game is Dan. They have been assigned to team Consensus

Initially, all network nodes will be white. Throughout the game you will be choosing the color of your node, either red or green, using the two buttons in the lower central part of the screen. In some games you will also be able to communicate with other players. Your objective will depend on which team you are assigned to in a particular game.

Consensus and No-Consensus Teams

In each game you may be assigned to one of two teams: **Consensus** or **No-Consensus**. The team that you are assigned to in a particular game is shown in the left part of the screen just below your assigned name. However, the team assignments of other players will not be given to you, though you may be able to figure them out by observing other players' behavior or through communication. Note that in some games there may be no players assigned to the No-Consensus team, but this will not be disclosed at the start of the game.

If the Consensus team reaches a consensus on a color, i.e. a state where all nodes of the Consensus team in the entire network (not just those in your neighborhood) have the same color, they win and the No-Consensus team loses. Thus, the objective of the players on the Consensus team is to coordinate among themselves and reach a consensus before the game clock expires. **Notice that consensus requires that only the Consensus team members have the same color.**

If the game clock expires without the Consensus team members reaching a consensus on a color, they lose and the No-Consensus team wins. If you are a member of the No-Consensus team, your objective is to disrupt the coordination efforts of the Consensus team and stop them from reaching a consensus. Note that having a different color from other players will not be enough: only members of the Consensus team need to agree on a color for that team to win.

The actions available to players of both teams in completing their respective objectives are changing one's color and, in some games, sending messages to other players.

Game Termination, Outcomes & Bonus Payments

The allotted time for each game is 60 seconds and you will be able to change the color of your node at any point during the game continuously. A game will terminate as soon as the Consensus team reaches a consensus on a color (that is, as soon as all Consensus team members choose the same color). Bars showing the remaining time and game progress (how close to reaching a consensus the Consensus team is) are located in the lower left corner of the game screen.

Games can have one of three outcomes: **no consensus**, **red consensus**, and **green consensus**. The potential bonus payments for each outcome are shown in the lower left part of the screen. In Figure 2, for example, Dan will receive a bonus payment only if no consensus is reached. In general, the Consensus team members will receive positive bonus for a consensus outcome and no bonus when consensus is not reached. On the other hand, the No-Consensus team members will receive a positive bonus if the Consensus team does not reach a consensus and a smaller bonus or no bonus at all if the Consensus team reaches a consensus. Your total bonus acquired will be shown in the upper left corner. After the end of each game, the bonus payment you receive for that particular game will be shown next to it, until the next game begins.

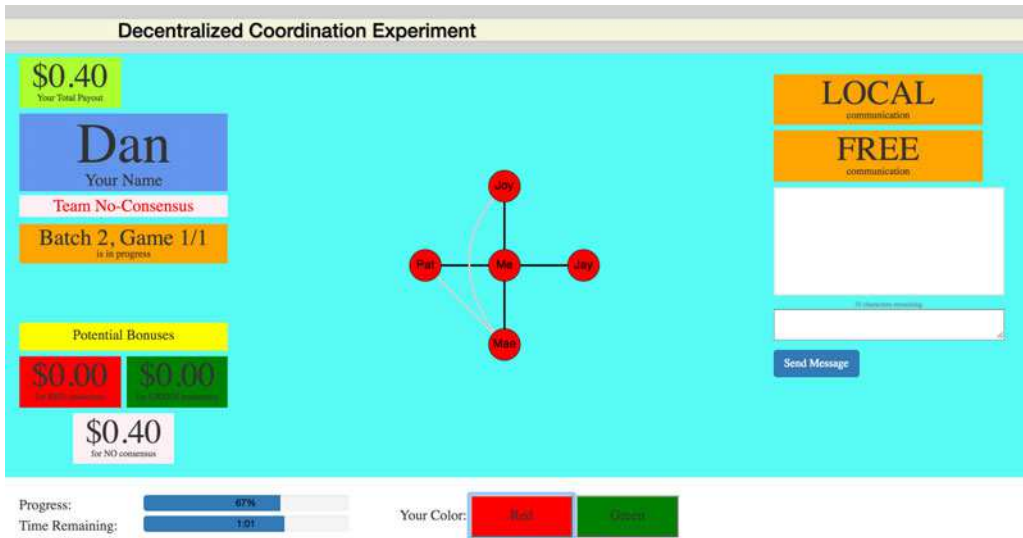


Figure 2. In this game, Dan is a member of team No-Consensus. He receives positive bonus (\$0.40) only if team Consensus does not reach consensus.

Note that your potential bonus payments may vary from game to game. In addition, within the same game potential bonuses may be different for different players, even if those players are in the same team. For example, in a given game the potential bonuses of Player 1 may be \$0.30 for red, \$0.10 for green, and \$0.00 for no consensus, while those of Player 2 may be \$0.10 for red, \$0.30 for green, and \$0.00 for no consensus. In such a case Players 1 and 2 have conflicting incentives for their color choice.

Also note that even when all nodes shown on your screen have the same color, it is possible that the Consensus team has not yet reached a consensus. There might be nodes of other Consensus team members in the network that are not shown on your screen, but are colored differently. The game progress bar located in the lower left corner of the screen is a better indicator of how close the Consensus team is to reaching a consensus.

Communication

In some of the games you will also have the opportunity to communicate with other players through a chat interface appearing on the right side of the screen. Note that there might be games in which you can only see messages sent by other players, but not be able to send any messages yourself.

Unconstrained vs. Constrained Communication

In the games with **unconstrained** communication, you will be able to send arbitrary messages to other players, as long as their length does not exceed the allowed characters limit per message. The number of remaining characters in a message will be shown just above the chat input box. There will also be a limit on the total length of unconstrained messages throughout a single game. For the current set of experiment sessions, you will be able to send unconstrained messages of up to 10 characters and your total communication throughout a single game may not exceed 50 characters. Thus, you will be able to send up to five 10-character messages, or a greater number of shorter messages whose total length does not exceed 50 characters.

In the games with **constrained** communication, you will be able to send pre-defined messages to other players, informing them of the status of your neighborhood. These messages will be of the format "RED:x/GREEN: y" and they will indicate that x of your neighbors have chosen red and y of them have chosen green. There will be a limit on the number of constrained messages you may send throughout a single game. The number of remaining constrained messages will be shown just next to the 'Report Neighborhood Color Counts' button.

You will know that you are in a game with **unconstrained** communication allowed, by the existence of a message input box. In games with **constrained** communication you will only see a button allowing you to send a constrained message.

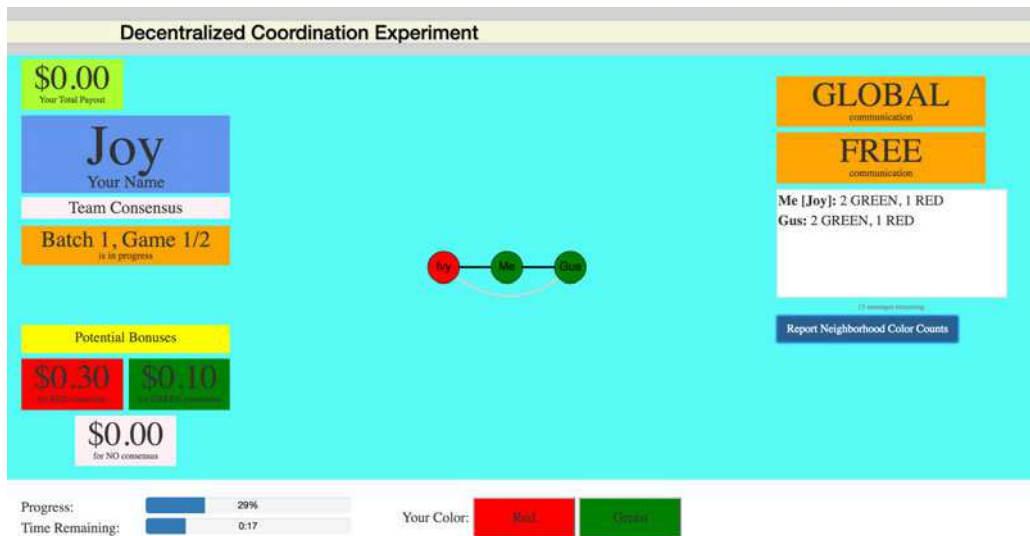


Figure 3. Game in which communication is allowed. Joy can send constrained messages globally (not just to her neighbors).

Local vs. Global Communication

Communication may be of two different scopes, **local** and **global**. In games with **local** communication, your messages can only be seen by your neighbors. On the other hand, in games with **global** communication, your messages can be seen by all players, even those that are not your neighbors. The scope of communication will be indicated in the right upper part of the screen, just above the chat box.

Cost of Communication

In some of the games in which communication is allowed, communication will be **cost-based**. By this we mean that each message you send reduces your potential bonus payments by a small fraction. Note that, if an outcome with positive bonus payment for you is not reached, you are not charged for any messages that you have sent during that particular game (i.e. your bonus payment, regardless of the outcome of each game and the level of your communication with other users, will never be negative). Information related to the costs of messaging will be indicated in the right upper part of the screen, just above the chat box.

Different Players May See Colors Differently

While our experiment application preserves consistent internal encoding of the colors, the way colors are shown on your screen may differ from how colors are shown on another player's screen. Figures 4 and 5, which show screens of two players with neighboring nodes in a given game, illustrate this. Mae and Sky are neighbors and they have selected the same color. However, Mae sees her and Sky's nodes as **green**, while Sky sees their nodes as **red**. We call this feature *anonymization of colors*.

This feature of the experiment application has several implications. Suppose that consensus was reached in the previous game and on your screen the consensus color was shown as **red**. If everyone chooses **red** in the next game would this immediately lead to another consensus? The answer is NO. For example, if in the game shown in Figures 4 and 5, Mae chooses **red** then she and Sky will have different colors which will not lead to a consensus.

What if in the next game everyone chooses the last game's consensus color as they saw it on their screen? Would this lead to an immediate consensus? The answer is again NO. How you see colors on your screen might also differ from game to game. For example, suppose that consensus was reached in the game shown in Figures 4 and 5 and that on Mae's screen the consensus color was shown as **green**, while on Sky's screen it was shown as **red**. In the next game it is possible that Mae and Sky see node colors in the same way. But then if they choose the last game's consensus color as they saw it, Mae will end up choosing **green** and Sky **red**, which will not lead to a consensus.

The reason behind the anonymization of colors feature of the experiment application is to prevent players from reaching a trivial consensus every time just by using the consensus color from the previous game. Instead, to reach a consensus players need to coordinate and communicate within each single game.

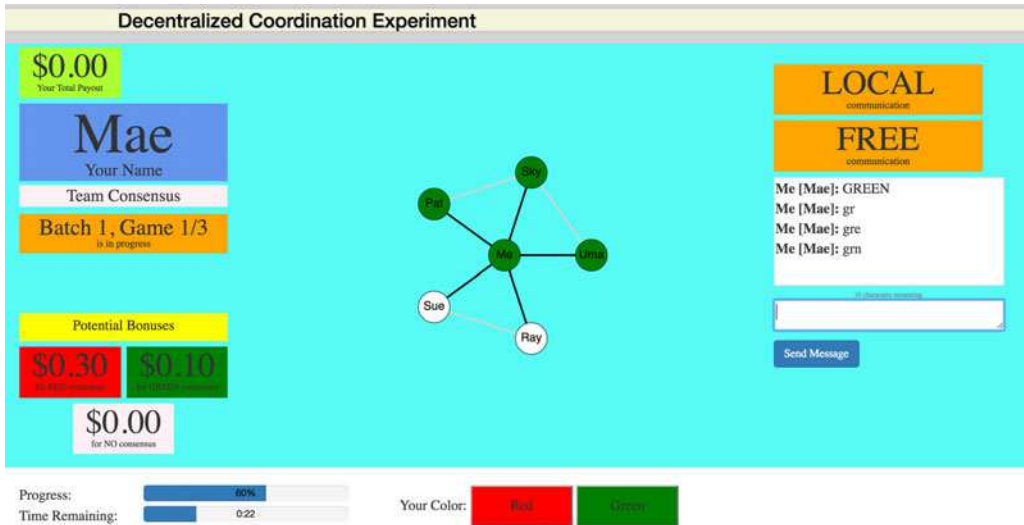


Figure 4. Mae's game screen.

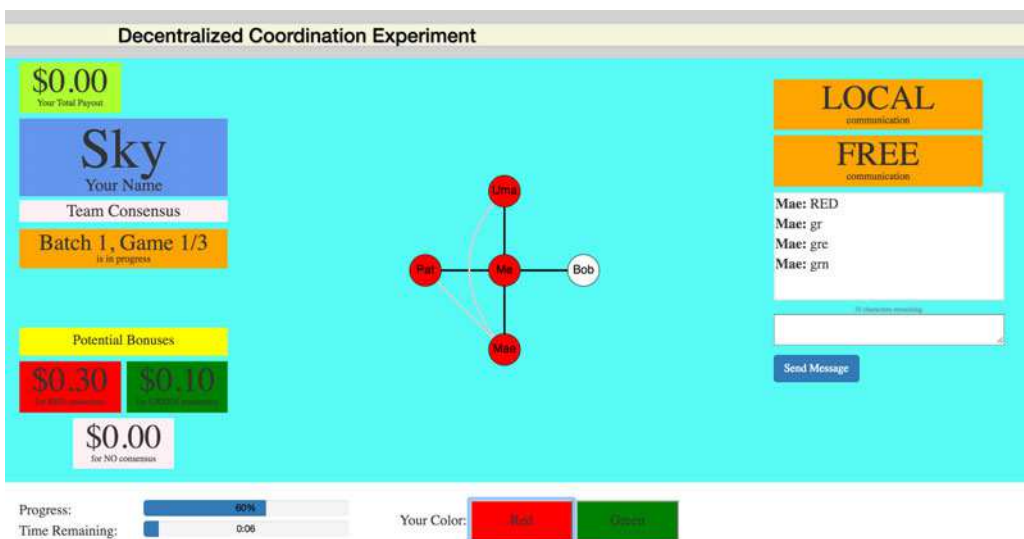


Figure 5. Sky's game screen.

Effects on Communication

Because you may see colors differently from how other players see them, the messages that refer to colors need to be converted appropriately for every recipient. Our experiment application is able to convert three types of messages.

- (1) If you refer to a color using the corresponding full English word our application will automatically convert your message depending on how you and the message recipients see color on your screen. For example, in the game shown in Figures 4 and 5 Mae first sent out the message 'GREEN'. Since Sky sees Mae's green color as red, in her chat box Mae's first message is shown as 'RED'.

(2) Our application also supports a shorthand notation for referring to colors using two-character codes.

Color	RED	GREEN
Code	\r	\g

For example, if Mae sends the message “Sky \g” on his screen this message will be converted to “Sky GREEN”, while on Sky’s screen this message will be shown as “Sky RED”, since Mae and Sky see colors differently.

(3) Constrained messages are automatically converted.

Our application, however, DOES NOT support the conversion of abbreviated color messages like “gr”, “grn”, or “r”. For example, suppose Mae (Figure 4) sends the message “gre” attempting to convince other players like Sue or Ray to choose that color. This message will not be converted and will be misleading for players like Sky (Figure 5) who see colors differently, i.e. Sky may think that Mae is asking her to change her color, which is not the case.

To avoid confusion and ensure that unconstrained messages do not negatively affect the chances for reaching a consensus, whenever you wish to refer to a color you should use full color words or, better yet, the two-character color codes ‘\r’ and ‘\g’.

Experimental Setup

A full experiment session will usually consist of 3-5 practice games and 50-65 regular games and incentives and communication parameters will vary across games. For each regular game you will receive a base payment of \$0.15. In addition, for each game in which you reach an objective, you will earn a bonus between \$0.10 and \$0.40.

Games will be grouped in batches of one or more games. The current batch and game numbers will be shown on the left part of the game screen. Throughout all games of a particular batch, your assigned name and team will remain the same. But, when the new batch of games starts, you may be assigned a different name and/or team.

Every game will be played by exactly 20 Consensus team players and some number of No-Consensus players. The size of the No-Consensus team may vary from batch to batch, usually in the range 0-5. We will usually have up to 30 experiment participants to make sure that individual worker’s technical issues (connection loss) or inactivity do not prevent us from running the full experiment session. If there are more workers online than the number of required participants for each batch, we will rotate participation, so that everyone gets the chance to play. The rotation will be performed at the start of each batch of games. You will never have to pause for more than one batch of games in a row, but bear in mind that batches may differ in the number of games. You will still receive the base payment of \$0.15, even for games that you are not participating in.

IMPORTANT: Please do not work on other tasks while participating in the experiment. Being able to quickly respond to color changes and messages of other participants greatly increases the chances for reaching one of the objectives, and with that, a higher bonus reward. In case a participant does not make at least one color choice in more than 1 game in which they were assigned to the Consensus team, they will be kicked out of the experiment, as such a behavior prevents the reaching of a consensus regardless of what the other Consensus team members do.

Experiment Session on [Day, Month Date]

This experiment session will consist of 5 practice games and 18 regular games organized in batches of size 1 (18 batches in total). You will receive a base payout of \$0.15 per game which will be included in the HIT base reward of \$2.70 (18 x \$0.15). Depending on the particular game incentives and your performance you can earn a bonus payment of up to \$0.40. (Note that in some games the maximum bonus payment you can earn could be as low as \$0.20.)