Promoting Resilience in Multi-Agent Reinforcement Learning via Confusion-Based Communication

Ofir Abu
Hebrew University of Jerusalem
ofir.abu@mail.huji.ac.il

Matthias Gerstgrasser
School of Engineering And Applied Sciences
Harvard University
matthias@g.harvard.edu

Jeffrey S. Rosenschein
Hebrew University of Jerusalem
jeff@cs.huji.ac.il

Sarah Keren
Taub Faculty of Computer Science
Technion - Israel Institute of Technology
sarahk@cs.technion.ac.il

Abstract

Recent advances in *multi-agent reinforcement learning (MARL)* provide a variety of tools that support the ability of agents to adapt to unexpected changes in their environment, and to operate successfully given their environment's dynamic nature (which may be intensified by the presence of other agents). In this work, we highlight the relationship between a group's ability to collaborate effectively and the group's *resilience*, which we measure as the group's ability to adapt to perturbations in the environment. To promote resilience, we suggest facilitating collaboration via a novel *confusion-based* communication protocol according to which agents broadcast observations that are misaligned with their previous experiences. We allow decisions regarding the width and frequency of messages to be learned autonomously by agents, which are incentivized to reduce confusion. We present empirical evaluation of our approach in a variety of MARL settings.

1 Introduction

Reinforcement Learning (RL) agents are typically required to operate in dynamic environments, and must develop an ability to quickly adapt to unexpected perturbations in those environments. Promoting this ability is challenging, even in single-agent settings [13]. For a group of agents this becomes even more of a challenge; in addition to the dynamic nature of the environment, the agents need to deal with high variance caused by changes in the behavior of other agents [16, 22, 9].

Our objective in this work is to highlight the relationship between a group's ability to collaborate effectively and the group's *resilience*, which we measure as the group's ability to adapt to perturbations in the environment. Contrary to investigations of *transfer learning* [24, 8] or *curriculum learning* [15], we do not have a stationary target domain in which the group of agents is going to be deployed, nor do we have a training phase dedicated to preparing agents for the deployment environment. Instead, we aim to equip a group with the ability to adapt to unexpected changes that can occur at random times.

Recent literature is rich with a variety of different definitions of resilience and robustness, for both single and multi-agent settings [23, 21, 14]. That research usually focuses on resilience in the presence of deliberate adversary attacks upon one or many agents in the system [18]. Instead of considering adversarial settings, we measure group resilience as the agents' performance in the presence of unexpected perturbations.

Preprint. Under review.

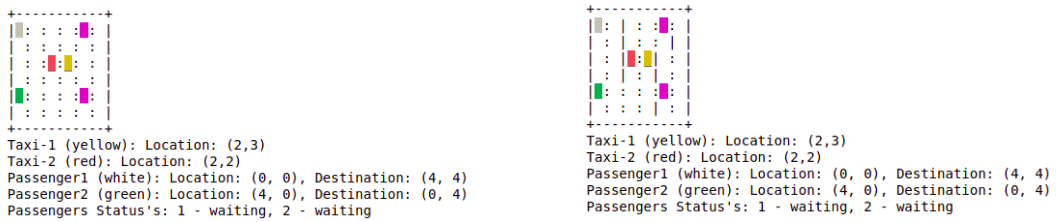


Figure 1: An illustration of a multi-agent taxi domain. Perturbations in the bottom image are depicted as solid lines that represent non-traversable walls.

To support collaboration and demonstrate its effect on group resilience, we introduce a communication protocol that defines the information that each agent shares with the group. Recent work demonstrates how communication in multi-agent reinforcement learning (MARL) settings allows a group to learn and operate efficiently in complex but stationary environments [6, 4]. To promote a group's ability to adapt to a randomly perturbed environment, we present a novel *confusion-based* communication protocol, that requires an agent to broadcast its current observations that are least aligned with its current model of the environment. We show that this effect can also happen through emergent communication, by incentivizing agents to reduce confusion.

Example 1 *Consider Figure 1, which depicts a multi-agent variation of the Taxi domain [2]. In this setting, taxis are associated with one operator, but each taxi receives a direct payment from each passenger when it drops her off at her destination. In addition to the taxis, we posit a* designer*, representing the taxis' operator, who aims to maximize the group's total revenue.*

The designer monitors the taxis' performance (revenue), and notices that taxis sometimes deviate from their usual routes. This may happen, for example, due to road construction, road congestion, or other reasons. In the attempt to maximize the group's performance despite these changes, and with the intention of not overwhelming the communication channel, the designer instructs taxis to broadcast 'confusing' experiences to the others, corresponding to actions that have unexpected outcomes. For example, when a taxi encounters a blocked road, and fails to drive through a street it has driven through regularly, it will broadcast this unsuccessful attempt to the other taxis. Similarly, the taxi will broadcast information about a vacant street that is typically busy. This may relieve the effect that perturbations have on the other taxis and help improve performance.

Our key contributions in this work are threefold. First, we suggest a new measure of group's resilience that corresponds to the group's ability to adapt to unexpected changes. As a second contribution, and in order to promote resilience, we facilitate collaboration within the group. To support collaboration, we offer a new confusion-based communication protocol according to which all agents are notified about notable changes in an agent's surroundings. We consider both the case where a designer instructs the group to follow the protocol as well as the case where communication emerges as a result of incentivizing agents to minimize confusion. Lastly, we offer an empirical evaluation that demonstrates how agents that operate according to the confusion-based protocol are more resilient compared to agents that do not.

2 Measuring Group Resilience

We aim to promote the ability of a group of agents to adapt to random perturbations in their environment. We refer to this ability as *resilience*, and formally define it below. We will then suggest promoting group resilience by facilitating collaboration between the agents.

To measure the resilience of a group of agents we take inspiration from the field of multi-agent robotics. Specifically, the work of [18] that aims to produce a control policy that allows a team of mobile robots to achieve desired performance in the presence of faults and attacks on individual members of the group. According to their work, a group of robots achieves *resilient consensus* if the cooperative robots' performance is in some desired range, even in the presence of up to a bounded number of non-cooperative robots. In essence, we want *resilience* to mean that if an environment undergoes an unexpected (but somehow bounded-in-magnitude) perturbation, then agents can still

achieve a fixed fraction of their original performance. However, we are interested in changes in the environment, rather than the other agents. That is, we differ from the original definition by implicitly taking into consideration any kind of change in the observed environment.

Our definition of resilience is with regard to a distance measure $\delta(M, M^{'})$ that quantifies the magnitude of the change between an original MDP M and the modified MDP $M^{'}$. It also relies on the specification of a utility measure $\mathcal{U}(M)$, quantifying the performance of a group of agents in a given MDP. Given these two measures, we require that a perturbation that results in an environment that is within a bounded distance K from the original environment, will result in a decrease in performance by a factor of at most some constant C_K.[1] We note that a range of subtly different formal definitions can satisfy this intuitive requirement. We defer a detailed discussion on some possible options to the appendix, and provide here only the definitions that are relevant to our experiments. Specifically, our definitions rely on the assumption that a designer might want to guarantee resilience over some subset $\mathcal{M}$ of perturbed environments within the specified distance. For example, a taxi station's manager might be interested in guaranteeing that a group of taxis is resilient under random road blockages, instead of being resilient to arbitrary perturbations that may occur.

Definition 1 (Relative to Origin C_K-resilience) *Given a class of MDPs $\mathcal{M}$, a source MDP $M \in \mathcal{M}$, and a bound $K \in \mathbb{R}$, we say that a group of agents is* universally C_K-resilient *in M over $\mathcal{M}$ if*

$$\forall M^{'} \in \mathcal{M} : \delta(M, M^{'}) \leq K \implies$$
$$\mathcal{U}(M') \geq C_K \cdot \mathcal{U}(M)$$

Resilience over $\mathcal{M}$ allows us to choose a set $\mathcal{M}$ of environments of interest for which the distance condition is easily verified. However, this condition still requires that the bound on performance holds for *any* $M^{'} \in \mathcal{M}$ (under the distance bound), which may be unreasonably strong and impractical in many cases. Therefore, equipped with a probability distribution (e.g., uniform distribution) Ψ over $\mathcal{M}$, we further define resilience-in-expectation as follows.

Definition 2 (Relative to Origin C_K-resilience in Expectation) *Given an MDP M, a distribution over a class of MDPs Ψ, and a bound $K \in \mathbb{R}$, we say that a group of agents is C_K-*resilient in expectation *in M over Ψ if*

$$\mathbb{E}_{[M' \sim \Psi | \delta(M,M') \leq K]} \mathcal{U}(M') \geq C_K \cdot \mathcal{U}(M)$$

Our definition above, requires the expected performance of a group to fulfill a performance guarantee, where the expectation is over a sampled set of MDPs in $\mathcal{M}$ within K-distance of M. It is a known result that polynomially-many samples from Ψ are sufficient to achieve arbitrarily close approximations of the true expectation, with arbitrarily high probability.[2]

We note that definitions 1 and 2 compare the performance of the agents in the perturbed environment against their performance in the original one, without considering its nominal value. This means that a group that follows a non-efficient policy (e.g., performing a no-op action repeatedly) will have high resilience. Our suggested measure should therefore be considered in concert with the group's measure of utility.

2.1 Perturbations

In this work, we are interested in settings in which we have an initial environment and a set of *perturbations* that can occur. In general, a perturbation $\phi : \mathcal{M} \mapsto \mathcal{M}$ is a function transforming a source MDP into a modified MDP. An *atomic perturbation* is a perturbation that changes only one of the basic elements of the original MDP. In the following, given an MDP $M = \langle S, A, R, P, \gamma \rangle$ and perturbation ϕ, the resulting MDP after applying ϕ is denoted by $M^{\phi} = \langle S^{\phi}, A^{\phi}, R^{\phi}, P^{\phi}, \gamma^{\phi} \rangle$.

[1] Notice that this is similar to the classical ϵ-δ-definition of the continuity of a function.

[2] Assume that the utility function $\mathcal{U}(M')$, considered as a random variable with M' drawn from $[M^{'} \sim \Psi | \delta(M, M^{'}) \leq K]$ as above, is i.i.d. for a random draw of $M^{'}$ and has a finite variance σ^2. Then it follows from Chebychev's inequality that in order to be within ϵ of the true mean with a probability of at least δ, it is sufficient to collect at least $\frac{\sigma^2}{\epsilon^2 \cdot (1-\delta)}$ samples.

Among the variety of perturbations that may occur, we focus here on three types of atomic perturbations. *Transition function perturbations* modify the distribution over next states for a single state-action pair. *Reward function perturbations* modify the reward of a single state-action pair. *Initial state perturbations* change the initial state of the MDP (if it is defined).

Definition 3 (Transition Function Perturbation) *A perturbation ϕ is a* transition function perturbation *if for every MDP $M = \langle S, A, R, P, \gamma \rangle$, M^ϕ is identical to M except that for a single action state pair $s \in S$ and $a \in A$, $\mathbb{P}_s^a[S] \neq \mathbb{P}^\phi{}_s^a[S]$.*

Definition 4 (Reward Function Perturbation) *A perturbation ϕ is a* reward function perturbation *if for every MDP $M = \langle S, A, R, P, \gamma \rangle$, M^ϕ is identical to M except that for a single action state pair $s \in S$ and $a \in A$, $r_s^a \neq r^\phi{}_s^a$.*

Definition 5 (Initial State Perturbation) *A perturbation ϕ is a* transition function perturbation *if for every MDP $M = \langle S, s_0, A, R, P, \gamma \rangle$, M^ϕ is identical to M except that $s_0 \neq s_0^\phi$.*

Example 1 (continued) *In our multi-taxi domain, a road blockage can be modeled as a perturbation comprised of atomic transition function perturbations that reduce to zero the probability of transitioning to a blocked cell from any adjacent cell. If the destination of a passenger changes, this can be represented by two atomic perturbations: one that replaces the reward for a dropoff at the original destination with a negative reward, and one that adds a positive reward for a dropoff at the new destination.*

There are a variety of metrics for measuring the distance between two MDPs [20, 1]. We want the *magnitude of a perturbation* to represent the extent by which a perturbed environment is different from the original one. Intuitively, the bigger the magnitude, the harder it would be for a set of RL agents to adapt.

A straightforward way to measure the distance between two MDPs is to count the minimal number of atomic perturbations that transition the original MDP into the transformed one. Another measure is the one suggested by Song et al. [20], where the distance between two MDPs M and $M^{'}$ is calculated by computing the accumulated distance between every state in M and its corresponding state in M'. This definition holds for a setting where the two MDPs are *homogeneous*, such that there exists a correspondence (mapping) between the states, action spaces, and reward functions of the par of MDPs. Given two homogeneous MDPs M and M', the distance $d(s, s')$ between any two states $s \in S_M$ and $s' \in S_{M'}$ is defined as:

$$d(s, s') = \max_{a \in A} \{|r_s^a - r'^a_{s'}| + cT_k(d)(\mathbb{P}_s^a[S_M], \mathbb{P}'_{s'}{}^a[S_{M'}])\}$$

where $r_s^a, \mathbb{P}_s^a[S_M], r'^a_{s'}, \mathbb{P}'^a_{s'}[S_{M'}]$ are the immediate reward and the transition probabilities for M and M' respectively, $T_k(d)$ is the Kantorovich distance [3] between the two probability distributions, and $c \in [0, 1]$ is some hyper-parameter defining the significance of the distance between the distributions.

In our setting, we focus on perturbations Φ that do not change the state space nor the action space, so $\Phi(M)$ and M are homogeneous according to Song et al. [20], and each state $s \in S_M$ corresponds to the same state in $S_\Phi(M)$. We therefore use the above measure to estimate the distance between an MDP and its perturbed variations in our empirical evaluation.

3 Promoting Group Resilience Via Confusion-Based Communication

Equipped with a measure for group resilience, we now focus on maximizing the resilience of a group of RL agents. Recent work in MARL suggests various approaches for promoting efficient collaboration within a group of agents. Such approaches include introducing a communication protocol [6] or a model of other agents' policies [10, 17]. The focus in these frameworks is on promoting collaboration in order to maximize the group's performance in spite of the dynamic nature of the environment and the existence of other agents. We suggest promoting collaboration as a way to promote resilience. We hypothesize that agents that learn to collaborate will adapt more quickly to changes in their environment.

One way to support collaboration is by allowing agents to communicate, and by formulating communication protocols for which the messages encode information that is valuable to the learning experience

of other agents. Communication protocols have been applied to enhance the performance of a group of RL agents [6, 4]. We offer a new *confusion-based communication* protocol, according to which agents broadcast to other agents observations that are not aligned with their previous experiences in the environment. The motivation for this protocol comes from the Prioritized Experience Replay (PER [19]). In PER, a DQN agent maintains a memory buffer containing its transitions, and instead of sampling the memory uniformly for training, the agent samples the memory with "importance weights", trying to improve the learning process. We generalize this idea to the multi-agent setting by instructing the agents to communicate observations or messages with prioritized importance, that is determined by measuring misalignments between expected and observed rewards.

3.1 Computing Confusion

We measure *confusion* according to the extent by which the immediate reward observed by an agent when performing an action in some state is misaligned with its estimated reward. Formally, we compute the level of confusion using the Q function. Let π_p be the policy of agent p, and let s_j be the next state the environment transitioned to after taking action a_i in s_i. The reward $\hat{r_i}$ of taking a_i in s_i is estimated by: $\hat{r_i} \approx Q_{\pi_p}(s_i, a_i) - Q_{\pi_p}(s_j, \pi_p(s_j))$. The confusion level for a given state and agent is defined as follows:

Definition 6 (Confusion) *Let r_i and $\hat{r_i}$ be the observed and estimated reward of agent p after taking action $a_i = \pi_p(s_i)$ in s_i. The level of confusion of agent p at s_i after taking action a_i, denoted J_{s_i,a_i}, is defined as:*

$$J^p_{s_i,a_i} = \frac{|r_i - \hat{r_i}|}{r_i}$$

3.2 Confusion-Based Communication

Confusion corresponds to the agents' level of familiarity with the environment. Thus, when a perturbation occurs, the confusion of the group should grow. Accordingly, our communication protocol is aimed at reducing the groups' confusion by having agents broadcast confusing transitions.

Our suggested protocol considers two settings.

1. **Mandatory broadcast**—agents broadcast their most confusing encountered states, i.e., states with the highest $J^p_{s_i}$. A message is a tuple consisting of the confusing transitions $< s_i, a, r, s_{i+1} >$ observed by the agent. Messages are received by all other agents, and considered as part of their experience, that is, they are inserted into the agents' replay buffers that maintain each agent's recent transitions. The size of the message is bounded by a parameter m_l that represents the channel's bandwidth.
2. **Emergent communication**—this is inspired by recent work on emergent communication in RL [6, 4]. Each agent p chooses at each time step t a discrete communication symbol m^p_t to broadcast. The individual messages of the N agents are concatenated into a single vector $m_t = [m^1_t \dots m^N_t]$, which is included as an additional observation that all agents receive at the next time step. In this setting, in which agents learn a joint communication policy, we distinguish between two sub-cases:
   (a) **Emergent self-centric communication**: Each agent p chooses a communication symbol m^p_t that would have minimized its own confusion at time step $t-1$.
   (b) **Emergent group-centric communication**: For cases where agents can observe the confusion level of all other agents at each time step, each agent p is rewarded for choosing a communication symbol m^p_t that minimizes the total confusion of the group at the next step.

While for the mandatory protocol the meaning of a message is set by the designer, for the emergent communication protocol agents need to jointly learn the semantic meaning of each message. Thus, for example, a taxi in Example 1 might broadcast an observation 'At(1,2), Passenger1_InTaxi, MOVE_UP, expected reward:-1, actual reward: -10'. In contrast, in the emergent protocol case, agents will jointly learn to effectively broadcast arbitrary symbols using their messaging policies that are optimized to minimize confusion. In addition, in the case of the emergent communication

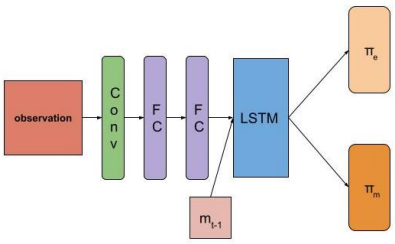


Figure 2: The communication model has two heads for two different policies. The vector of messages broadcasted is an additional input to the agent's LSTM unit as part of the observations

protocols, agents learn two separate policies (implemented via training two separate networks): π_e outputs the action to perform and is trained with the environmental reward, and π_m learns which signal to broadcast and is trained with the intrinsic reward based on the confusion level (see Figure 2).

4 Empirical Evaluation

The objective of our empirical evaluation is to assess the effect collaboration has on the resilience of a group of agents. Specifically, we measure and compare the total reward of groups of agents in randomly perturbed environments, where each group implements a different communication protocol.

4.1 Environments

Our dataset consists of three different environments.

1. **Multi-taxi:** the domain described in Example 1. The environment is episodic, resetting when all passengers arrive at their destinations. The observation of each taxi is a symbolic vector which contains raw information about the environment including the taxis' current locations and the current passengers' locations and their destinations. At the beginning of each episode, the taxis appear at random start locations, while passengers' source and destination locations are chosen at random from a set of possible locations. Each taxi receives a high positive reward for each passenger that is successfully dropped off at her destination, a small negative reward for each step in the environment, and a high negative reward when trying to drive through a wall. For each instance, the environment is represented by a 5×5 to 8×8 grid, with 2–3 taxis and 2–3 passengers.
2. **Cleanup:** An SSD domain suggested by [7] in which agents can maximize their long-term reward by cooperating and coordinating their behavior. Cells with apples (green tiles) provide individual positive reward, but are limited in number. Agents can punish each other with a *fining beam* which costs 1 and fines the agent hit with 50. The observation of each agent is a raw image consisting of its surrounding. The dilemma arises due to the adjacent river that must be cleaned so apples can grow. However, agents cannot harvest any apples while cleaning. The group therefore needs to learn to jointly keep the river clean.
3. **Harvest:** Another SSD domain suggested by [7] which is similar to the Cleanup domain. In this variation there is no river, but apples grow at rate that is proportional to the amount of nearby apples. Agents therefore need to coordinate the rate and locations from which they harvest apples.

In order to train our RL agents, we use a separate neural network per agent. For the SSD domain the agents are implemented using the Distributed Asynchronous Advantage Actor-Critic (A3C) approach by [11]. For the multi-taxi domains they are implemented using a Deep Q-Network [12]. Our neural networks' structure is inspired by the architecture presented by [6] and is depicted in Figure 2. The network consists of a convolutional layer, fully connected layers, a Long Short Term Memory (LSTM) recurrent layer [5], and output layers. All networks take raw images as input and output both the policy (π_e, π_m) and the value function V_π.

4.2 Perturbations

Given an environment M and a perturbation Φ, we measure the perturbation's magnitude as the sum of distances between all matching states as suggested by Song et al. [20] and described in Section 2.

We experiment with perturbation bounds of 50, 150 and 200. We introduced three types of perturbations to our environments:

1. **Changing the initial state** - randomly change the initial configuration of the environment. This include changing the initial position of taxis and passengers in the multi-taxi domain or the river location in the Cleanup domain.
2. **Introducing obstacles** - randomly add non-traversable obstacles (e.g walls) to the map.
3. **Reward and resource reallocation** - randomly eliminate passengers or apples from the domain map.

For each initial, unperturbed environment M, and perturbation bound K, we sample perturbed environments M' such that $\delta(M, M^{'}) \leq K$. To generate the set of perturbed environments, single perturbations are randomly generated and combined with other perturbations of the three types mentioned above. We bound the magnitude of the combined perturbations by K.

In each experiment, a perturbation frequency t_{pert} sets the number of episodes between perturbations. Accordingly, agents first train in the original non-perturbed environment for t_{pert} episodes, and then keep training in subsequent perturbed environments (we do not reset the agents training process after perturbations).

4.3 Communication Protocols

In our experiments we assume there exists a communication channel that allows agents to broadcast messages, and compare the following communication protocols.

1. **No communication:** Each agent interacts with the environment relying only on its own observations. Agents are not explicitly aware of one another, and do not communicate.
2. **Random communication:** Agents broadcast random observations from their experience.
3. **Social influence:** As suggested by [6], agents chooses to broadcast messages that will have the maximal influence on the immediate behavior of other agents.
4. **Mandatory communication:** Agents broadcast the top m_l most-confusing observations as described in Section
5. **Emergent self-centric communication:** Agents broadcast a discrete symbol that would have minimize their confusion at the previous step, as described in Section
6. **Emergent group-centric communication:** Agents observe other agents' current confusion level and broadcast a discrete symbol trying to minimize the overall confusion of the group at each step.

The last three policies are our suggested confusion-based protocols and are described in detail in Section 3.

4.4 Results

To measure the effect perturbations have on a group of agents we measured both the average utility of the group in the original environment, and the performance at each perturbed environment. We measure the utility of a group as the total reward achieved by the agents over a certain number of time steps (which we chose proportional to the amount it takes for the group's total reward to stabilize in each environment) and averaged over 10 repeats of the experiment.

Tables 1, 2 and 3 present the average maximal resilience C_K- calculated over 10 sampled environments within $K = 50$, $K = 150$, and $K = 200$ from the original environment. In parenthesis we indicate the standard deviation.

Protocol	K=50	K=150	K=200
No Communication	0.801	0.450	0.238
Random Observations	0.822	0.242	0.187
Mandatory Communication	**0.888**	**0.55**	**0.347**

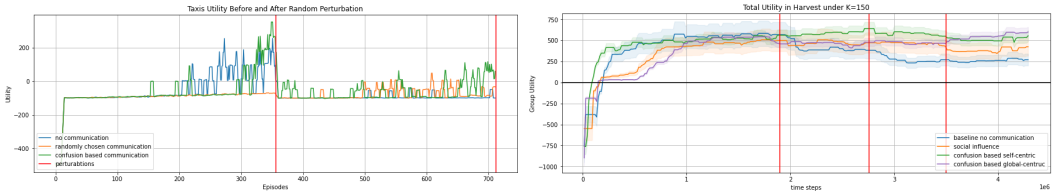
Table 1: Average maximal C_K-resilience for multi-taxi.

Protocol	K=50	K=150	K=200
No Communication	0.73 (0.11)	0.65 (0.04)	0.59 (0.2)
Social Influence	0.79 (0.05)	0.68 (0.02)	0.64 (0.14)
Emergent Global-Centric	0.82 (0.07)	0.8 (0.08)	0.71 (0.08)
Emergent Self-Centric	0.84 (0.12)	0.72 (0.02)	0.61 (0.1)

Table 2: Average maximal C_K-resilience for Cleanup.

In order to measure the resilience level under each perturbation bound K for each group, denoted by C_K we use the measurement defined in Definition 2, using the uniform distribution as Ψ, which is $\frac{avg(\mathcal{U}(M'))}{\mathcal{U}(M)}$.

Due to limitations of our current implementation, we compare for the multi-taxi domain only the no communication setting against the mandatory communication setting. Agents are trained using DQN. For the SSD domains we compare the social influence and emergent communication approaches. Agents are trained using A3C.



(a) (b)

Figure 3: Left: Average utility for $K = 200$ in the Taxi domain. Right: Average utility for $K = 150$ over repeated perturbations for Harvest.

Figure 3a shows the learning curves of three groups. Each group consists of DQN agents deployed to the multi-taxi domain. The groups depicted by the blue, orange and green curves correspond to groups that do not communicate at all, and groups that use mandatory random and mandatory confusion-based communication, respectively. It is interesting to see that the protocol that randomly selects the observations to broadcast achieves in average lower resilience than the no communication setting. This shows that merely sharing experiences between agents is not sufficient to achieve resilience. On the other hand, our specific confusion-based approach does lead to significantly improved resilience, showing the merit of using confusion to determine what information to share. Thus, we conclude that the confusion based communication protocol contains valuable properties for achieving high performance in randomly perturbed environments.

Protocol	K=50	K=150	K=200
No Communication	0.82 (0.15)	0.68 (0.1)	0.649 (0.13)
Social Influence	0.93 (0.03)	0.71 (0.13)	0.71 (0.17)
Emergent Global-Centric	0.93 (0.06)	0.91 (0.06)	0.85 (0.03)
Emergent Self-Centric	0.91 (0.02)	0.86 (0.09)	0.80 (0.02)

Table 3: Average maximal C_K-resilience for Harvest.

Figure 3b shows the learning curves of the groups implementing the emergent protocols deployed to the Harvest SSD domain. Results show that the groups that use the confusion-based protocol achieve higher performance in the perturbed environments.

Table 1 presents the average C_K-resilience calculated in the Taxi domain using the mandatory communication protocols described above. Tables 2 and 3 present the average C_K-resilience calculated in the SSD domains using the emergent communication protocols. As can be seen from the results the confusion based protocols achieves the highest values of C_K.

The results presented above show that collaboration, and the use of *confusion based communication* in particular, improves group resilience in the presence of perturbations of high magnitude. We conclude that *confusing* observations encapsulate valuable information that helps agents learn about the perturbed environment. In addition, instead of sharing raw observations with each other, agents can learn to use a communication channel in order to reduce the confusion level of the group.

Our results open many questions for future work. Specifically, we intend to directly compare the mandatory and emergent communication regimes.

5 Conclusion

We introduced novel formulations to evaluate the resilience of a group of agents based on the group's ability to adapt to perturbations in the environment. To the best of our knowledge, this is the first measurement of group resilience that is relevant to MARL settings. In addition, we suggested a novel *confusion-based communication* protocol to promote group resilience. Our evaluation shows that collaboration via our confusion-based communication protocol improves the group's resilience over the examined baselines.

Confusion based communication, as shown in this work, is a promising approach for promoting collaboration with many extensions. Some examples are to use this logic in pricing information when trying to estimate the value of information that is broadcast among agents, or to incorporate it in robotic systems trying to achieve cooperation in unstable environments.

The ability of autonomous agents, individually or as a group, to adapt to changes in the environment is highly desirable in real-world settings where dynamic environments are the rule, not the exception. Therefore, if a a group is to reliably pursue its objective function, it should be able to handle unexpected environmental changes. Obviously, those who have delegated tasks to autonomous agents stand to benefit from those agents being more likely to succeed. Societal benefit of resilience is thus clear, assuming the original tasks were of societal benefit themselves.

It is noteworthy that the recent global pandemic perturbed many aspects of the environments in which we operate. In such cases, people used to certain kinds of collaboration before the pandemic may have found it easier adjusting to the unfamiliar constraints that were imposed. We believe our results reflect a quite specific kind of benefit that automated agents can derive from collaborating with one other. We do note that many usual caveats on AI research apply, for instance, where the original task itself is not of societal benefit; We leave this for future work and note potential solutions in existing work on differential privacy or federated learning.

References

[1] H. B. Ammar, E. Eaton, M. E. Taylor, D. C. Mocanu, K. Driessens, G. Weiss, and K. Tüyls. An automated measure of MDP similarity for transfer in reinforcement learning. *AAAI Workshop - Technical Report*, WS-14-07:31–37, 2014.

[2] T. G. Dietterich. An overview of MAXQ hierarchical reinforcement learning. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 1864:26–44, 2000.

[3] R. Dobrushin. Prescribing a system of random variables by conditional distributions. *Theory of Probability and Its Applications*, 15:458–486, 1970.

[4] J. N. Foerster, Y. M. Assael, N. De Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, pages 2145–2153, 2016.

[5] F. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with lstm. In *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, volume 2, pages 850–855 vol.2, 1999.

[6] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. A. Ortega, D. J. Strouse, J. Z. Leibo, and N. de Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:5372–5381, 2019.

[7] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel. Multi-agent reinforcement learning in sequential social dilemmas. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 1:464–473, 2017.

[8] Y. Liang and B. Li. Parallel knowledge transfer in multi-agent reinforcement learning. *arXiv*, 2020.

[9] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 2017-December:6380–6391, 2017.

[10] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson. MAVEN: Multi-agent variational exploration, 2019.

[11] V. Mnih, A. P. Badia, L. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *33rd International Conference on Machine Learning, ICML 2016*, 4:2850–2869, 2016.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[13] S. Padakandla. A Survey of Reinforcement Learning Algorithms for Dynamically Varying Environments. *arXiv*, pages 1–15, 2020.

[14] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary. Robust Deep Reinforcement Learning with adversarial attacks. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 3:2040–2042, 2018.

[15] R. Portelas, C. Colas, L. Weng, K. Hofmann, and P. Y. Oudeyer. Automatic curriculum learning for deep RL: A short survey. *IJCAI International Joint Conference on Artificial Intelligence*, 2021-Janua:4819–4825, 2020.

[16] Y. Qian, J. Wu, R. Wang, F. Zhu, and W. Zhang. Survey on Reinforcement Learning Applications in Communication Networks. *Journal of Communications and Information Networks*, 4(2), 2019.

[17] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement Learning. In *35th International Conference on Machine Learning, ICML 2018*, volume 10, 2018.

[18] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar. Resilient Flocking for Mobile Robot Teams. *IEEE Robotics and Automation Letters*, 2(2):1039–1046, 2017.

[19] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pages 1–21, 2016.

[20] J. Song, Y. Gao, H. Wang, and B. An. Measuring the distance between finite Markov decision processes. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 468–476, 2016.

[21] E. Vinitsky, Y. Du, K. Parvate, K. Jang, P. Abbeel, and A. Bayen. Robust Reinforcement Learning using Adversarial Populations. *arXiv*, 2020.

[22] C. Z. Xu, J. Rao, and X. Bu. URL: A unified reinforcement learning approach for autonomic cloud management. *Journal of Parallel and Distributed Computing*, 72(2):95–105, 2012.

[23] T. Zhang, W. Zhang, and M. M. Gupta. Resilient robots: Concept, review, and future directions. *Robotics*, 6(4):1–14, 2017.

[24] Z. Zhu, K. Lin, and J. Zhou. Transfer learning in Deep Reinforcement Learning: A survey, sep 2020.

A Appendix

A.1 General Variation of the C_K Definition

In some use-cases, we may have knowledge about the optimal utility that could be achieved in a certain setting, or even know all the possible perturbed environments (under some perturbation magnitude). For these scenarios, we give two supplementary definitions. The strongest would not place any conditions on $M^{'}$ other than distance from the original environment M and the least strong:

Definition 7 (Relative to Optimum C_K-resilience) *Given an MDP M and a bound $K \in \mathbb{R}$, we say that a group of agents α is* universally C_K-resilient *in M if*

$$\forall M^{'} : \delta(M, M^{'}) \leq K \implies$$
$$\mathcal{U}(M') \geq C_K \cdot \mathcal{U}^*(M')$$

where $\mathcal{U}^*(M') = \max_\pi \mathcal{U}(\pi, M')$

The second definition is designed for settings where we only know or especially care about the preserved utility of the group of agents relatively to the non-perturbed environment.

Definition 8 (Relative to Origin C_K-resilience) *Given an MDP M and a bound $K \in \mathbb{R}$, we say that a group of agents α is* universally C_K-resilient *in M if*

$$\forall M^{'} : \delta(M, M^{'}) \leq K \implies$$
$$\mathcal{U}(M') \geq C_K \cdot \mathcal{U}(M)$$

A.2 Additional Results

In this section we present results of our methodology in another domain, which we call "apple picking". In this domain, agents have to visit as many "apple" states as they can to collect a reward at each such state. Once all apples have been picked, the episode ends.

The perturbations we experimented with are similar as in section 5: adding obstacles and changing the possible "apple" locations.

We note that this environment in a way is easier than the original one, as the agents get reward for even completing part of the task. It is interesting to see in Table-4 that in this setting, where the objective is less complex, the different group of agents achieve similar resilience levels. However, confusion-based communication still achieves the highest resilience throughout our experiments.

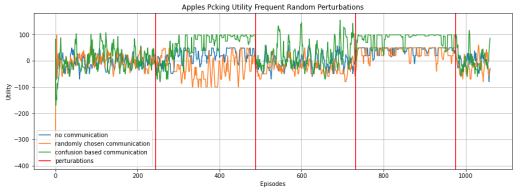


Figure 4: Apple Picking domain, Average utility for $K = 150$

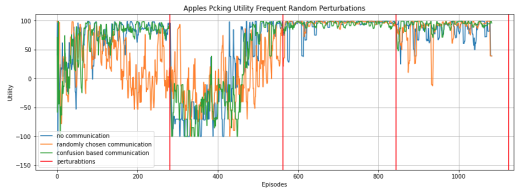


Figure 5: Apple Picking domain, Average utility for $K = 50$

Table 4: C-resilience values for groups of agents with different communication protocols in the "apple picking" domain, calculated over 10 experiments.

Communication Protocol	**K=50**	**K=150**	**K=200**
No communication	0.844	0.418	0.265
Communication of randomly selected observations	0.719	0.433	0.262
Confusion based communication	**0.881**	**0.451**	**0.327**

Table 5: C-resilience values for groups of agents with different communication protocols in the "apple picking" domain, calculated over 10 experiments.