Learning to Mitigate Adversarial Threats in Mixed-Motive Societies

Melissa Kretchmer , Jacob W. Crandall
Computer Science Department, Brigham Young University, Provo, UT, USA
melissa.a.kretchmer@gmail.com, crandall@cs.byu.edu

Abstract

Human societies thrive on cooperation, yet differing interests often lead to the formation of adversarial coalitions—groups that actively work against societal well-being. Addressing these coalitions quickly is critical, as they can grow stronger and more difficult to mitigate once they become entrenched. Prior work has illustrated that it is difficult for autonomous agents to learn how to mitigate these strategies in mixed-motive environments. To this end, in this paper, we study mechanisms to help agents evolve strategies that counter adversarial threats in the Junior High Game (JHG), a mixed-motive environment that models coalition dynamics. Specifically, we investigate the impact of three mechanisms on the ability of self-interested agents to learn to mitigate adversarial coalitions in the JHG: fear-based coordination, communication, and alternate training conditions. Our results show that, while communication and fear-based coordination lead to positive improvements in some conditions, training conditions have the most consistent impact on how well our agents to learn to both mitigate adversarial threats and cooperate when no adversarial threats are present.

1 Introduction

A key driver of human success is our ability to work together [Henrich, 2017]. However, even though people *can* effectively work together, different goals (mixed motives) and inadequate coordination often hinder this ability, leading societies to fragment. While this is not inherently a problem, groups with directly conflicting goals, especially ones that oppose the goals of society as a whole, can substantially inhibit societal prosperity if allowed to grow strong. If left unmitigated, so-called *adversarial coalitions* can lead to social disruption, economic instability, and political destabilization. For example, the Sinaloa Cartel (a real-world adversarial coalition) is one of the world's largest drug trafficking organizations and has fueled addiction, violence, political corruption, and hindered economic development in both the U.S. and Mexico [Gootenberg, 2010].

In this paper, we consider how AI agents, as individual members of society, can learn to work with others to mitigate adversarial coalitions. This is valuable because (1) learning to work with others to mitigate adversarial coalitions in mixed-motive settings is a core capability for agents in multi-agent systems and (2) such agents may help humans counter real-world adversarial threats.

While coordination is often required to pool sufficient resources to counter adversarial coalitions, learning to coordinate can be challenging in mixed-motive settings [Skaggs *et al.*, 2024]. AI algorithms often fail to learn to coordinate, even in well-defined teaming scenarios [Fosong *et al.*, 2023], due to several factors including (1) non-stationary environments, as agents continually adapt their strategies; (2) large strategy and state spaces, driven by the combinatorial growth of joint actions; and (3) multiple equilibria, including inefficient local optima that often trap learning processes. Addressing these challenges requires learning architectures that support robust coordination under adversarial pressure.

Toward this end, in this paper, we study how well several algorithmic mechanisms help self-interested agents evolve strategies that mitigate adversarial threats in the Junior High Game (JHG) [Skaggs *et al.*, 2024], a strategic mixed-motive game that models coalition dynamics. We consider three mechanisms: (1) *fear*, a threat-assessment mechanism inspired by behavioral psychology that helps agents identify and respond to potential dangers [Smith and Lazarus, 1990; Fanselow and Lester, 1988; Schauer and Elbert, 2010]; (2) *cheap talk* [Charness and Grosskopf, 2004; Crawford and Sobel, 1982; Crawford, 1998; Farrell and Rabin, 1996; Farrell, 1987; Green and Stokey, 2007; Sally, 1995; Balliet, 2010], which can help agents to share intentions and coordinate strategies; and (3) *alternative training processes*, in which—motivated by the concept of homophily [Jackson, 2019]—we explore how training agents in the presence of like-minded associates impacts coordination against adversarial coalitions. Similar to centralized training and decentralized execution (CTDE) in team-based scenarios [Amato, 2024], this potentially allows agents to learn

group norms while optimizing for individual success.

Our results show that, while fear and communication mechanisms have some impact on the ability of agents to learn to work together to mitigate adversarial threats, the training process is by far the most influential. Training agents with like-minded peers promotes non-myopic behavior that both counters adversarial coalitions and supports cooperation in their absence. These results have important implications for designing agents that can learn to mitigate adversarial threats.

2 Problem Domain and Prior Work

To study how self-interested agents can (collectively) learn to mitigate adversarial coalitions, we require a testbed that models power dynamics and the individuals' choices to join, ignore, or mitigate such coalitions. The Junior High Game (JHG) [Skaggs *et al.*, 2024] models key aspects of societal dynamics relevant to adversarial coalitions. This section overviews the JHG and prior work on (a) coordinated threats in the JHG and (b) failed attempts to learn to counter them.

2.1 Evaluation Environment: The Junior High Game

Many game-based environments model strategic decision-making, but most lack the features needed to study adversarial coalitions. Traditional social dilemmas often restrict interactions, assume shared goals, or do not support coalition formation [Skyrms, 2003; Axelrod, 1984; Lerer and Peysakhovich, 2017; Fosong *et al.*, 2023; Bishop *et al.*, 2020; Baek *et al.*, 2022; Rosero *et al.*, 2021; Fehr and Gächter, 2002; Walton-Rivers *et al.*, 2019; Bard *et al.*, 2020; Shi *et al.*, 2022; Biely *et al.*, 2007]. Without these capabilities, they fail to capture complexities of adversarial coalitions and how they can be mitigated. Diplomacy [Kraus and Lehmann, 1988; Paquette *et al.*, 2019; De Jonge *et al.*, 2019; Dafoe *et al.*, 2021; Meta AI *et al.*, 2022] provides a rich setting for studying coordination and coalitions, but as a zero-sum game, all players eventually turn against each other.

While Welfare Diplomacy [Mukobi *et al.*, 2023] offers a general-sum alternative, we use the Junior High Game (JHG) [Skaggs *et al.*, 2024] due to its advantages for studying adversarial coalitions. The JHG explicitly models power dynamics, where coalition strength depends on size and composition, and gives players clear strategic choices including joining, ignoring, or resisting an adversarial group. As in real-world societies, the JHG forces individuals to make decisions given a large and flexible strategy space. The game is punctuated by conflicting interests, high interdependence between players, and power asymmetry across players. It can also be played by an arbitrarily large number of individuals (although we consider ten in this paper). These elements make the JHG a strong choice for investigating the formation and mitigation of adversarial coalitions.

Full descriptions of the JHG are provided by Skaggs *et al.* [2024] and Skaggs and Crandall [2025]; here we give a brief overview. The JHG is a strategic network game in which players seek to maximize their *popularity*, which represents wealth and power in the game. Each player starts with a popularity of 100 which changes over time based on token exchanges between the players. In each round, each player has N tokens of which it can (1) *keep* (to maintain, but not increase, its own popularity and defend itself), (2) *give* to another player (to increase that player's popularity), or (3) use to *attack* another player (which steals popularity from that player). After token allocation, the popularity of each player is updated and a new round begins with those new popularities. Importantly, allocations from more popular players have greater impact, thus reinforcing power asymmetries that arise due to historical token allocations.

In the JHG, all players in a society can increase in popularity by giving tokens to each other. However, players can rise even faster by either attacking others (potentially in groups to overcome defenses and the threat of retaliation) or not giving as much as they receive.

Algorithm 1 CAB token allocations in round τ for player i (adapted from Skaggs *et al.* [2024]).

procedure ALLOCATETOKENS($\mathcal{G}_i(\tau)$, Θ_i)
  $\mathbf{c}(\tau) \leftarrow$ detectCommunities($\mathcal{G}_i(\tau), \Theta_i$)
  $C_i^*(\tau) \leftarrow$ getDesiredCommunity($\mathbf{c}(\tau), \mathcal{G}_i(\tau), \Theta_i$)
  return distributeTokens($C_i^*(\tau), \mathbf{c}(\tau), \mathcal{G}_i(\tau), \Theta_i$)
end procedure

2.2 Adversarial Coalitions in the JHG

Many forms of adversarial coalitions can emerge in the JHG. One especially effective example is the CAT (Conspiring Autonomous Thieves) strategy defined by Skaggs *et al.* [2024]. A CAT behaves in the JHG as follows:

- Round $t = 1$: The CAT keeps all of its tokens. This both protects it against potential attacks from other players and signals its identity to other CATs.
- Rounds $t > 1$: The CAT uses all of its tokens to attack the weakest non-CAT player who has non-zero popularity. This behavior again serves two purposes: (1) it enables coordinated attacks (by all attacking the same weak player) that can overpower individual defenses and (2) it continues to signal to other CATs its identity. If no viable victims exist, the CAT keeps its tokens.

Despite its simplicity, this strategy is highly effective. As few as two CATs can dominate a population of eight other players if those players fail to coordinate.

2.3 Community Aware Behavior

While CATs can be effective, this strategy does not represent the only way to play the JHG. The CAB (*C*ommunity-*A*ware *B*ehavior) algorithm, also created by Skaggs *et al.* [2024], offers a more general-purpose algorithm for playing the JHG based on community formation. In each round, a CAB agent, summarized in Algorithm 1, takes as input observations of game play

Agent	Agent Popularity	CAT Popularity	% Mitigated
eCAB	71.77	0.39	99.17
hCAB	56.78	48.03	50.33

Table 1: Average ending popularities of CAB agents in games with CATs. The percent mitigated is the proportion of CAT agents that end the game with popularity less than 5.

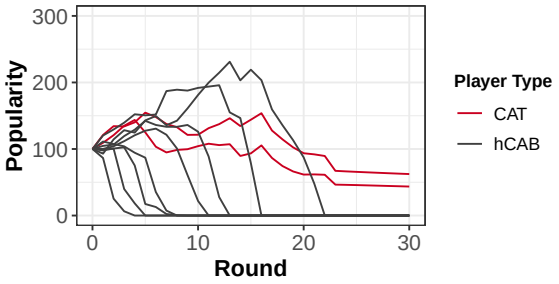

Figure 1: Popularity over time in an example game involving eight randomly selected hCABs and two CATs.

$\mathcal{G}_i(\tau)$ and a parameter set Θ_i. It then uses a three-step process to determine its token allocations: (1) identify existing communities based on prior token allocations, (2) select a desired community to belong to, and (3) allocate tokens to both form the desired community and to make this community successful.

The parameter set Θ determines how CAB agents identify, select, and carry out community formation. Different combinations of parameters can cause the agents to be more aggressive, collaborative, isolated, etc. Prior work has studied two methods for learning parameter values. First, Skaggs *et al.* [2024] studied the impact of learning parameter values through genetic evolution. In this paper, we refer to these agents as *evolved CABs* or eCABs. Second, subsequent work studied tuning CAB parameters to mimic human behavioral data in the JHG [Skaggs and Crandall, 2025]. We refer to these CAB agents as *human-tuned CABs* or hCABs.

Table 1 shows that neither eCABs nor hCABs thrive in societies with CAT agents. Prior work [Skaggs *et al.*, 2024] has shown that, while eCABs successfully foster cooperation with other eCABs and human players, the popularity of all eCAB agents is typically driven to zero by CATs when they are not trained with CATs. On the other hand, when they are trained with CATs, eCABs learn to keep about 90% of their tokens. This learned strategy results in the popularity of CATs being driven to zero in nearly all cases (Table 1), meaning the eCABs have successfully mitigated the CATs. However, this defensive behavior reduces overall societal prosperity, with the average popularity of eCABs falling from 100 to about 72 over 30 rounds.

hCABs, while behaviorally similar to humans [Skaggs and Crandall, 2025], are even less successful than eCAB on average in eliminating CATs (Table 1). In many cases, they are destroyed by CATs. A common scenario is illustrated in Figure 1, which displays the popularities of the players over time. The figure shows that the two CATs pick off hCAB agents one by one until the popularity of each hCABs is reduced to zero.

Symbol	**Usage**
$\theta_{\text{fearAggression}}$	How much the agent fears aggression in others
$\theta_{\text{fearGrowth}}$	How much the agent fears growth in others
θ_{fearSize}	How much the agent fears larger groups
$\theta_{\text{fearContagion}}$	How much the agent fears players that others fear
$\theta_{\text{fearThreshold}}$	Threshold for triggering a fear response
$\theta_{\text{fearPriority}}$	Priority of attacking out of fear
$\theta_{\text{trustRate}}$	Determines how quickly the agent builds trust
$\theta_{\text{distrustRate}}$	Determines how quickly the agent loses trust
$\theta_{\text{startingTrust}}$	Specifies the agent's initial trust in other players
$\theta_{\text{wChatAgree}}$	Degree agreement impacts community selection
θ_{wTrust}	Degree trust impacts community selection
$\theta_{\text{wAccusations}}$	Degree accusations impact community selection

Table 2: The parameters Θ' used by SChMUSR in addition to the set Θ used by CAB agents [Skaggs *et al.*, 2024]. Each parameter takes on an integer value in the range $[0, 100]$.

While eCABS and hCABs often produce compelling behavior [Skaggs *et al.*, 2024; Skaggs and Crandall, 2025], these findings highlight a limitation of these algorithms. Specifically, the learning mechanisms of these algorithms (genetic evolution and mimicking human behavioral data) do not produce strategies that allow these agents to work together to consistently mitigate adversarial threats in the JHG. Thus, in the next section, we propose modifications to the CAB algorithm in an attempt to help these agents learn to work with others to mitigate adversarial threats.

3 SChMUSR

We consider three ways that eCAB agents can be modified to learn to mitigate adversarial threats. First, we consider changes to their internal mechanisms. Specifically, perhaps they would find collaborative methods for mitigating adversarial coalitions if their internal mechanisms allowed them to perceive *fear*. Second, we consider that eCABs might fail to learn to mitigate CATs because they cannot effectively communicate. Thus, our second modification is to give them the ability to communicate with each other via cheap talk (chat). Finally, we consider that perhaps eCABs could better learn to mitigate adversarial threats if we altered their *training processes*.

We encode these modifications into an extended eCAB agent we call SChMUSR (*S*trategic *Ch*atbot for *M*otivating *U*nified *S*ynergistic *R*esponse). SChMUSR adds the twelve additional parameters listed in Table 2 to the 30 parameters already encoded in the CAB algorithm. We present each of these modifications in turn.

3.1 A Fear Mechanism

A key challenge in threat mitigation is identifying which groups or individuals pose a danger, especially in dy-

namic and mixed-motive environments. In behavioral psychology, the feeling of being threatened can be triggered by threat severity, social context, proximity, inescapability, and unpredictability [Smith and Lazarus, 1990; Fanselow and Lester, 1988; Schauer and Elbert, 2010]. Thus, we adapt these ideas to create a threat assessment within our agents which we refer to as *fear*. Fear encodes the threat an external group or individual poses on an individual based on four factors: aggression, size, growth, and fear contagion. These factors are combined to determine the overall fear the agent perceives, thus providing a context-sensitive measure of danger.

Formally, SChMUSR's fear of another group G is:

$$F_G(t) = \frac{1}{4}\Big(\theta_{\text{fearAggression}} \cdot F_G^{\text{agg}}(t) + \theta_{\text{fearSize}} \cdot F_G^{\text{size}}(t) + \theta_{\text{fearGrowth}} \cdot F_G^{\text{growth}}(t) + \theta_{\text{fearContagion}} \cdot F_G^{\text{contagion}}(t)\Big) \quad (1)$$

Here, $F_G^{\text{agg}}(t)$ is the fear that a SChMUSR agent has toward G due to past attacks members of G have made, $F_G^{\text{size}}(t)$ is fear due to G being more powerful than SChMUSR's own group, $F_G^{\text{growth}}(t)$ is fear from group G quickly increasing in strength, and $F_G^{\text{contagion}}(t)$ is fear derived from expressions of fear by other agents. Details regarding each of these variables is provided in SM-2.1.

If $F_G(t) > \theta_{fearThreshold}$, then group G is considered a threat. SChMUSR has three options for responding to a threat: attack the threat (using tokens to steal), keep tokens to protect against further attacks, or ignore the threat (continuing with its previous behavior). Its choice depends on available coordination (via chat) or estimated behavior (if chat is unavailable). If others are likely to act and are strong enough, SChMUSR attacks the player in G who offers the best tradeoff between inflicted damage and expected popularity gain. If recently attacked directly or through a friend, SChMUSR may choose to keep tokens, governed by internal thresholds $\theta_{dPropensity}$ and θ_{fearD}. Otherwise, it will ignore it.

3.2 Chat Capabilities

In adversarial settings where coalitions can form and shift dynamically, effective communication can be critical for forming groups and coordinating behavior. Without it, agents must rely solely on behavioral cues, which are often ambiguous. Through chat, agents can explicitly signal intentions, negotiate alliances, and warn others about emerging threats. Thus, we give SChMUSR chat capabilities that support strategic coordination with other players in attempt to overcome adversarial threats.

Incorporating communication into learning agents is challenging but not without precedent. Cheap talk has been shown to promote cooperation in repeated games with humans [Charness and Grosskopf, 2004; Crawford and Sobel, 1982; Crawford, 1998; Farrell and Rabin, 1996; Farrell, 1987; Green and Stokey, 2007; Sally, 1995; Balliet, 2010], and even between humans and AI [Crandall *et al.*, 2018; Oudah *et al.*, 2018]. Cicero [Meta AI *et al.*, 2022] integrated chat into an agent for playing Diplomacy, though later analysis suggests its success stemmed more from game-playing strategy rather than from chat [Wongkamjan *et al.*, 2024]. This highlights the need to design agents that use communication to foster cooperation in mixed-motive settings.

Type of Message	Example Message
Group Formation	"Alpha, would you like to join our group?"
Token Allocation	"I am attacking Charlie with 3 tokens"
Accusations	"Bravo attacked me last round"
Fear Contagion	"I am afraid of Delta"
Agreement	"Yes" or "No"

Table 3: The five message types used by SChMUSR.

SChMUSR uses the five types of chat messages outlined in Table 3 to attempt to improve group formation, trust-building, threat identification, and threat response. SM-2.2 outlines when SChMUSR sends each message type and how it influences behavior.

3.3 Alternate Training Processes

Skaggs *et al.* [2024] learned parameter values for eCAB agents using genetic evolution, where fitness was based on performance in games with agents randomly selected from the gene pool. We refer to this training process as *Random*. In this paper, we consider two other training conditions: *Homogeneous* where the fitness of an agent is evaluated in games played exclusively with other agents that have the same parameter values (genes) and *Mixed* where half of the agents have the same genes, while the other half are selected at random from the pool of agents.

Training agents with similar peers is motivated by two ideas. First, humans often display homophily, wherein they tend to associate with similar others [Jackson, 2019]. While homophily is often viewed as having negative implications, it does allow individuals of like mind to learn together, leading to norms and behaviors that help behavior coordination (including mitigating threats).

Second, when agents have common interests, Centralized Training, Decentralized Execution (CTDE) [Amato, 2024] is known to improve coordination. Homogeneous training is designed to yield similar benefits in mixed-motive environments, enabling like-minded agents to explore joint strategies effectively.

All other training procedures were the same across conditions. Each 30-round training game included two CAT agents to simulate adversarial pressure. We used the same genetic evolution procedure as in [Skaggs *et al.*, 2024]. The size of the gene pool was 100 agents, each of which participated in 100 games per generation. Fitness was measured by average final popularity across all games played in the generation, and evolution, via selection and mutation, occurred over 200 generations.

4 Experiments

We conducted a series of experiments to evaluate how well the three SChMUSR modifications—fear, chat, and

training process—help agents (1) learn to mitigate adversarial coalitions in the JHG and (2) learn effective behaviors in non-adversarial settings.

4.1 Experimental Design

We evaluate four different variants of the SChMUSR agent, defined by whether or not they include the *fear* and *chat* mechanisms. These variants are: (1) no fear, no chat (note: these agents are essentially CAB agents); (2) fear, no chat; (3) no fear, chat; (4) fear, chat. In addition to these four SChMUSR variants, we also compare the three training conditions described in the previous section. They are (1) Random, (2) Homogeneous, and (3) Mixed. Combined, this creates 12 different agents (4 variants x 3 training conditions).

We measure the success of these agents based on how well they *eliminate* CATs (measured by CAT popularity at the end of each game and the percentage of CAT agents mitigated) and how well the SChMUSR agents *survive* (measured by how high their popularity is at the end of each game). As baselines, we also compare SChMUSR's performance to both hCABs and eCABs.

4.2 Evaluation Scenarios

To determine the ability of the various agents to both effectively mitigate adversarial threats and perform effectively in settings without these threats, we evaluate the agents in three scenarios. First, we evaluate the agents in games with adversarial threats. These games are played with eight SChMUSR agents and two CATs. Next, to evaluate the ability of SChMUSRs to perform effectively in societies without adversarial threats, we conducted games in self play and with agents designed to mimic human behavior (hCABs). Games evaluating self play were played with ten SChMUSR agents, while games played with agents that mimic human behavior had five SChMUSRs and five hCABs.

For each scenario, we evaluated the agents over 100 games. Parameterizations were randomly selected from the 25 best parameterizations after 200 generations.

5 Results

In this section, we compare how well the twelve SChMUSR variants learn to (1) mitigate adversarial threats and (2) cooperate in the absence of such threats.

5.1 Societies With Adversarial Threats

As summarized in Table 4 and Figure 2, many of the SChMUSR variants learned to mitigate CATs more effectively and consistently than baselines algorithms. These results show that the training condition makes the largest impact. An analysis of variance (ANOVA) confirms that the training condition has a statistically significant ($\alpha = 0.05$) impact on the ending popularity in this scenario ($F(2, 100) = 1122.8$, $p < 0.001$). A pairwise comparison (Tukey-HSD) shows that agents trained in the Homogeneous and Mixed conditions are significantly better off in games with CATs than SChMUSRs

Condition	SChMUSR	CAT	% Mitigated	Keep	Give	Steal
No Fear, No Chat (R)	81.49	3.05	95.56	0.81	0.18	0.02
No Fear, Chat (R)	85.00	0.52	99.09	0.80	0.18	0.02
Fear, No Chat (R)	72.63	2.41	96.83	0.83	0.16	0.02
Fear, Chat (R)	95.60	0.40	99.50	0.74	0.24	0.03
No Fear, No Chat (H)	149.04	5.58	93.43	0.44	0.52	0.04
No Fear, Chat (H)	180.89	1.86	97.06	0.34	0.62	0.04
Fear, No Chat (H)	146.51	3.03	96.26	0.42	0.53	0.05
Fear, Chat (H)	176.18	1.17	98.08	0.38	0.58	0.04
No Fear, No Chat (M)	136.31	3.24	96.09	0.48	0.43	0.03
No Fear, Chat (M)	153.25	1.91	97.67	0.45	0.52	0.04
Fear, No Chat (M)	135.41	4.27	94.88	0.49	0.47	0.04
Fear, Chat (M)	159.73	1.54	98.07	0.40	0.54	0.05

Table 4: Results, averaged over 100 games, for societies with CAT agents. R (Random), H (Homogeneous, and M (Mixed) denote the training condition. *SChMUSR* and *CAT* give average ending popularities, while *% Mitigated* is the proportion of CATs that end the game with a popularity score below 5.

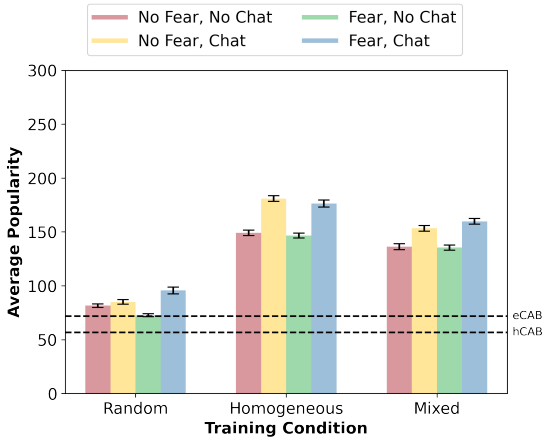

Figure 2: The average popularity of SChMUSRs in different training conditions when evaluated with CATs. The horizontal lines represent the average ending popularity of baseline agents. Error bars show the standard error of the mean.

trained in the Random condition ($p < 0.001$ for both comparisons). Agents trained in the Homogeneous condition are also significantly better than those trained in the Mixed condition ($p < 0.001$).

Figure 3 shows a typical game played by SChMUSRs in the homogeneous, no fear, no chat condition. While two SChMUSRs are initially attacked and reduced by the two CATs, the SChMUSRs subsequently attack and eliminate the CATs. Those SChMUSR agents that have not been reduced by the CATs then cooperate with each other and rise together (with the exception of one of the agents, which loses favor with the main group).

Table 4 and Figure 2 also show that chat capabilities substantially help SChMUSRs overcome CATs. The difference in final popularity is statistically significant ($F(1, 100) = 228.2$, $p < 0.001$). Chat-enabled agents have higher average ending popularity. CATs are also mitigated slightly more often. Variants without chat tend to keep a higher proportion of their tokens and give less than agents with chat. On the other hand, fear alone

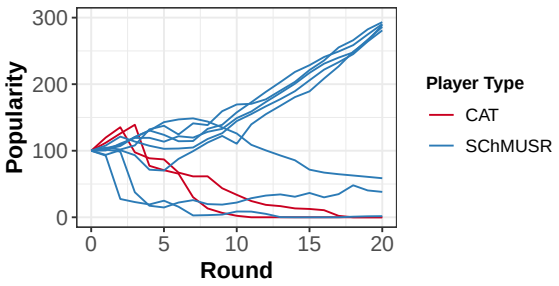

Figure 3: Popularity over time an example game with 8 SChMUSRs (Homogeneous, no fear, no chat) and 2 CATs.

Condition	SChMUSR	Keep	Give	Steal
No Fear, No Chat (R)	122.81	0.62	0.36	0.02
No Fear, Chat (R)	119.06	0.66	0.31	0.02
Fear, No Chat (R)	130.50	0.58	0.40	0.02
Fear, Chat (R)	143.18	0.56	0.41	0.03
No Fear, No Chat (H)	282.94	0.12	0.86	0.02
No Fear, Chat (H)	265.12	0.14	0.79	0.02
Fear, No Chat (H)	243.40	0.18	0.78	0.04
Fear, Chat (H)	255.01	0.19	0.78	0.03
No Fear, No Chat (M)	239.53	0.20	0.77	0.03
No Fear, Chat (M)	225.57	0.25	0.71	0.04
Fear, No Chat (M)	225.30	0.22	0.74	0.04
Fear, Chat (M)	218.35	0.24	0.70	0.05

Table 5: Results in self play evaluation in the Random (R), Homogeneous (H), and Mixed (M) conditions. The popularity and token distributions are averaged over all agents across 100 games for each simulation that was run.

does not cause any significant difference ($F(1, 100) = 0.3$, $p = 0.93$). However, there is a statistically significant interaction effect–fear and chat combined has a small impact in the Random training condition.

5.2 Societies Without Adversarial Threats

Effective agents must not only counter adversarial threats but also maintain cooperative behavior in their absence. To assess this, we evaluated SChMUSR variants both in self play and in mixed populations that include human-like agents (hCABs).

Table 5 and Figure 4 summarize the performance of the various SChMUSR variants in self play. These results show that the training condition had a statistically significant impact on the agents' ending popularity ($F(2, 100) = 1401.0$, $p < 0.001$), similar to when CATs were present. Agents trained in the Homogeneous condition perform the best, with those trained in the Mixed condition following closely behind. The fact that Homogeneous agents performed better than eCABs (also trained in societies with CATs) and about the same as hCABs in self play indicates that training with CATs does not substantially reduce how (Homogeneous) SChMUSR agents cooperate with each other. However, we do note that CATs do inflict a substantial toll on the population, as average ending popularities of Homogeneous SChMUSR agents are much higher when

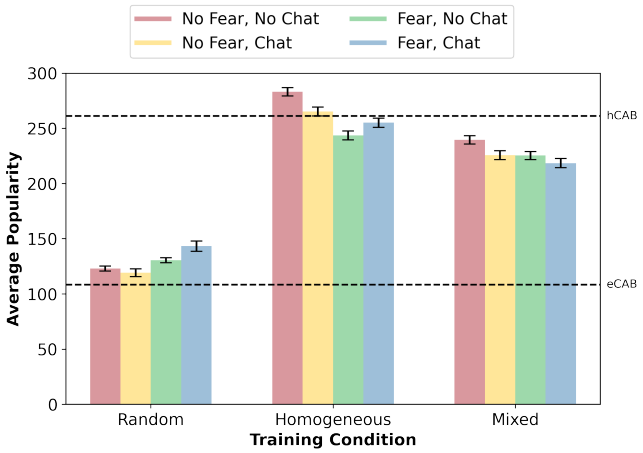

Figure 4: The average popularity of SChMUSR in self play under different training conditions. Error bars show the standard error of the mean.

Condition	SChMUSR	hCAB	Keep	Give	Steal
No Fear, No Chat (R)	113.88	263.10	0.59	0.37	0.03
No Fear, Chat (R)	110.97	231.92	0.64	0.32	0.04
Fear, No Chat (R)	120.04	245.84	0.56	0.40	0.04
Fear, Chat (R)	135.88	210.08	0.54	0.41	0.05
No Fear, No Chat (H)	200.28	321.02	0.17	0.79	0.04
No Fear, Chat (H)	229.78	243.50	0.15	0.81	0.04
Fear, No Chat (H)	184.02	285.29	0.21	0.73	0.06
Fear, Chat (H)	226.67	221.92	0.18	0.78	0.04
No Fear, No Chat (M)	174.33	305.91	0.24	0.71	0.04
No Fear, Chat (M)	205.02	201.80	0.24	0.71	0.05
Fear, No Chat (M)	170.29	300.89	0.25	0.70	0.05
Fear, Chat (M)	205.15	212.91	0.21	0.73	0.06

Table 6: Results in the hCAB evaluation for Random (R), Homogeneous (H), and Mixed (M) conditions. The popularity and token distributions are averaged over all agents across 100 games for each simulation that was run.

CATs are not present (compare Figures 2 and 4).

Fear and chat mechanisms did not yield overall gains in self play, though there was a statistically significant interaction between chat and training condition ($p < 0.001$). In the Homogeneous and Mixed training conditions, having no fear and no chat are the most effective in self play, while with Random training, having fear and chat produces the highest performance. These impacts are substantially less drastic, however, than the training condition. In short, the chat mechanism helps SChMUSRs eliminate CATs, but does not appear to substantially help them get along with each other.

To further evaluate SChMUSR in conditions without specific adversarial threats, we tested SChMUSRs alongside hCABs, which have behavior that is consistent with people in many regards [Skaggs and Crandall, 2025]. Thus, evaluating SChMUSR agents with hCABs potentially gives insight into how well these agent might perform when associating with people.

Table 6 and Figure 5 compare the popularity scores of

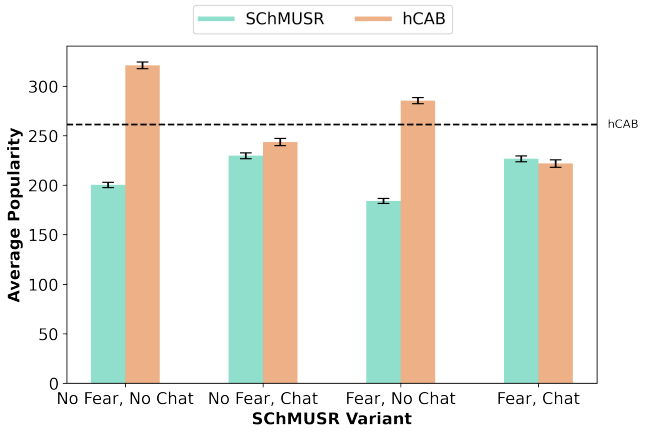

Figure 5: The average popularity of SChMUSR variants of the Homogeneous training condition side by side with the hCABs. Error bars show the standard error of the mean.

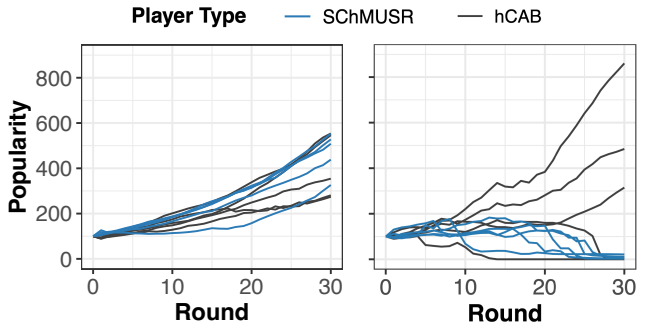

Figure 6: Popularity levels of players over time in two games played by 5 hCABs and 5 SChMUSRs (Homogeneous, no fear, no chat). (Left) A game in which players largely cooperate with each other. (Right) A game in which several hCABs eventually dominate.

SChMUSR agents to the hCAB agents in these games. To simplify this discussion, we consider primarily the Homogeneous condition because those SChMUSR agents performed the best in this and previously discussed scenarios. As shown in Figure 5, the hCABs do substantially better on average than the SChMUSR agents when they do not have the ability to communicate. However, when SChMUSRs can use chat, they perform somewhat similarly to the hCABs (on average) in all but the Random training condition (Table 6).

The reason that hCABs have higher average ending popularities than SChMUSRs when SChMUSRs do not chat is quite interesting. In many trials, the popularities of SChMUSRs and hCABs rise steadily together, as illustrated by the sample game shown in Figure 6 (left). However, in other games, one or more hCAB agents come to dominate the population. For example, Figure 6 (right) shows a scenario in which several hCABs eliminate the SChMUSRs (and some of the other hCABs as well) once they gain high power in the population. On the other hand, we have not observed a game in which a SChMUSR agent does this after gaining a large amount of power. As a result, the ending popularities of hCABs have a higher average popularity than those of SChMUSR agents when no chat is available.

hCABs occasionally eliminating SChMUSR agents is interesting on several levels. First, it shows a trade off between the abilities of hCAB and SChMUSR agents. While hCABs have interesting capabilities that allow them to outperform SChMUSR agents in head to head play, hCABs are not nearly as effective as SChMUSRs at eliminating CATs. Second, these results show that, in a sense, hCAB agents become another kind of adversarial threat once they become strong. However, SChMUSRs (without chat) were frequently unable to handle this threat. In fact, eliminating such a threat (which only becomes a threat once the agents become strong) is extremely difficult. The best way to do so is to avoid allowing agents to become strong in the first place. The fear mechanism was designed to stop this. However, the training process resulted in SChMUSR agents not learning to use it in this way.

6 Conclusion

This paper explored how AI agents can learn to mitigate adversarial threats. Using the Junior High Game (JHG) as a test environment, we evaluated how fear and chat mechanisms as well as several training processes impact whether SChMUSR agents learn to effectively mitigate adversarial threats. Our results show that the training process has the most substantial impact in this regard. The chat mechanism is also helpful in some situations, while the fear mechanism has little positive impact.

Because coordination among agents is critical for countering adversarial threats quickly, like-minded agents should be trained together to co-evolve behavior norms, as is done in the Mixed and Homogeneous training conditions presented in this paper. Without such training, agents tend to learn self-preserving strategies that produce less desirable social outcomes.

While we initially hypothesized that our fear mechanism would help SChMUSRs manage adversarial risks, that was not the case. Fear, combined with chat, did seem to have a small positive impact in the Random training condition, but it was not otherwise beneficial. We suspect that it may have the potential to help mitigate some forms of adversarial threats, such as those of high power hCABs shown in the last set of experiments we analyzed (e.g., Figure 6-right). However, the training scenarios we used did not result in agents learning to do so. Future work could study how alternative training experiences (such as training with hCABs) might benefit from fear. These and other future work can better help us learn how to create agents that can effectively learn to mitigate adversarial threats in mixed-motive societies.

References

C. Amato. An introduction to centralized training for decentralized execution in cooperative multi-agent re-

inforcement learning. *arXiv:2409.03052*, 2024.

Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.

In-Chang Baek, Tae-Gwan Ha, Tae-Hwa Park, and Kyung-Joong Kim. Toward cooperative level generation in multiplayer games: A user study in overcooked! In *2022 IEEE Conference on Games (CoG)*, pages 276–283. IEEE, 2022.

Daniel Balliet. Communication and cooperation in social dilemmas: A meta-analytic review. *The journal of conflict resolution*, 54(1):39–57, 2010.

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020.

Christoly Biely, Klaus Dragosits, and Stefan Thurner. The prisoner's dilemma on co-evolving networks under perfect rationality. *Physica D: Nonlinear Phenomena*, 228(1):40–48, apr 2007.

Justin Bishop, Jaylen Burgess, Cooper Ramos, Jade B Driggs, Tom Williams, Chad C Tossell, Elizabeth Phillips, Tyler H Shaw, and Ewart J de Visser. Chaopt: a testbed for evaluating human-autonomy team collaboration using the video game overcooked! 2. In *2020 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6. IEEE, 2020.

Gary Charness and Brit Grosskopf. What makes cheap talk effective? experimental evidence. *Economics letters*, 83(3):383–389, 2004.

Jacob W. Crandall, Mayada Oudah, Tennom, Fatimah Ishowo-Oloko, Sherief Abdallah, Jean-François Bonnefon, Manuel Cebrian, Azim Shariff, Michael A. Goodrich, and Iyad Rahwan. Cooperating with machines. *Nature Communications*, 9(1):1–12, 2018.

V. Crawford and J. Sobel. Strategic information transmission. *Econometrica*, 50(6):1431–1451, 1982.

V. Crawford. A survey of experiments on communication via cheap talk. *Journal of Econ. Theory*, 78:286–298, 1998.

Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. Cooperative AI: machines must learn to find common ground. *Nature*, 593(7857):33–36, 2021.

Dave De Jonge, Tim Baarslag, Reyhan Aydoğan, Catholijn Jonker, Katsuhide Fujita, and Takayuki Ito. The challenge of negotiation in the game of diplomacy. In *Agreement Technologies: 6th International Conference, AT 2018, Bergen, Norway, December 6-7, 2018, Revised Selected Papers 6*, pages 100–114. Springer, 2019.

Michael Fanselow and Laurie Lester. A functional behavioristic approach to aversively motivated behavior: Predatory imminence as a determinant of the topography of defensive behavior. *Evolution and Learning*, 01 1988.

Joseph Farrell and Matthew Rabin. Cheap talk. *The journal of economic perspectives: a journal of the American Economic Association*, 10(3):103–118, 1996.

J. Farrell. Cheap talk, coordination, and entry. *The RAND Journal of Economics*, 18(1):34–39, 1987.

Ernst Fehr and Simon Gächter. Altruistic punishment in humans. *Nature*, 415(6868):137–140, 2002.

Elliot Fosong, Arrasy Rahman, Ignacio Carlucho, and Stefano V. Albrecht. Learning complex teamwork tasks using a given sub-task decomposition. *Adaptive Agents and Multi-Agent Systems*, page 598–606, 2023.

Paul Gootenberg. Blowback: The mexican drug crisis. *NACLA report on the Americas*, 43(6):7–12, 2010.

J. R. Green and N. L. Stokey. A two-person game of information transmission. *J. of Econ. Theory*, 135:90–104, 2007.

Joseph Henrich. *The secret of our success: How culture is driving human evolution, domesticating our species, and making us smarter*. Princeton University Press, Princeton, NJ, 2017.

M. O. Jackson. *The Human Network*. Pantheon Books, 2019.

Sarit Kraus and Daniel Lehmann. Diplomat, an agent in a multi agent environment: An overview. In *IEEE International Performance Computing and Communications Conference*, pages 434–435. IEEE Computer Society, 1988.

Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *ArXiv preprint arXiv:1707.01068*, 2017.

Meta AI, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Gabriel Mukobi, Hannah Erlebach, Niklas Lauffer, Lewis Hammond, Alan Chan, and Jesse Clifton. Welfare diplomacy: Benchmarking language model cooperation. *arXiv preprint arXiv:2310.08901*, 2023.

Mayada Oudah, Talal Rahwan, Tawna Crandall, and J. Crandall. How AI wins friends and influences people in repeated games with cheap talk. *AAAI Conference on Artificial Intelligence*, page 1519–1526, 2018.

Philip Paquette, Yuchen Lu, Seton Steven Bocco, Max Smith, Satya O-G, Jonathan K Kummerfeld, Joelle Pineau, Satinder Singh, and Aaron C Courville. No-press diplomacy: Modeling multi-agent gameplay. *Advances in Neural Information Processing Systems*, 32, 2019.

Andres Rosero, Faustina Dinh, Ewart J de Visser, Tyler Shaw, and Elizabeth Phillips. Two many cooks: Understanding dynamic human-agent team communication and perception using overcooked 2. *arXiv preprint arXiv:2110.03071*, 2021.

David Sally. Conversation and cooperation in social dilemmas: A meta-analysis of experiments from 1958 to 1992. *Rationality and society*, 7(1):58–92, 1995.

Maggie Schauer and Thomas Elbert. Dissociation following traumatic stress. *Zeitschrift für Psychologie*, 218, 01 2010.

Zhenyu Shi, Wei Wei, Matjaž Perc, Baifeng Li, and Zhiming Zheng. Coupling group selection and network reciprocity in social dilemmas through multilayer networks. *Applied Mathematics and Computation*, 418:126835, 2022.

J. B. Skaggs and J. W. Crandall. Modeling human behavior in a strategic network game with complex group dynamics. *arXiv*, 2025.

Jonathan Skaggs, Michael Richards, Melissa Morris, Michael A. Goodrich, and Jacob W. Crandall. Fostering collective action in complex societies using community-based agents. *International Joint Conference on Artificial Intelligence*, page 211–219, 2024.

Brian Skyrms. *The Stag Hunt and the Evolution of Social Structure*. Cambridge University Press, 2003.

Craig Smith and Richard S. Lazarus. Emotion and adaptation. In Lawrence A. Pervin, editor, *Handbook of Personality: Theory and Research*, pages 609–637. Guilford Press, New York, 1990.

Joseph Walton-Rivers, Piers R. Williams, and Richard Bartle. The 2018 Hanabi competition. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2019.

Wichayaporn Wongkamjan, Feng Gu, Yanze Wang, Ulf Hermjakob, Jonathan May, Brandon Stewart, Jonathan Kummerfeld, Denis Peskoff, and Jordan Boyd-Graber. More victories, less cooperation: Assessing cicero's diplomacy play. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12423–12441, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

1 Supplementary Material

This section contains supplementary materials, such as additional figures, technical details, or data.

Table 1: The set of parameters Θ used in the CAB and SChMUSR algorithm. Each parameter can take on an integer value in the range $[0, 100]$. Adjusting the parameters can produce vastly different behavior. The value column refers to the hand-coded value that was used in prior research done with the CAB agents. The * denotes parameters that are not found in the CAB agents, only SChMUSR agents

Symbol	Usage	Value
θ_{α}	Specifies the weight between positive and negative influence when computing	20
θ_{strength}	Specifies the agent's desired community strength as a percentage of the whole	70
θ_{modu}	Weight of modularity in scoring potential communities	100
θ_{target}	Weight of target collective strength in scoring potential communities	80
$\theta_{\text{prominent}}$	Weight of prominence in scoring potential communities	50
θ_{familiar}	Weight of familiarity in scoring potential communities	50
$\theta_{\text{prosocial}}$	Weight of prosocial in scoring potential communities	70
θ_{initialD}	Specifies the percentage of its tokens the agent keeps in round $\tau = 1$	20
θ_{dUpdate}	Specifies how quickly the agent adapts its keeping based on new attacks on itself	50
$\theta_{\text{dPropensity}}$	Specifies the degree to which the agent keeps tokens based on past attacks against it	50
θ_{fearD}	Specifies the degree to which the agent keeps tokens based on attacks on its friends	0
θ_{minD}	Specifies the minimum number of tokens the agent keeps in a round	20
θ_{pFury}	Percentage of tokens to use in a pillage attack	0
θ_{pDelay}	Number of rounds to wait at the start of the game before considering pillage	10
$\theta_{\text{pPriority}}$	The priority of pillage attacks	0
θ_{pMargin}	Determines the gain margin required before a particular pillage attack is considered	0
$\theta_{\text{pCompanion}}$	The amount of help the agent initially believes it will get in a pillage attack over the first 5 rounds	50
θ_{pFriends}	Determines whether an agent will considering pillaging a friend.	0
θ_{vMult}	Specifies how much vengeance to reciprocate	100
θ_{vMax}	Specifies the maximum number of tokens that can be used in a vengeance attack	100
$\theta_{\text{vPriority}}$	The priority of vengeance attacks	100
θ_{dfMult}	Determines how much to reciprocate in defend-friend attacks	100
θ_{dfMax}	Specifies the maximum number of tokens that can be used in a defend-friend attack	100
$\theta_{\text{dfPriority}}$	The priority of defend-friend attacks	90
$\theta_{\text{attackGGuys}}$	Determines whether the agent will consider defend-friend attacks against players that were not the first to attack and did not over-attack	0
$\theta_{\text{groupAware}}$	Determines whether the agent will consider defend-friend attacks against players that belong to communities that have more collective strength than its own.	0
θ_{Safety}	Specifies whether the agent will defend or attack if it cannot do both	0
$\theta_{\text{debtLimits}}$	Defines how much the agent will give to another player without reciprocation	25
$\theta_{\text{limitGive}}$	Sets limits how much the agent can give to a poor player	100
$\theta_{\text{fixedUsage}}$	Determines the percentage of tokens the agent gives uniformly to its group (as opposed to distributing based on past friendship)	50
$*\theta_{\text{trustRate}}$	Determines how quickly the agent builds trust	0
$*\theta_{\text{distrustRate}}$	Determines how quickly the agent loses trust	0
$*\theta_{\text{startingTrust}}$	Specifies the amount of trust the agent initially has for everyone	0
$*\theta_{\text{wChatAgreement}}$	Weight of the agreement of others in scoring potential communities	0
$*\theta_{\text{wTrust}}$	Weight of the trust of group members in scoring potential communities	0
$*\theta_{\text{wAccusations}}$	Weight of accusations in scoring potential communities	0
$*\theta_{\text{fearAggression}}$	How much the agent fears aggression in others	0
$*\theta_{\text{fearGrowth}}$	How much the agent fears growth in others	0
$*\theta_{\text{fearSize}}$	How much the agent fears larger groups	0
$*\theta_{\text{fearContagion}}$	How much the agent fears players that others fear	0
$*\theta_{\text{fearThreshold}}$	The threshold at which the agent's fear triggers a response	0
$*\theta_{\text{fearPriority}}$	The priority of attacking out of fear over other kinds of attacks	0

2 SChMUSR: Algorithmic Details

In this section, we provide algorithmic details for how fear is computed by SChMUSR and how it derives and uses chat messages.

2.1 Fear

SChMUSR's fear (Eq. **??** is based on four variables. In this subsection, we describe how $F_i^{\text{agg}}(t)$, $F_G^{\text{size}}(t)$, and $F_G^{\text{growth}}(t)$ are computed. Computation of $F_G^{\text{contagion}}(t)$ is described in the next subsection.

$F_i^{\text{agg}}(t)$ is calculated from the influence matrix of the JHG [5]. Let $\mathcal{I}_{i,j}(t)$ be the influence of player i on j in round t. To determine the amount of fear from aggression that comes from player i, we take the total negative influence (denoted $\mathcal{I}_{i,j}^{-}(t) = \max(0, -\mathcal{I}_{i,j}(t))$) player i has on others and divide it by the total influence player i has on others. Formally, the aggression value of player i is

$$F_i^{\text{agg}}(t) = \frac{\sum_j \mathcal{I}_{i,j}^{-}(t)}{\sum_j |\mathcal{I}_{i,j}(t)|}. \tag{1}$$

$F_G^{\text{agg}}(t)$ is the average of $F_i^{\text{agg}}(t)$ over all $i \in G$.

$F_G^{\text{size}}(t)$, which measures the fear a SChMUSR agent derives due to group G being more powerful than its own group, is calculated by comparing the collective popularity of group G to the agent's own perceived group G_{in}. Let $\mathcal{P}_G(t) = \sum_{j \in G} \mathcal{P}_j(t)$ be the collective popularity of group G in round t, where $\mathcal{P}_j(t)$ is the popularity of player j in round t. Then

$$F_G^{\text{size}}(t) = \max\left(0, \min(1, [\mathcal{P}_G(t)/\mathcal{P}_{G_{\text{in}}}(t)] - 1)\right). \tag{2}$$

Note that SChMUSR derives no fear from any group smaller than its own and treats any group larger than twice its size as if it were exactly twice its size.

$F_G^{\text{growth}}(t)$ measures the fear a SChMUSR agent has when another group G increases in strength more quickly than its own group does. This is calculated as the relative change in popularity over the last Δt rounds between the two groups. Formally,

$$F_G^{\text{growth}}(t) = \max\left(0, \min[1, \frac{\mathcal{P}_G(t) - \mathcal{P}_G(t - \Delta t)}{\mathcal{P}_{G_{\text{in}}}(t) - \mathcal{P}_{G_{\text{in}}}(t - \Delta t)} - 1]\right). \tag{3}$$

Note that SChMUSR derives no fear from groups that are growing in strength slower than its own, and any growth exceeding twice its own group's growth is treated as if it were twice the growth.

2.2 Chat

SChMUSR sends and processes the five types of chat messages listed in Table **??**. In this subsection, we describe how it uses these messages.

Group Formation. SChMUSR uses chat to help in group formation by proposing and agreeing or disagreeing to changes in group composition. Specifically, SChMUSR alters the group formation mechanisms used by CAB agents to track whether each member wants to be in a group with others. Chat agreement is represented as a matrix C. For C_{ij}, a value of 1 means player i wants player j in their group, a value of -1 means player i does not want player j in their group, and a value of 0 means it is unknown. C is updated anytime another player proposes a group or agrees or disagrees to a proposal. Additionally, C is decayed at a rate of 0.9 each round to favor more recent chat messages. C is used to update the community score for each community G that the agent is considering. Formally,

$$\text{Score}'_{\text{G}} = \left(\frac{\sum_{i,j \in G} C_{ij}}{|C|}\right) \cdot \frac{\theta_{\text{wChatAgreement}}}{100} \cdot \text{Score}_{\text{G}}$$

where G is the set of players the agent is considering for a group, and Score_{G} is the score for a group as computed by CAB agents.

Trust-Building. The trust aspect of SChMUSR bridges the gap between what players say and what they actually do. This form of trust has been shown to be important in decision making and in forming relationships [1, 2, 4, 3]. SChMUSR's trust in another player increases when that player consistently follows through on their stated intentions in chat. This trust value then influences how much SChMUSR relies on that player's future statements.

SChMUSR models the trust it has in all players with the vector $T = (T_0, \cdots, T_{|I|})$, where T_i corresponds to the trust SChMUSR has in player i. The trust value for each player can be anywhere between 0, representing no trust, and 1, representing full trust. Initially, $T_i = \theta_{startingTrust}/100$. As the game progresses, T_i is

increased when player i does what they say they will do, and decreased when they fail to do so. For example, when player i states that they will give SChMUSR x tokens and then SChMUSR actually receives at least x tokens from player i, then T_i is increased by $\theta_{trustRate}/100$. However, when player i states that they will give SChMUSR x tokens, but SChMUSR does not receive at least x tokens from player i, then T_i is decreased by $\theta_{trustRate}/100$. T_i is also decreased when another player j states that player i (a) did not give a stated amount or (b) stole from them (i.e., accusations). In these cases, it is updated by subtracting $(\theta_{distrustRate}/100)*(T_j)$ from T_i.

When evaluating how much the agent wants to be in a community G, the community score is updated as follows:

$$\text{Score}''_{\text{G}} = \left(\frac{\sum_{i \in G} T_i}{|C|}\right) \cdot \frac{\theta_{\text{wTrust}}}{100} \cdot \text{Score}'_{\text{G}}.$$

Threat Identification. Chat is also used by SChMUSR to enhance threat identification through fear contagion. This mechanism analyzes messages exchanged throughout the game to determine perceived threats. When SChMUSR's fear level (described in the previous subsection) toward an individual or group exceeds its threshold ($\theta_{fearThreshold}$), it will express it in the chat. SChMUSR keeps track of when others express fear and uses this to influence its own fear. Let $F^{other}_{j,i}(t) = 1$ when player j has stated that they fear player i in round t, and 0 otherwise. Then,

$$F_i^{\text{contagion}}(t) = \sum_j F^{other}_{j,i}(t) \tag{4}$$

Threat Response. Once SChMUSR has decided to attack a threat, it will attempt to coordinate the attack with others. First, SChMUSR calculates the amount of strength needed for the attack to be effective. Then, it recruits others to try to achieve that amount of collective strength by issuing a *token allocation* message (e.g., "I will attack ¡player¿ with ¡x¿ tokens").

2.3 Additional Modifications

Beyond the fear and chat mechanisms, our implementation of SChMUSR includes two additional changes to the CAB framework from **(author?)** [5]. First, we eliminate the $\theta_{minKeep}$ parameter, which previously controlled the minimum proportion of tokens an agent keeps each round. We found its behavior was better captured dynamically by other parameters, and its inclusion limited broader parameter exploration. Accordingly, we fix $\theta_{minKeep} = 0$.

Second, whereas **(author?)** [5] used separate parameter sets for poor, middle-class, and rich agents, we use a single unified parameter set. We observed no loss in performance from this simplification.

3 Code and Additional Data

Additional data, along with all code used to run simulations, define agent algorithms, process and analyze the data is included in the following repository: https://github.com/MELICI0US/SChMUSR-SM.git

References

[1] A. Baier. Trust and antitrust. *Ethics*, 96(2):231–260, 1986.

[2] Russell Hardin. *Trust and trustworthiness*. Russell Sage Foundation, April 2004.

[3] Thomas M. Jones. Ethical decision making by individuals in organizations: An issue-contingent model. *Academy of management review*, 16(2):366, 1991.

[4] Roger C. Mayer, James H. Davis, and F. David Schoorman. An integrative model of organizational trust. *Academy of management review*, 20(3):709, 1995.

[5] Jonathan Skaggs, Michael Richards, Melissa Morris, Michael A. Goodrich, and Jacob W. Crandall. Fostering collective action in complex societies using community-based agents. *International Joint Conference on Artificial Intelligence*, page 211–219, 2024.