RLupus: Cooperation through Emergent Communication in *The Werewolf* Social Deduction Game

Nicolo' Brandizzi
Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza University of Rome
Via Ariosto, 25, 00185 Roma, Italy.
brandizzi@diag.uniroma1.it

Luca Iocchi
Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza University of Rome
Via Ariosto, 25, 00185 Roma, Italy.
iocchi@diag.uniroma1.it

Davide Grossi
Bernoulli Institute for Maths, CS and AI, University of Groningen
Groningen, The Netherlands.
ACLE, ILLC, University of Amsterdam
Amsterdam, The Netherlands.
d.grossi@rug.nl

ABSTRACT

Multi-agent systems have been studied intensively for their ability to develop complex behaviors from a simple set of rules. One such behavior is cooperation achievable through communication which can be either hand-crafted or emergent. The environmental setting is crucial to determine what kind of cooperation is needed and how communication should be exchanged by working agents. Social games are a good choice to study the emergence of sophisticated communication patterns for their ability to cherry-pick certain aspects of interaction which are then easier to contextualize.

This paper focuses on social deduction games (SDG) and the emergence of communication in them. We study a specific SDG, known as *The Werewolf*, and study if and how various forms of communication influence the outcome of the game. Experimental results show that the introduction of a communication signal greatly increases the winning chances of a class of players. We also study the effect of the signal's length and range on the overall performance showing a non-linear relationship between the two.

KEYWORDS

Multi-agent systems, Social deduction games, Deep reinforcement learning, Emergent communication

ACM Reference Format:
Nicolo' Brandizzi, Luca Iocchi, and Davide Grossi. 2021. RLupus: Cooperation through Emergent Communication in *The Werewolf* Social Deduction Game. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), London, UK, May 3–7, 2021*, IFAAMAS, 8 pages.

1 INTRODUCTION

Social deduction games (SGDs) are games characterized by the confrontation between two or more parties of which one is usually seen as an evil faction. The other parties must deduce the real intention of the former, seeing through lies and deceptions. While the details of these games may change, free communication[1] is a common aspect for them all.

In artificial settings communication would lead to increased complexity in the environment both on the user side, where engineers are tasked to design an expressive and robust syntax [13, 15, 32], and on the artificial players which have to learn the syntax and the meaning of the available words. For shallow players this task quickly becomes unfeasible. A common solution is to define a game-specific language containing communication semantics. This language is devised by usually injecting some kind of expert knowledge into the system. The language becomes then part of the game, providing for new available communication actions augmenting the agents' choices in the decision-making process. Instead of developing a language for the agents, in this article, we propose another approach enabling players to use a free communication mechanism.

Context of the paper. Games arising from social interaction have been extensively studied within the multi-agent reinforcement learning (MARL) literature. Indeed, MARL systems have been successfully used to model a wide variety of social systems found in nature and in modern society, from food-collective ants systems to complex sharing of information in social networks [6]. In these settings, the agents' behavior can be cooperative, competitive or a mix of the two. In [6] a study of these different kinds of settings and the algorithms associated with them is pursued while [26, 39] focuses on the difference between competitive and cooperative agents.

Although MARL has been used for a wide variety of applications, the reinforcement learning paradigm may not scale up easily in complex multi-agent systems. For this reason, RL has been integrated with deep neural networks (DeepRL, DRL) [33]. This allows RL to scale to problems that were previously intractable, such as playing video games using pixels as input [18]. Shortly after, this approach has been applied to multi-agent systems to study the complex emergent behavior of multiple agents interacting with each other to reach a goal. Although DRL is a well-established field of study, agent interaction via actions or communication remains a challenging problem.

This research area is strictly tied to the study of complex behavior arising from the interaction of simple agents; these aspects are mainly studied through the usage of structured game systems. In their work, Baker et al. [2] showed how a few simple rules from the *Hide'n Seek* game can generate complicated behaviors to the point of exploiting environmental errors to their advantages. Along the same line, Leibo et al. [24] carried out extensive analysis on the problem of autocurricula and non-stationary learning in multi-agent deep reinforcement learning (MADRL); they pointed out how the interaction of competitive agents can culminate in an endless cycle of counter-strategies due to the non-stationarity of the environment.

[1] Players can communicate with each other with no limitations.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, London, UK.

In particular, one such complex behavior consists in the emergence of communication between cooperative agents. Recent works investigate this aspect in various environments varying from joined image captioning [16], to negotiation [7] and simulated pointing games [30]. Our paper is a contribution to this line of research, focusing on communication in social deduction games.

Paper contribution and outline. The paper makes two main contributions:

- a formal description of social deduction games coupled with a general reinforcement learning solution framework, which allows for free communication among agents without requiring to provide game-specific knowledge;
- an analysis of the performance obtained through learned communication behaviors in an instance of the above class of games: *The Werewolf*[2] social deduction game.

The paper is organized as follows. An overview of the related work is provided in Section 2. In Section 3, the problem statement is formalized together with the general RL framework. Section 5 describes the *Werewolf* game instance in detail, defining both the game logic and the actual implementation. This game provides a fit ground to study language emergence since its whole system is based on communication, indeed it is the subject of an annual AiWolf context in Japan. The experimental settings and results are reported and commented in Section 6 together with the comparison between our work and [20]. Finally, a discussion is provided in Section 7.

The code is available at https://github.com/nicofirst1/rl_werewolf.[3]

2 RELATED WORK

In our work we rely on the findings coming from the social interaction field of psychology, coupled with the multi-agent systems and reinforcement learning.

Social Deduction Games. Social deduction games have been studied in the broader context of social interactions [12]. In particular, they have been used to study the role of rationality in inter-personal interaction [9], analyze the different forms of social mechanics [10], and research the role of communital topology [1], however, the deduction part of these games has been neglected.

Indeed, in their work [8], the authors give a mathematical formulation for a general social game in order to simplify the way to design such games, however, no specific formulation is given for deduction games.

On the other hand, [46] study the most influential information source in social deduction games and concludes that the interaction that occurred prior to the game are the one regarded as most important to the player. Although this approach is reasonable in the context of acquaintances, no result is given for games in which the playing parties do not know each other.

In all these works, the goal is centered around the social interaction between players. In our work, we shift the focus to the finding of an optimal policy to improve the performances of a party.

Multi-agent Deep RL. Applying RL paradigms to find an optimal policy for multi-agent systems is a well established line of work [5, 35, 39]. In recent years deep neural networks (DNN) have been used to solve the issue of complex tasks such as playing Atari games [18], cooperating in Hide-and-seek [2] and competing with humans on strategic games [36, 42].

A common paradigm for taring RL agents in a deep setting is policy gradient methods [37, 38] where the gradient of a parametrized policy is used to guide the agent in the direction of maximal expected reward. In our work we leverage a particular instance of these methods called Proximal Policy Optimization [34].

In all such cases, the complexity of the environment coupled with the presence of a DNN generated unexpected behaviors that are usually counter-intuitive for humans but achieve greater performances in the task at hand. In our work, we leverage this aspect and further study how the coordination between agents varies under different communication instances.

Emergent Communication. This phenomenon has been exploited in the newly born field of emergent communication [44], where agents are given the choice to use a communication channel in order to achieve a common goal.

The workshop on Emergent Communication (Emecom) includes many publications in the field of natural language processing [25] strictly tied to MADRL [14, 23, 26] and social deduction games as an environment. Standard games for this line of research are the Task & Talk [22], which is centered around dialogue, on the other hand, the Pointing Game [30] grounds the communication into natural image processing.

Another instance of these settings is *The Werewolf* game where the players find themselves split into two opposite groups in a partially known environment. This game has gained increased popularity, in the field of cooperation through emergent communication, especially in Japan where the annual AiWolf contest [4, 17, 21, 31] sees artificial agents competing with and against human players to win the game with fixed language syntax. In particular, [45] set up a 5-player game with additional roles and use a Deep Q-Network to determine who to trust or to kill.

In our work, we choose *The Werewolf* as an instance of the general SDG framework. However, our implementation differs from the ones in the *AiWolf contest*, the closest being [20], where the authors use Q-learning to study the winning chances of the villagers in a game with 16 players, divided into 14 villagers and 2 werewolves. Indeed we drop the hand-coded syntax and let the players develop their own communication by defining some general attributes of the channel.

3 PROBLEM STATEMENT

Social deduction games (SDG) are characterized by the presence of a number of opposing parties

$$P^{(1)}, P^{(2)}, \dots, P^{(M)}$$

(typically $M = 2$), each containing a defined number of players

$$P^{(m)} = \{p_1^{(m)}, p_2^{(m)}, \dots, p_n^{(m)}\}$$

[2] Also known as *Lupus in Fabula*.
[3] Supplementary material can be consulted via this link.

The game evolves as a sequence of actions performed by the players of the parties, typically in turns. The effect of these actions contributes to the definition of the game score.

The goal of each party is to maximize some score, by performing suitable social behaviors, including for example leveraging other players by means of bluffs and lies. These deceitful methods are the base for any SDG and force every player to perform a deductive analysis on the member of the other parties.

During the execution of the game, agents can communicate among them, either implicitly or explicitly. Decisions about how, when, and what to communicate are critical choices for the success of the game.

We thus distinguish two categories of actions performed by the players: 1) *game actions*, that are actions that affect the evolution of the game, 2) *communication actions*, that are actions affecting only the mental state (i.e., the knowledge state) of the players.

In this formalization, we consider only forms of explicit communication, while studying forms of implicit communication is left as future work.

With the previous assumptions, in this paper we consider the formalization of a SGD with the following elements:

- an environment S implementing the game logic;
- a discrete set of possible game actions

$$\{a_1, a_2, \ldots, a_n\} = A$$

- unidirectional communication actions $C_{i,j}(b)$ intended to convey some information b between two players:

$$C_{i,j}(b) : p_j \rightarrow p_i \quad \forall p_j, p_i \quad j \neq i$$

Notice that, while it is relatively straightforward to formalize the specifications of game actions, for example in terms of preconditions and post-conditions using action representation formalisms, it is less clear how to formalize communication actions, since it would require an explicit model of agents' knowledge. For modeling this kind of communication actions, the use of typical action formalisms is not straightforward. For example, they may need to be extended with epistemic operators.

4 GENERAL SOLUTION FRAMEWORK

In this article, we study social deduction games that can be formalized as a multi-agent (deep) reinforcement learning (RL) scenario. In such scenarios, each party has to choose an optimal strategy or policy (i.e., an optimal assignment from states to actions), in order to maximize the game score. As already mentioned, a particular feature of SDGs is the presence of communication actions and the need to choose optimal communications among the players within a party.

The problem of learning optimal policies in multi-agent games is indeed well known and many solutions are available. However, when communication actions are involved, the use of artificial intelligence techniques to make optimal decisions about how what and when to communicate is still a challenging problem under investigation and fewer research works are available.

The advantage in defining a solution based on RL is that it does not require an explicit model of the transition function for communication actions, so in other words, optimal behavior can be computed without the need of associating a semantic meaning to the communication actions. While this feature can be considered not desirable for some kinds of applications (e.g., for mixed human-AI teams), it is very convenient for AI teams based on RL that can learn their own communication language in order to win the game. Indeed AI agents can effectively learn a communication language without the need of making the semantics of communication explicit. Explainability of learned communication actions is left as future work.

Embedding communication in action. When the communication $C^{(t)}(b)$[4] at time-step t can be considered a description of the action to take a_{t+1}, then the channel can be embedded into the action space and regarded as a descriptive action.
In such cases one policy :

$$\pi(\{a_t, C^{(t)}(b)\}) \rightarrow \{a_{t+1}, C^{(t+1)}(b)\}$$

is needed to interact with other agents.

Communication turns. On the other hand, when the communication needs to convey information disjointed from the action a turn-based communication strategy must be implemented.

In these cases the agents will have two codependent policies: one to interpret the communication coming from other players:

$$\pi_C(C^{(t)}(b)) \rightarrow C^{(t+1)}(b)$$

and one for the actions $\pi_a(a_t) \rightarrow a_{t+1}$.

5 R-LUPUS FRAMEWORK

In this section, we present the *RLupus* framework for the Werewolf social deduction game in which we apply the above formalization. In this specific context, our aim is to study if and how various forms of communication can influence the outcome of the game, in which only one party is able to learn while the other one has a fixed, hand-coded policy.

The hypothesis is that, under the same circumstances, the agents which are able to communicate will perform much better than the ones not allowed to exchange information. Moreover, we speculate that different communication settings will have diverse influences on the amount of coordination among the agents as well as on the final outcome of the game.

In the following sub-sections, we give a brief introduction on *The Werewolf* game logic, a description of the RL environment with all its components, and the policies used for the players in the game.

5.1 The Werewolf Game

Werewolf is a social deduction game modeling conflicts between two groups in a partially known environment. In its easiest version, the game sees two groups ($M = 2$), villagers $P^{(v)}$ and werewolves $P^{(w)}$ where $P^{(v)} > P^{(w)} + 1$. The wolves know exactly the identity of each player, while the villager are certain exclusively about their role and the number of werewolves. In an open setup, an additional moderator is needed to coordinate the players.
The game is divided into two phases: night and day, interleaving each other.

[4] The dependency on i, j has been dropped for simplicity.

The game ends either when the villagers execute the last werewolf or there is an even number of both roles. The latter case implies the wolves winning since the execution phase can be stalled, thus taking away the only possibility for villagers to kill the wolves.

Table 1: RLupus: Multi-channel metrics. The first column reports the type of **comm**(unication)channel regarding the **S**ignal**L**ength and the **S**ignal**R**ange. The next four show the metrics values for villagers winning rate, suicide rare, number of days elapsed and accordance rate

Comm	Win Vil	Suicide	Days	Accord
0SL	0.044	0.086	1.55	0.47
1SL-2SR	0.19	0.078	1.58	0.47
1SL-9SR	0.21	0.078	1.58	0.47
9SL-2SR	**0.45**	**0.067**	**1.9**	**0.47**
9SL-9SR	0.19	0.077	1.58	0.46

5.2 RLupus Environment

Dealing with RL implies the presence of an action and state set; the latter are referred to as action space and observation space which are presented in the following section [5]

Action Space. As mentioned in Section 3, the Werewolf can be implemented with one policy $\pi(\{a_t, C^{(t)}\}) \rightarrow \{a_{t+1}, C^{(t+1)}\}$, since the communication can be seen as an extension of the action.

Indeed the action space is divided into two parts:

- *Target*: The target a_t is a discrete value in range $a_t \in [0, N-1]$, where N is the number of players. Its intended usage is to allow players to vote for other players during the game. The range of possible values never changes during the execution, instead, illegal actions, such as voting for dead players, are filtered out later in the model.
- *Signal*: The signal $C^{(t)}$ has length $SL \in (0, \infty)$ and range $SR \in (2, N)$, both are used in order to define the valid space for communication before the training.

Observation Space. The observation space characterizes what the agents perceive in the environment. This space includes both the actions space and other information about the environment[6].

Rewards. The rewards, or penalties, are the core of the environment and determine how the players interact, learn and develop new strategies; the main goal of an agent is to take actions that will maximize the expected reward.
Following our formalization, the environment is responsible for delivering a reward to each player [7].

Metrics. In order to measure the changes in the agent behavior the following normalized metrics are logged:

- *Suicide*: the number of times an agent votes for itself during an execution phase.
- *Wins*: the villagers' wins are plotted in the normalized range of values.
- *Average days* : average number of days before a match ends.
- *Accord*: This value represent, on average, the percentage of agents that vote for the same target during the two execution phases.

5.3 Policies

An agent's policy defines the behavior of a player during the game. In this environment, there are two kinds of policies: *trainable* policies use custom algorithms to collect experience and learn to maximize the reward; *static* policies are hard-coded behaviors that are used in order to guarantee a fixed baseline trough out the evaluation.

In this work, we assign static policies to opponent players (werewolves) and training policies to AI agents (villagers) that are learning how to win the game.

Static policies for werewolves. Static policies are reserved for the werewolf agents; their aim is to allow a baseline evaluation of the villager learning. Since wolves are more likely to win in a completely random environment, the application of such policies is enough to prove the development of new strategies for the villagers if the winning rates are to change significantly.
Three policy are implemented:

- *Random Target*: simply chooses a random non-dead player among the villagers during the execution phase.
- *Random Target Unite*: this policy targets the same player for every wolf, both during day and night execution; this allows the werewolf to dominate the day execution phase with random villagers.
- *Revenge Target*: with this policy, the wolves will either vote randomly or target a villager who previously voted for a wolf.

Trainable policies for villagers. The trainable policies are obtained through the RL framework described in the previous section and the Proximal Policy Optimization (PPO) [34] algorithm with a simple fully connected network and a LSTM cell. PPO uses a surrogate loss function to keep the difference between the old and the new policy within a safe range[8].

Learning Werewolves. Finally, we must address the possibility of an environment where both villagers and werewolves are able to learn. Having multiple agents learning at the same time void the stationary assumption which is necessary for optimality in RL. This assumption is already invalidated by the presence of multiple villagers but the coordination between them alleviates the problem. Introducing an adversarial set of learning agents would cause an increased complexity that would cloud the goal of this paper which is to study the emergence of a language between agents.

6 RESULTS

Following, an analysis of the results for both nine (Section 6.1) and twenty-one (Section 6.2) players is given. In both cases, a baseline setting with no communication is compared with multi-channel

[5]For the sake of conciseness some formulations are omitted from the paper. The interested reader can refer to the Appendix.
[6]The complete observation space is described in Appendix 8.1.1
[7]The complete set of rewards is described in Appendix 8.1.2
[8]More information on the policy loss are available in Appendix 8.1.3

Table 2: RLupus: single/no channel metrics. The Design Choice part of the table shows which kind of policy Random, Unite or Revenge has been used in relation to the communication, while the Results half present the metric' values

Design Choice				Results			
Comm	**Rnd**	**Unt**	**Rvg**	**Vil Win**	**Accord**	**Suicide**	**Days**
0SL	X			0.044	0.478	0.086	1.55
0SL		X		0.03	**0.695**	**0.059**	**1.5**
0SL			X	**0.12**	0.482	0.078	1.64
1SL-2SR	X			0.19	0.47	0.078	1.58
1SL-2SR		X		0.08	**0.685**	**0.055**	**1.52**
1SL-2SR			X	**0.4**	0.479	0.065	1.9

communication to show the increased performance reported using the metrics in Section 5.2.

Moreover, for the nine players instance, results for the additional *revenge* and *unite* policies are reported against the *random* one.

Finally, in Section 6.3, a summary for a setting with 16 players is given for both the *AiWolf* environment [20] and the *RLupus* one.

6.1 Nine Players

A game with nine players is relatively short, but not trivial for the villagers. Indeed, in a completely random environment, they have a probability of 3.12% to win[9].

Since the random policy is believed to hold the least expert knowledge about the game environment, we present the results for this policy only in the next section. On the other hand, Section 6.1.2 also shows the results for the *revenge* and *unite* policies.

6.1.1 Multi-channel communication. Table 1 presents the results (columns) obtained with different forms of communication (rows), with different communication settings (SL=signal length, SR = signal range). From the table, it is clear that any form of communication improves over the non-communication setting (0SL). Moreover, it shows how the bit communication (2SR) is performing much better than the full ranged one (9SR). Indeed, with the addition of just one bit of communication, the villagers are able to perform as well as the full extended communication (9SL-9SR). In fact, for the settings where $SR = 9$, increasing the channel length SL has the only effect of speeding up the convergence by a factor of circa 25% for each increase.

On the other hand, the bit communication, independently from the channel length, provides generally better results. This leads to the conclusion that the two parameters dictating the change in the communication channel are not equally influential when it comes to the agent's learning.

On a final note, the average time to train the agents was 6 hours on a single GPU[10] machine.

6.1.2 Policies comparison. Table 2 reports results related also to the werewolves policies (Rnd = Random, Unt = Unite, Rvg = Revenge).

When no communication is available (0SL), the accord value is almost identical for both the revenge and the random policy and much higher for the unite one, as previously anticipated.

The number of days reports a similar trend being smaller in the unite settings where the game ends sooner. Finally, the revenge winning rate reaches a much greater value than the other policies, 12%. The motivation being the simplicity of the policy itself, i.e. the wolves can not hide behind purely random action anymore and are not strong enough to drive the majority of the votes toward a villager.

Following the considerations for the three policies and bit communication ($SL = 1$ $SR = 2$) referred to Table 2:

- *Random*: the villager's winning rate reaches 20% ; 4.5 times more than the previous condition and 6.5 times the theoretical winning rate.
- *Unite*: as in the previous case, the coordination is much greater. Indeed the villagers are able to increase their winning rate by a factor of ×3. This result alone proves how even in the most disadvantageous setting the introduction of a limited communication channel can greatly favor the final outcome.
- *Revenge*: again, the revenge policy is the easiest to spot for a trained agent reaching a winning rate of 40%.

Table 3: RLupus: 21 Player metrics.

Comm	Win	Suicide	Days	Accord
0SL	0.42	0.072	7.84	0.56
1SL-2SR	0.25	0.075	7.74	**0.57**
9SL-2SR	**0.98**	**0.04**	**7.3**	0.56
21SL-2SR	0.94	0.05	7.6	0.56
1SL-21SR	0.72	0.062	8	0.56
21SL-21SR	0.61	0.066	7.9	0.56

6.2 Twenty one players

Different to what studied in the previous section, the twenty-one players (21P) environment has more room for the villagers to win in a completely random setting. Indeed, by building a tabular representation of the expanded tree (shown in Table 4), the reader can see how the total villagers' winning possibilities increased to 11.62% in a completely random environment.

6.2.1 Multi channel communication. A comparison among all the multi-channel settings is reported in Table 3.

[9] See Appendix 8.2 for the complete estimation.
[10] Nvidia Geforce GTX 1080 Ti.

In accordance with the findings from the previous section, the bit communication ($9SL - 2SR$) seems to perform better than the full-ranged one ($*SL - 21SR$) both in terms of winning ratio and suicides.

On the other hand, the $1SL - 2SR$ instance performs worse than the no communication one, this anomaly can be attributed to an incorrect exploration of the environment. As mentioned in Table 4, the expanded tree size is greater than the instance with nine players, thus the algorithm can get more easily stuck in a local minima. However, every other setting with a communication channel has a higher winning rate than the one without, thus one could confidently say that the communication signal is indeed helping the agents exploring the possible branches better than in other cases. For these reasons, we believe this outcome shows how the shape of the communication signal can improve the agents' capacity to explore the environment and thus should be further investigated in future work.

6.3 Sixteen Players

As previously mentioned, Kajiwara et al. [20] presented the AiWolf framework, implementing a well-defined semantic [11] constrained to be both easy to use for the artificial players and understandable by humans. They extended the agents' adaptive capabilities by adopting a Deep Q-network architecture and reported an increase in the villagers' winning rate when the latter are the only ones capable of learning.

Similarly, we trained an environment with 16 players, (2 werewolves and 14 villagers) and reported our results in Table 5, together with the results of the AiWolf framework.

The results are not directly comparable because the two frameworks represent the game in different ways. However, it is interesting to observe the increase in performance due to the learning process. Indeed, the table shows baseline results without learning in the two frameworks and results obtained with RL.

In comparison with AiWolf, it can be seen how the 0SL setting in RLupus already improves the winning chances by 33% reaching 90.2% and adding a single bit communication results in almost perfect victory on the villagers' side (98.4%).

[11] More details in Appendix 8.3

Table 4: RLupus: Mapping between outcomes (Villager-Werewolves) and probabilities.

Outcome	Prob. %	Leaves	Total %
0-12	0.029	1	11.62
0-10	0.143	4	
0-8	0.447	10	
0-6	1.162	20	
0-4	2.819	35	
0-2	7.02	56	
1-1	21.06	56	88.38
2-2	27.965	21	
3-3	26.017	6	
4-4	26.017	1	
Total	1.0	210	1.0

Table 5: Winning rate for [20] and RLupus in both baseline and active learning.

Design		Win Vil	
		AiWolf	RLupus
No Learning	AiWolf baseline	38.6	/
	RLupus baseline	/	56.9
Learning	DQN / hand-coded	52.9	/
	PPO / 0SL	/	90.2
	PPO / 1SL-2SR	/	98.4

7 CONCLUSIONS AND FUTURE WORK

In this paper, we defined a formalism for social deduction games, in which communication is an essential part of the game together with the allowed actions. On top of that, a general resolution framework based on reinforcement learning was defined and applied to the *The Werewolf* SDG by studying how various forms of communication influenced the outcomes of a match.

As shown by the experimental results, the introduction of different forms of communication greatly increases the performance of the agents (villagers). In particular, we observed that a Boolean signal range is preferred to an integer one. The reason for this is unclear. We speculate it may lay in the duality of the roles of the game.

Moreover, we found how much of the villagers' winning rate is determined by the agents not voting for themselves while keeping the accord value to a maximum. This translates into better coordination, which is possible only when there is a sufficient amount of communication present in the environment.

Also, we noticed that there is no linear map between the amount of communication permitted, i.e. *SL* and *SR*, and the overall performance. Indeed there seems to be an optimal combination of the two which depends mainly on the signal length.

We conclude by identifying three directions in which our work could be extended.

Model of SDGs. One possible line of research consists of studying an instance of an SDG where the communication cannot be intrinsically tied to the action space, i.e. multi-step communication game such as Task & Talk [22]. Alternatively, one could decide to depart from the deep part of the RL resolution framework and choose an SDG instance whose environment can be optically solved with standard reinforcement methods, e.g. Pointing game [30].

The Werewolf. On the other hand, various possibilities arise from the study of The Werewolf game. One such could be the analysis of the language used for communication to highlight potential patterns; given that performance alone should not be considered a valid metric for the study of emergent communication [27]. Moreover, one could extend the RLupus environment either by adding other roles, for example, the medium and the witch, or using normalized continuous vectors for the communication channel which allow

for the backpropagation of errors directly through the communication channel [14] [12]. Finally, a deeper analysis could be applied to understand the impact of the communication parameters (SR and SL) on the metrics of the system and the overall performance.

Human-machine coordination. Coordination among artificial agents is a key engineering challenge: from task allocation [11, 43], knowledge management [47], distributed constraint optimization problems [29] to multi-robot SLAM [40, 48] and language models [3, 41]. And even more challenging is coordination between artificial and human agents. The study of emergent communication in SDGs could provide useful novel insights for the development of more efficient coordination among artificial agents, as well as between artificial and human agents.

ACKNOWLEDGMENTS

We would like to thank the reviewers of ALA'2021 for several useful comments that helped us improve our manuscript. Davide Grossi wishes to acknowledge partial support for this research by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

REFERENCES

[1] Guillermo Abramson and Marcelo Kuperman. 2001. Social games in a social network. *Physical Review E* 63, 3 (2001), 030901.

[2] Bowen Baker, Ingmar Kanitscheider, Todor M. Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. 2020. Emergent Tool Use From Multi-Agent Autocurricula. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=SkxpxJBKwS

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research* 3 (2003), 1137–1155.

[4] Xiaoheng Bi and Tetsuro Tanaka. 2016. Human-side strategies in the werewolf game against the stealth werewolf strategy. In *International Conference on Computers and Games*. Springer, 93–102.

[5] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (2008), 156–172.

[6] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. 2010. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*. Springer, 183–221.

[7] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z. Leibo, Karl Tuyls, and Stephen Clark. 2018. Emergent Communication through Negotiation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=Hk6WhagRW

[8] Kam Tong Chan, Irwin King, and Man-Ching Yuen. 2009. Mathematical modeling of social games. In *2009 International Conference on Computational Science and Engineering*, Vol. 4. IEEE, 1205–1210.

[9] Andrew M Colman. 2003. Cooperation, psychological game theory, and limitations of rationality in social interaction. *Behavioral and brain sciences* 26, 2 (2003), 139–153.

[10] Mia Consalvo. 2011. Using your friends: Social mechanics in social games. In *Proceedings of the 6th International Conference on Foundations of Digital Games*. 188–195.

[11] Mathijs M de Weerdt, Yingqian Zhang, and Tomas Klos. 2012. Multiagent task allocation in social networks. *Autonomous Agents and Multi-Agent Systems* 25, 1 (2012), 46–86.

[12] Markus Eger and Chris Martens. 2018. Keeping the story straight: A comparison of commitment strategies for a social deduction game. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[13] Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. 1994. KQML as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*. 456–463.

[14] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*. 2137–2145.

[15] Michael R Genesereth, Richard E Fikes, et al. 1992. Knowledge interchange format-version 3.0: reference manual. (1992).

[16] Laura Graesser, Kyunghyun Cho, and Douwe Kiela. 2019. Emergent Linguistic Phenomena in Multi-Agent Communication Games. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3698–3708. https://doi.org/10.18653/v1/D19-1384

[17] Yuya Hirata, Michimasa Inaba, Kenichi Takahashi, Fujio Toriumi, Hirotaka Osawa, Daisuke Katagami, and Kousuke Shinoda. 2016. Werewolf game modeling using action probabilities based on play log analysis. In *International Conference on Computers and Games*. Springer, 103–114.

[18] Ionel-Alexandru Hosu and Traian Rebedea. 2016. Playing Atari Games with Deep Reinforcement Learning and Human Checkpoint Replay. *CoRR* abs/1607.05077 (2016). arXiv:1607.05077 http://arxiv.org/abs/1607.05077

[19] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rkE3y85ee

[20] Kengo Kajiwara, Fujio Toriumi, Hirotada Ohashi, Hirotaka Osawa, Daisuke Katagami, Michimasa Inaba, Kosuke Shinoda, Junji Nishino, et al. 2014. Extraction of optimal strategies in human wolf using reinforcement learning. *Proceedings of the 76th National Convention* 2014, 1 (2014), 597–598.

[21] Daisuke Katagami, Shono Takaku, Michimasa Inaba, Hirotaka Osawa, Kosuke Shinoda, Junji Nishino, and Fujio Toriumi. 2014. Investigation of the effects of nonverbal information on werewolf. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 982–987.

[22] Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. 2017. Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 2962–2967. https://doi.org/10.18653/v1/d17-1321

[23] Angeliki Lazaridou and Marco Baroni. 2020. Emergent Multi-Agent Communication in the Deep Learning Era. *CoRR* abs/2006.02419 (2020). arXiv:2006.02419 https://arxiv.org/abs/2006.02419

[24] Joel Z. Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. 2019. Autocurricula and the Emergence of Innovation from Social Interaction: A Manifesto for Multi-Agent Intelligence Research. *CoRR* abs/1903.00742 (2019). arXiv:1903.00742 http://arxiv.org/abs/1903.00742

[25] Yaoyiran Li, Edoardo Maria Ponti, Ivan Vulic, and Anna Korhonen. 2020. Emergent Communication Pretraining for Few-Shot Machine Translation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, Donia Scott, Núria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, 4716–4731. https://doi.org/10.18653/v1/2020.coling-main.416

[26] Paul Pu Liang, Jeffrey Chen, Ruslan Salakhutdinov, Louis-Philippe Morency, and Satwik Kottur. 2020. On Emergent Communication in Competitive Multi-Agent Teams. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, Amal El Fallah Seghrouchni, Gita Sukthankar, Bo An, and Neil Yorke-Smith (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, 735–743. https://dl.acm.org/doi/abs/10.5555/3398761.3398849

[27] Ryan Lowe, Jakob N. Foerster, Y-Lan Boureau, Joelle Pineau, and Yann N. Dauphin. 2019. On the Pitfalls of Measuring Emergent Communication. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, 693–701. http://dl.acm.org/citation.cfm?id=3331757

[28] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=S1jE5L5gl

[29] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. 2005. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence* 161, 1-2 (2005), 149–180.

[30] Igor Mordatch and Pieter Abbeel. 2018. Emergence of grounded compositional language in multi-agent populations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[12] On the other hand, [19] and [28] use a Gumbel approximation to backpropagate the error through a discrete distribution.

[31] Noritsugu Nakamura, Michimasa Inaba, Kenichi Takahashi, Fujio Toriumi, Hirotaka Osawa, Daisuke Katagami, and Kousuke Shinoda. 2016. Constructing a human-like agent for the werewolf game using a psychological model based multiple perspectives. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1–8.

[32] Patrick D O'Brien and Richard C Nicol. 1998. FIPA—towards a standard for software agents. *BT Technology Journal* 16, 3 (1998), 51–59.

[33] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347 http://arxiv.org/abs/1707.06347

[35] Yoav Shoham, Rob Powers, and Trond Grenager. 2003. Multi-agent reinforcement learning: a critical survey. *Web manuscript* 2 (2003).

[36] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529, 7587 (2016), 484–489.

[37] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms.

[38] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems* 12 (1999), 1057–1063.

[39] Ming Tan. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*. 330–337.

[40] Sebastian Thrun and Yufeng Liu. 2005. Multi-robot SLAM with sparse extended information filers. In *Robotics Research. The Eleventh International Symposium*. Springer, 254–266.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[42] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[43] Perukrishnen Vytelingum, Thomas D Voice, Sarvapali D Ramchurn, Alex Rogers, and Nicholas R Jennings. 2010. Agent-based micro-storage management for the smart grid. (2010).

[44] Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. 2003. Progress in the simulation of emergent communication and language. *Adaptive Behavior* 11, 1 (2003), 37–69.

[45] T. Wang and T. Kaneko. 2018. Application of Deep Reinforcement Learning in Werewolf Game Agents. In *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*. 28–33.

[46] Sarah Wiseman and Kevin Lewis. 2019. What Data do Players Rely on in Social Deduction Games?. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*. 781–787.

[47] Dong-Jun Wu. 2001. Software agents for knowledge management: coordination in multi-agent supply chains and auctions. *Expert Systems with Applications* 20, 1 (2001), 51–64.

[48] Xun S Zhou and Stergios I Roumeliotis. 2006. Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case. In *2006 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 1785–1792.