EMERGENT COMMUNICATION IN NETWORKED MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

With the ever increasing demand and the resultant reduced quality of services, the focus has shifted towards easing network congestion to enable more efficient flow in systems like traffic, supply chains and electrical grids. A step in this direction is to re-imagine the traditional heuristics based training of systems as this approach is incapable of modelling the involved dynamics. While one can apply Multi-Agent Reinforcement Learning (MARL) to such problems by considering each vertex in the network as an agent, most MARL-based models assume the agents to be independent. In many real-world tasks, agents need to behave as a group, rather than as a collection of individuals. In this paper, we propose a framework that induces cooperation and coordination amongst agents, connected via an underlying network, using emergent communication in a MARL-based setup. We formulate the problem in a general network setting and demonstrate the utility of communication in networks with the help of a case study on traffic systems. Furthermore, we study the emergent communication protocol and show the formation of distinct communities with grounded vocabulary. To the best of our knowledge, this is the only work that studies emergent language in a networked MARL setting.

1 INTRODUCTION

Co-existing intelligent agents affect each other in non-trivial ways. Consider for example, two agents playing a modified variant of archery in two dimensions. Agent A controls the speed at which the arrow is released but it can only shoot along y-axis. Agent B controls wind speed along x-axis. The arrow drifts along the direction of wind with a magnitude proportional to wind speed. A target is specified by (x, y) coordinates and the agents must act cooperatively to shoot the target. In this setup, the optimal action for agent A depends on the current policy of agent B and vice versa. Any change in one agent's policy modifies the other agent's perception about the environment dynamics. Formally, this issue is referred to as non-stationarity of environment in a multi-agent setup. This non-stationarity makes the learning problem hard and approaches that try to independently learn optimal behavior for agents do not perform well in practice (Tan, 1993). Thus, it is important to develop models that have been tailored towards training multiple agents simultaneously.

In this paper, we focus on a specific multi-agent setup where the agents are connected to each other via an underlying fixed network topology. We cast the problem in the multi-agent reinforcement learning (MARL) framework and assume that agents are rewarded by the environment based on their actions and their goal is to cooperatively maximize their rewards. We further assume that the agents have been endowed with the ability to communicate with one another along the network edges to achieve cooperation. However, the communication protocol is not fixed and the agents must learn a protocol to communicate with each other in order to maximize their rewards.

Communication is essential in a multi-agent setup. In many practical scenarios, agents may only observe a small portion of the global environment state and they must take actions based on their local observations. As discussed above, agents affect each other in non-trivial ways through their actions. Thus, for achieving long term cooperation, it is essential for agents to be able to share their intents to complement the information provided by the local observation of each agent. Communication provides the ability to do so to the agents.

Many real world problems can be cast in this framework and we provide a number of concrete examples after formally defining the problem setup in Section 2. For clarity of exposition and

to be more concrete, in this paper, we focus on a particular real world problem as a case study, the problem of intelligently managing traffic. We present the traffic management problem as a particular instantiation of the abstract multi-agent reinforcement learning problem that we have informally defined above (see Section 2 for a formal definition). In this context, the agents correspond to traffic lights and the underlying network is the network of roads. Agents receive rewards from the environment based on factors like queue length at the traffic junction and must communicate with each other to cooperatively maximize their rewards, thereby ensuring a smooth flow of traffic.

We propose a MARL-based traffic system that allows coordination between traffic signals (agents) via: **(i)** inter-agent communication; and **(ii)** a cooperative reward structure. At each time-step, the agents communicate with their immediate neighbours in the underlying network by broadcasting a message in the form of a discrete symbol (each message corresponds to a word, represented by a binary vector in our experiments). Over time, the agents are trained to exploit this broadcasted message to coordinate with each other and maximize their rewards. As the agents are trained, a language *emerges* between pairs of agents. Since the agents learn the communication protocol themselves, our approach is different from methods that use a fixed protocol for communication, like smart-grids.

We empirically demonstrate the utility of communication in this setup and also investigate a cooperative reward structure over the network of traffic junctions. Our model uses a *query-based soft attention* mechanism to help the agents come up with more complex cooperative strategies. We perform extensive experimental evaluation to demonstrate that **(i)** our method outperforms baseline approaches; **(ii)** communication is useful, **(iii)** communication is grounded in actions taken by the agents, and **(iv)** the cooperative reward structure promotes communication and hence coordination.

2 Problem Setup

2.1 Notation and Preliminaries

Markov games generalize the notion of a Markov decision process (MDP). A Markov game is used to model an environment where N intelligent agents co-exist. Let $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ be the set of all possible environment states (i.e., state-space) and denote by $\mathcal{O}_i \subseteq \mathbb{R}^{d_{o_i}}$ the observation space of agent i. At each step, agent i chooses an action from its action space $\mathcal{A}_i$ and in response to the actions chosen by all agents, the environment state is updated using the transition function $\mathcal{T}$: $\mathcal{S} \times \mathcal{A}_1 \times \cdots \times \mathcal{A}_N \rightarrow \Delta(\mathcal{S})$ where $\Delta(\mathcal{S})$ is the set of all probability distributions defined over set $\mathcal{S}$. The environment provides a reward to each agent at each time-step using a reward function r_i: $\mathcal{S} \times \mathcal{A}_i \rightarrow \mathbb{R}$. For $i = 1, 2, \ldots, N$, the goal of i^{th} agent is to find a policy $\boldsymbol{\pi}_{\theta_i} : \mathcal{O}_i \rightarrow \Delta(\mathcal{A}_i)$ that chooses optimal actions so as to maximize the long term reward, $\mathcal{R}_i = \sum_t \gamma^t r_i^t$ where r_i^t is the reward received by agent i at time-step t and $\gamma \in (0, 1]$ is the discount factor.

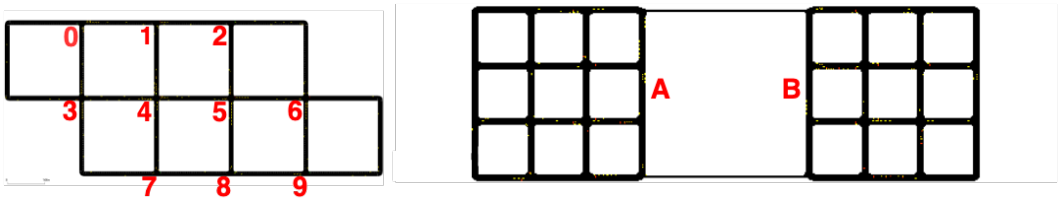

Figure 1: Traffic networks used in our experiments. **Network 1** : On the left is a 10-agent network (denoted by $\{0, \ldots, 9\}$) interconnected by bidirectional lanes. **Network 2** : On the right is a 28-agent network with 14 agents in each of the sub-networks (A & B) connected by one-way lanes.

2.2 Problem Definition

We model the problem as a Markov game with two additional assumptions: **(i)** let $\mathcal{V} = \{1, 2, \ldots, N\}$ be the set of all agents, we assume that the agents are connected to each other via an underlying network whose edge set is given by $\mathcal{E}$; and **(ii)** agents can communicate with their immediate neighbors in the underlying network. To communicate, at each step, agents broadcast a message. This message is received by the neighbors at the next time-step. The observation space of each agent is

augmented to consider the messages received from all of its neighbors in addition to considering the local observation made by that agent.

The agents act in a cooperative setting as follows: first, the environment provides a reward r_i^t to all agents. Then, each agent i, augments this reward to include the information about rewards received by its neighbors, i.e., $\bar{r}_i^t = \beta r_i^t + (1-\beta)\frac{1}{|\mathcal{U}(i)|}\sum_{j\in\mathcal{U}(i)} r_j^t$. Here, β assigns relative importance to agent's own rewards and the rewards of its neighbors ($\beta \in (0, 1]$) and $\mathcal{U}(i)$ is the set of neighbors of agent i. The agents are trained to maximize long term rewards $\bar{\mathcal{R}}_i = \sum_t \gamma^t \bar{r}_i^t$. As agents also benefit from the rewards obtained by their neighbors, they act cooperatively.

Such a setup can for instance be used to model an intelligent electrical distribution network. In this application, agents represent power stations and the underlying network corresponds to the electrical grid. Each agent has a production capacity and it may choose to share the power generated by it with a neighboring agent (action space). All agents have to meet the local demand which changes stochastically and they observe various attributes like power requirements, consumer demand and so on (observation space). Rewards r_i are provided based on how successful an agent is in meeting the demand. The agents must learn to communicate with each other to effectively share the power generated by them to maximize their cooperative long term rewards $\bar{R}_i$.

As another application, consider a supply chain where interconnected warehouses (agents) have to manage their inventory to meet the local demand. As before, agents can choose to ship goods that they have in their inventory, to their neighbors (action space). The warehouses have to meet stochastically changing consumer demands and must learn to communicate effectively in order to share goods so that an appropriate level of inventory is maintained at each warehouse. As in the case of electrical distribution network, rewards received by agents depend on their ability to effectively meet the local demand.

While many applications are possible, in this paper, we focus on the problem of traffic management as a concrete instantiation of the proposed abstract problem. Each traffic junction corresponds to an agent which are connected to each other via the underlying network of roads. Agents can control the flow of traffic in the system by modulating the traffic lights (action space). They take actions based on their local observations, which in our case, is a image containing snapshot of the immediate surroundings of the agent. Agents are rewarded based on predefined attributes like queue length at the traffic junction, average waiting time of vehicles at the junction and so on. We again train the agents to maximize the long term cooperative rewards. Clearly, in order to ensure a smooth flow of traffic, the agents must communicate with each other to share their intent. We use the SUMO traffic simulator (Krajzewicz et al., 2012) to simulate the traffic for the experiments.

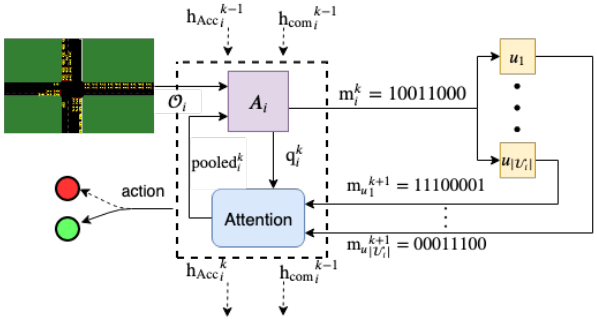

Figure 2: Visualizing an exchange of messages between agent i (enclosed by dotted lines) and its neighbours $\mathcal{U}_i$ for a traffic network.

In our setup, we use binary messages with $d = 8$. As such, agents are limited to a vocabulary of 256 words. Communication is one-hop, i.e. we only allow agents to communicate with its immediate neighbours. At each time-step, the agents broadcast 8-bit messages to its respective neighbours. Agents rely on their local observations and received messages to modulate the traffic signals and to decide the next message to be broadcasted. We opted for discrete communication rather than continuous since it is more interpretable. For the real world applications that we have mentioned above, transparency in decision making is important and using discrete communication is a step in this direction. Towards this end, we demonstrate that the emergent language is grounded, i.e. messages correspond to actions in Section 5. By forcing agents to communicate with the help of discrete symbols, we ensure that humans can inspect and interpret the conversation logs.

3 Related Work

The simplest and one of the most widely used approaches to tackle the traffic management problem is the Fixed-time Control (Miller, 1963; Webster, 1958), which uses a predefined cycle for planning. Network level control like the Max-Pressure Controller (Varaiya, 2013) differ from the conventional approaches in respect to greedily switching phases based on demand (queue length in adjoining lanes). Similarly, Self-Organizing Traffic Light Control (SOTL) (Cools et al., 2008) method switches the traffic lights based on when the number of vehicles in the lanes crosses a predefined threshold. These methods are locally adapted to the traffic conditions which in turn is used to achieve global synchronization. Optimization methods like TUC (Diakaki et al., 2000) assume uniform traffic flow in a certain time period to minimize the vehicle travel time.

We argue that conventional traffic control methods base their problem setup on assumptions which do not hold while modelling the dynamics of traffic. In the recent past, people have resorted to the use of Reinforcement Learning methods to dynamically adapt to the traffic conditions (Prabuchandran K.J. et al., 2014; Li et al., 2016; Wei et al., 2018), wherein, each junction is treated as an agent and the changes in traffic lights are actions. The agents try to maximize their long term reward by mapping a predefined observation space (representation of the traffic conditions at its junction) to the action space. However, these Deep-Q Learning (DQN) based approaches often consider the agents to be independent of one another, thus rendering the environment non-stationary, and raising convergence and stability issues.

A straightforward way to address the above problems is to make the agents coordinate amongst themselves by including the information from the neighbouring agents in its state space (Dresner & Stone, 2006; El-Tantawy et al., 2013; Du, 2019), assuming total observability. Another way would be to use centralized training (Lowe et al., 2017) to address stationary related issues. However, centralized training poses scalabilty problems (especially in a traffic-management system where the complexity increases linearly with the junctions) and cannot be used to learn policies in real time. Scalable multi-agent approaches like Van der Pol & Oliehoek (2016) propose to train a smaller source problem involving fewer agents (2 agents) and transfer it to a larger target problem using *transfer-learning*.

Prior work on extending MARL setup to networks using a fully decentralized training has been proposed by Zhang et al. (2018), wherein the authors propose to share the local parameters of the agents through communication. Drawing motivation from the recent advancements in emergent communication (Havrylov & Titov, 2017; Mordatch & Abbeel, 2017), we try to induce coordination amongst agents with partial observability. We argue that with a cooperative reward structure, agents can be made to pass on relevant information to their neighbours, which the independent agents were previously incapable of modelling using their own observation space. Concurrent to our work, Sukhbaatar et al. (2016) proposed a similar idea of using communication in traffic networks. Their approach differs from ours in the following ways: **(i)** They consider cars as agents, whereas we formulate this as a network problems with nodes, which are fixed, as agents; **(ii)** They use continuous communication, whereas we use discrete communication in our setup for better interpretability; **(iii)** In their setup, every agent can communicate with the other agents, whereas we restrict the communication to its immediate neighbours with a priority assigned to each message.

4 Proposed Method

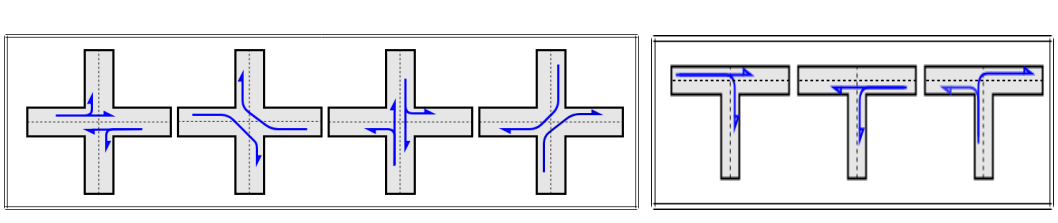

Figure 3: Permitted actions for 4-way and 3-way junctions. [Left-hand traffic]

Image Encoder: As shown in Fig. 1, we use a 10-agent traffic network (Network 1) which comprises both, 3-way as well as 4-way junctions. The observation space of the agents in the traffic

network comprises the image representation of the junction. This amorphous representation of the observation space propels the agent to extract necessary information from the image, like length of the queue in the adjoining lanes or the current phase at the junction. One can always add these information to the state directly and, in general, adding more information to the state space often leads to better results (Havrylov & Titov, 2017), nonetheless, such systems carry redundant information and are impractical to deploy. For instance, consider lining the lanes with sensors to calculate the number of vehicles. A Convolution Neural Network (CNN) (Lecun et al., 1998) is used to process the input image and extract specific features required for optimizing traffic.

Accumulator: The agents, then, use this information to take actions. Since, it is unreasonable to change the traffic lights every second, we constrain the agents to take an action once in every t time-steps ($t = 5s$ in our setup). The Accumulator is a Recurrent Neural Network (RNN) (Hochreiter & Schmidhuber, 1997) which keeps track of the observations for t time-steps before an action is taken. Let the encoded image information of agent i at time-step k be o_i^k, given as, $\mathrm{o}_i^k = f_{\mathrm{CNN}i}(o_i^k)$, wherein, $o_i \in \mathcal{O}_i$; $\mathrm{o} \in \mathbb{R}^{d_o}$ (d_o is the output dimension of the encoder CNN). The Accumulator uses this encoded vector to update its memory by $\mathrm{h}_{\mathrm{Acc}i}{}^k = f_{\mathrm{Acc}i}(\mathrm{o}_i^k, \mathrm{h}_{\mathrm{Acc}i}{}^{k-1})$. Here $\mathrm{h}_{\mathrm{Acc}i}{}^k \in \mathbb{R}^{d_h}$, where d_h is the hidden dimension of the Accumulator.

Communicator: In order to establish complex cooperative strategies, the agents must learn to prioritize messages received from its neighbours. In that context, we incorporate a query-based soft-attention mechanism in our communication setup. At each time-step, an agent generates a query (q) which is used to assign weights to the received messages in the previous time-step. Intuitively, an agent enquires about the unaccounted information (which it cannot model) in $\mathcal{O}_i$ and the weights (α) are a way to understand the contributions of its neighbours in the state information. Let us assume that the messages broadcasted are denoted by m, where $\mathrm{m} \in \mathcal{M}$ ($\mathcal{M}$ is set of all messages). For agent i at k^{th} time-step with neighbours belonging to the set $\mathcal{U}_i$,

$$\begin{aligned} \mathrm{q}_i^k &= \mathrm{W}_i \mathrm{h}_{\mathrm{Acc}i}{}^k \\ \alpha_i^k &= \mathrm{softmax}\left[\mathrm{q}_i^{k^T}\mathrm{m}_1^{k-1}, \ldots, \mathrm{q}_i^{k^T}\mathrm{m}_{|\mathcal{U}_i|}^{k-1}\right] \\ \mathrm{pooled}_i^k &= \sum_{j \in |\mathcal{U}_i|} \alpha_{ij}^k \mathrm{m}_j^{k-1} \end{aligned} \tag{1}$$

Here $\mathrm{W} \in \mathbb{R}^{d \times d_h}$, $\mathrm{q} \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}^d$ are the attention parameters. At each time-step, the pooled message is fed to the Communicator RNN and is processed as $\mathrm{h}_{\mathrm{com}i}{}^k = f_{\mathrm{com}i}(\mathrm{pooled}_i^k, \mathrm{h}_{\mathrm{com}i}{}^{k-1})$. It should be noted that we reinitialize the hidden state of the RNN as soon as the action is taken (i.e. after every t time-steps). A weighted mixture of the output of the RNN along with the output of the image encoder (o_i^k) is passed through a setup which emits discrete symbols with the help of *Straight-Through Gumbel-Softmax* ($\mathbb{G}$) (Jang et al., 2016), i.e. $\mathrm{m}_i^{k+1} = \mathbb{G}(\mathrm{W}_{\mathrm{com}}\mathrm{h}_{\mathrm{com}i}{}^k + \mathrm{W}_{\mathrm{CNN}}\mathrm{o}_i^k)$. This renders the model differentiable even with discrete symbols. The output message is then broadcasted to all its neighbours ($\mathcal{U}_i$). Refer Appendix A.2 for more details.

Action Selector: A 4-way junction can have 2^{16} combinations of actions (2^9 for a 3-way junction), however, most of them are either highly dangerous for traffic or don't contribute to a smoother flow. Adopting conventional approaches (Linkenheld et al., 1992), we fix the action space to 4 actions for a 4-way junction and 3 for a 3-way junction (Fig. 3). The actions are determined such that the vehicles can move smoothly without conflicts. The action values are a function of Accumulator-RNN and Communicator-RNN hidden states i.e. $f_{\mathrm{action}i}(\mathrm{h}_{\mathrm{Acc}i}{}^k, \mathrm{h}_{\mathrm{com}i}{}^k)$. At every t^{th} time-step, the action values are generated and the traffic lights are switched to the action with the maximum value.

Reward Structure: The reward at a junction i is a combination of the following factors: **(i)** Queue Length: Total number of waiting vehicles (with speed $< 0.1m/s$) on the adjoining lanes; **(ii)** Waiting Time: Sum of waiting time of all vehicles in the adjoining lanes, wherein, a vehicle is considered to be waiting if it travels at a speed $< 0.1m/s$; **(iii)** Delay: For lane l, delay is calculated as $D_i^l = 1 - \frac{\mathrm{avg\ speed}_i^l}{\mathrm{max\ speed}_i^l}$. Here $\mathrm{avg\ speed}$ and $\mathrm{max\ speed}$ denotes the average speed of vehicles and the maximum allowable speed, respectively, on lane l; **(iv)** Emergency Deceleration: The number of times, a vehicle undergoes emergency braking during switching of phases.

Communication What gives the agents the incentive to communicate? An agent might turn out to be selfish and focus on solving its own problems even while it broadcasts unstructured messages to its neighbours. Even if the agents are forced to communicate, what makes us believe that the agents will pay any heed to the messages that they receive, considering the individual reward structure? In view of this, we highlight two important aspects of communication: **(i)** Cooperation: Meaningful information exchange will arise only if there is a dearth of knowledge in the local observations of the agents. In other words, there are factors that the agent is uncertain of and cannot model in the current setup. To see why, let us consider a scenario where there are two interconnected independent agents (say A and B) trying to model the traffic. The arrival rate of agent A is modelled using $\mathrm{Poisson}$, with rate λ, which is a function of the environment. As agent B takes an action, the environment is no longer stationary and λ varies in a way that agent A is incapable of modelling (let us assume λ now becomes $(\lambda - \delta\lambda)$). Hence, the agent learns to ignore these variables altogether. In order to make sense of the unaccounted changes in its observation space, agent B has to communicate the difference in λ to agent A which can be assumed to be $\mathrm{Poisson}$ distributed with rate $\delta\lambda$ (superposition of independent Poisson processes results in a new Poisson process whose rate is the sum of the rates of the independent Poisson processes); **(ii)** Coordination: Even while groups of agents implicitly come up with a common communication protocol, there should be a higher level harmony amongst all agents, i.e. the groups of agents in a network must systematically synchronize or coordinate to realize the same goal. In order to drive the agents to work for the greater good, they must arrive at a common protocol which can be used universally. This leads to the development of a common language with overlapping vocabularies from different communities of agents. Additionally, if an agent in the sub-network fails, the other agents in that group can still communicate with other groups.

5 EXPERIMENTS

Comparison with baselines: We compare our model with the following baselines:

(i) Fixed-time Control: We use our action space and periodically switch from one action to the other in a round-robin fashion after fixed time intervals of duration $t = 5$ steps.

(ii) Self Organizing Traffic Light control (SOTL): SOTL (Cools et al., 2008) switches from one phase to other when the queue length at any of the adjoining lanes exceed a predefined threshold. In our implementation, we fix this threshold to 5 while retaining our action space.

(iii) Deep-Q Learning (DQN): We also implement the standard DQN framework with the same observation and action space as ours where each agent is trained independently.

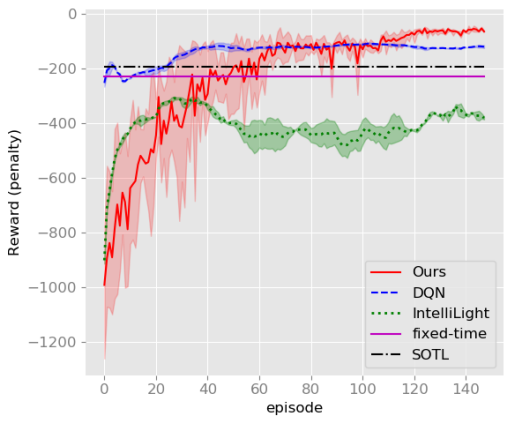

Figure 4: Comparison of convergence of different baselines with our setup. (Averaged over 5 independent runs)

(iv) IntelliLight: IntelliLight (Wei et al., 2018) uses a more refined state representation which includes queue length, number of vehicles and updated waiting time at the adjoining lanes of a junction, in addition to its image representation. We replicate their architecture with the same state representation but with our action space, since their action space is fraught with danger in a real life traffic management system.

The results (Fig. 4) reveal that our method outperforms all baselines in terms of the end result (final reward at convergence). While DQN training commences at a high reward (because of high initial value of ϵ in the ϵ-greedy strategy), it converges quickly without much improvement. We also wish to point out that parameter updates in DQN happen every time-step once the buffer-size exceeds the batch-size, as opposed to once every episode in our setup. On an average, the difference between the final rewards at convergence of the DQN and the our setup is $\approx \mathbf{55}$.

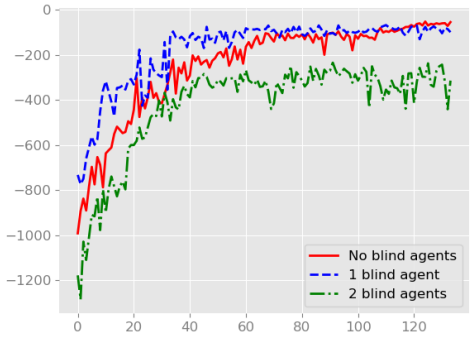

Figure 5: Comparison of performance (Reward vs. episodes) of blind agents.

Necessity of communication: An important question that naturally follows is whether communication is required at all. It may indeed be possible for the agents to adapt even while they broadcast random set of symbols which do not bear any significance. To test this possibility, we perform two sets of experiments:

(i) We made the agents broadcast 8-bit messages with all bits set to zero. We found that the training not only converged slowly, it also stabilized at a lower reward compared to the original setting. Post convergence the difference in rewards was $\approx \mathbf{40}$.

(ii) We visually impaired one of the agents (we call it a blind-agent) such that it no longer receives its local observation while taking an action, although it can still receive messages from its neighbours and transmit its state information to them i.e. the agent processes its input observation at each time-step but the weight $\mathrm{W}_{\mathrm{CNN}}$ is set to zero. Hence, it cannot backpropagate valuable gradients from its actions to the image encoder. We noticed that the overall setup still converged to the same result as the original setup. We attribute it to the fact that the blind-agent receives necessary information from its neighbours through communication. On repeating the experiment for two neighbouring blind-agents, the convergence worsens (Fig. 5). Thus, we can safely conclude that communication not only works but is also necessary. We performed the above experiments on agents 4 & 5 (Network 1 of Fig. 1).

Grounding in communication: In order to verify grounding, we performed experiments using Pointwise Mutual Information (PMI) (Church & Hanks, 1990). We constructed a matrix for each pair of agents where the rows correspond to the actions of one agent (say i) and the columns correspond to the message sent by the other agent in the pair. We computed the Singular Value Decomposition (SVD) of the PMI matrix ($\mathrm{pmi} \in \mathbb{R}^{|\mathcal{A}_i| \times 256}$) given as $\mathrm{pmi} = \mathrm{USV}^{\mathrm{T}}$, where $\mathrm{U} \in \mathbb{R}^{|\mathcal{A}_i| \times k}$, $\mathrm{S} \in \mathbb{R}^{k \times k}$ and $\mathrm{V} \in \mathbb{R}^{k \times 256}$ ($k \leq |\mathcal{A}_i|$). Figure 6 shows the t-SNE (van der Maaten & Hinton, 2008) plot of the rows of V matrix. The rows of V can be interpreted as word embeddings.

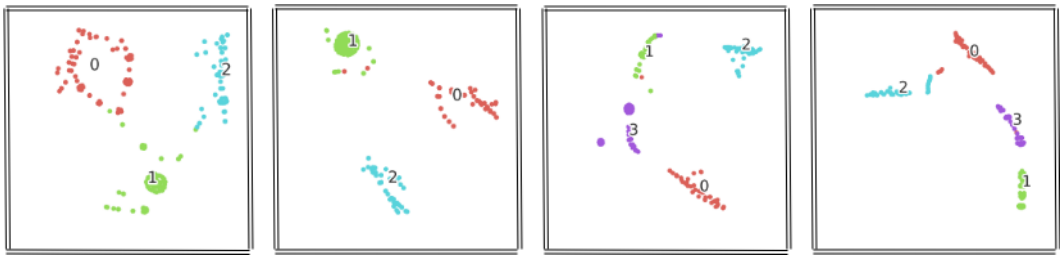

Figure 6: **[Best viewed in colour]** Word embeddings of neighbour j ($j \in \mathcal{U}_i$) corresponding to actions of agent i for a 3-way junction (indexed by $\{0, 1, 2\}$) and for a 4-way junction (indexed by $\{0, 1, 2, 3\}$. The clusters can be interpreted as distinct words used to mean different actions. 3-way junctions have an action space of size 3 and 4-way junctions have an action space of size 4.

As shown in Fig. 6, we see distinct clusters forming for different actions. In other words, we can say that neighbouring agents use specific set of words to indicate actions. The words which are haphazardly placed can be considered as those that haven't been assigned a specific meaning or the ones which can be used in many different contexts. Additionally, we plot the the rows of U matrix (with $k = 2$) i.e. the action embeddings corresponding to the broadcasted messages of all agents. For instance, as shown in Fig. 8, we pair the actions of agent 0 with the broadcasted messages from the rest of the agents. For each such pair, we get a U matrix which we plot using different colors for different agents. We notice that the points corresponding to the U matrix of the neighbours tend to overlap as highlighted by the red circles (for agent 0, agents 1 and 3 overlaps). This implies that neighbours $\mathcal{U}_0$ are consistent in the use of messages to agent 0, which in turn bases its actions on the received messages. See Appendix A.3 for the remaining plots on grounding.

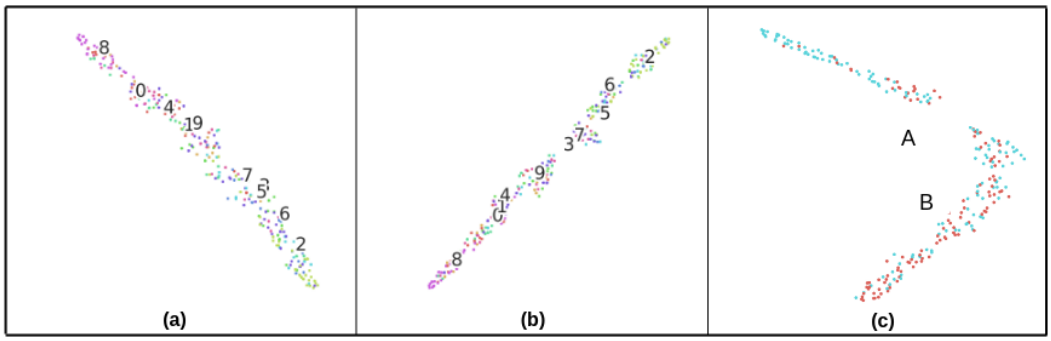

Figure 7: **[Best viewed in colour]**. Clustering of agents in the 10-agent network (Network 1) for different runs [Fig.(a,b)]. The numbers $\{0, \ldots, 1\}$ denote the agents in the network. Fig.(c) on the right represents the clustering in the 28-agent network (Network 2) with A & B denoting two 14-agent sub-networks. The position of the numbers denote the mean of the clusters.

Effect of network topology: We obtain a tf-idf matrix where rows correspond to agents and columns correspond to the words spoken by these agents. On plotting a t-SNE plot of the rows of this matrix for Network 1 (Fig. 1), we noticed that the agents that are broadcasting to the same neighbour tend to be clustered together. For instance, from Fig. 7 [(a), (b)], one can infer that the following triads are formed: **(i)** agents (0 & 4) broadcasting to agent 1; **(ii)** agents (3 & 7) broadcasting to agent 4; **(iii)** agents (3 & 5) broadcasting to agent 4. It is also consistent with our findings in Fig. 8, wherein actions embeddings, corresponding to the neighbours, overlap. These trends reflect that one agent can dominate the emergent communication protocol among its neighbours. Nevertheless, as is evident from the plots, a common set of vocabulary also comes into usage to make communication span across all agents.

Experiments on larger networks: Due to communication, we expect well-defined communities to emerge (in terms of the vocabulary usage) if huge networks are sparsely connected to one another. To test this hypothesis, we take a pair of networks, each having 14 agents and connect them by a single one-way lane (Network 2 in Fig. 1). We plot the t-SNE embeddings of the rows of tf-idf matrix (Fig. 7 (c)). We notice two distinct clusters (denoted by A and B) with a overlapping region, which we argue, comprises the common vocabulary used by agents from both networks. To sum it up, we not only find evidence of cooperation among agents but also coordination among groups of agents across networks. On removing the cooperative reward structure, the model takes twice as much time to converge, presumably because the agents have little incentive to communicate.

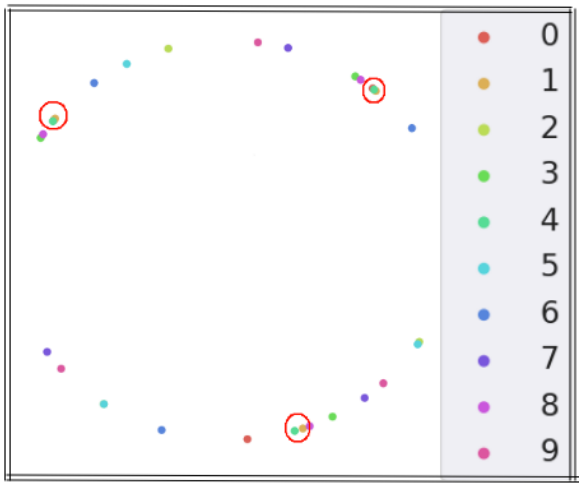

Figure 8: **[Best viewed in colour]**. Action embeddings from matrix U (orthonormal matrix) of agent 0 corresponding to messages from all agents. As highlighted by the red circles, the action embeddings of agent 0 in response to messages from neighbours $(1, 3)$ are overlapped. The color bar represents different agents.

6 Conclusion

In this paper, we proposed an approach to mitigate network congestion with the help of traffic networks. Though extensive experiments, we demonstrated the benefit of emergent communication to optimize traffic flow over existing MARL approaches. Additionally, we performed qualitative studies on the emergent language and showed that it is grounded in actions.

REFERENCES

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, 1990.

Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. *Self-Organizing Traffic Lights: A Realistic Simulation*, pp. 41–50. 2008.

Christina Diakaki, Markos Papageorgiou, and Kostas Aboudolas. A multivariable regulator approach to traffic-responsive network-wide signal control. *IFAC Proceedings Volumes*, 33:561–566, 2000.

Kurt Dresner and Peter Stone. Multiagent traffic management: Opportunities for multiagent learning. In *Proceedings of the First International Conference on Learning and Adaption in Multi-Agent Systems*, LAMAS'05, pp. 129–138, 2006.

Yunshu Du. Improving deep reinforcement learning via transfer. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, pp. 2405–2407, 2019.

S. El-Tantawy, B. Abdulhai, and H. Abdelgawad. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3): 1140–1150, 2013.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems 30*, pp. 2149–2159. 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *CoRR*, abs/1611.01144, 2016.

Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker-Walz. Recent development and applications of sumo - simulation of urban mobility. *International Journal On Advances in Systems and Measurements*, 3,4, 2012.

Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

L. Li, Y. Lv, and F. Wang. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 3(3):247–254, 2016.

J. S. Linkenheld, Rahim F Benekohal, and J. H. Garrett. Knowledge-based system for design of signalized intersections. *Transportation engineering journal of ASCE*, 118:241–257, 1992.

Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems 30*, pp. 6379–6390. 2017.

Alan J. Miller. Settings for fixed-cycle traffic signals. *Journal of the Operational Research Society*, 14(4):373–386, 1963.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *AAAI*, 2017.

Prabuchandran K.J., Hemanth Kumar A.N, and S. Bhatnagar. Multi-agent reinforcement learning for traffic signal control. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 2529–2534, 2014.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2244–2252. 2016.

M. Tan. Multi-agent reinforcement learning - independent vs cooperative agents. *Proceedings of the tenth International Conference on Machine Learning (ICML)*, pp. 330–337, 1993.

L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9(nov):2579–2605, 2008.

Elise Van der Pol and Frans A. Oliehoek. Coordinated deep reinforcement learners for traffic light control. In *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems*, 2016.

Pravin Varaiya. The max-pressure controller for arbitrary networks of signalized intersections. *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pp. 27–66, 2013.

F. V. Webster. *Traffic signal settings*. Paper No. 39, Road Research Laboratory, England, published by HMSO, 1958.

Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '18, pp. 2496–2505, 2018.

K. Zhang, Z. Yang, and T. Basar. Networked multi-agent reinforcement learning in continuous spaces. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2771–2776, 2018.

A APPENDIX

A.1 TABLE OF NOTATIONS

$\mathcal{S}$	Agent state configurations
$\mathcal{A}_i$	Action of agent i
$\mathcal{O}_i$	Observation space of agent i
$\mathcal{T}$	Transition function
$\boldsymbol{\pi}_{\theta_i}$	stochastic policy of agent i
r_i^k	reward obtained by agent i at k_{th} time-step
$\mathcal{R}_i$	long-term reward of agent i over all time-steps in an episode
γ	discount factor
$\mathcal{V}$	set of all agents $\{1, 2, \ldots, N\}$
$\mathcal{M}$	set of all possible broadcasted messages (words) ($\mathcal{M} \in \{0,1\}^d$)
m	$\mathrm{m} \in \mathcal{M}$
q	query vectors
d	dimension of messages ($d = 8$)
$\mathcal{U}_i$	set of neighbours of agent i
$\bar{r}_i^k$	cooperative reward (weighted combination of r of $\{i, \mathcal{U}_i\}$)
β	relative importance to agent's own rewards ($\beta \in (0, 1]$)
$\bar{\mathcal{R}}_i$	cumulative cooperative reward for agent i
f_{CNN}	Image Encoder model (CNN)
f_{Acc}	Accumulator model (RNN)
f_{com}	Communicator model (RNN)
o	$o \in \mathcal{O}$
o	encoded image (encoded observation)
$\mathrm{h}_{\mathrm{Acc}}$	hidden state of the Accumulator
d_o	output dimension of the image encoder
d_h	hidden dimension of the Accumulator RNN
W	attention weights for $\mathrm{h}_{\mathrm{Acc}}$
α	attention scores ($0 \leq \alpha \leq 1$)
pooled	attention output (convex combination of messages from $\mathcal{U}_i$)
$\mathrm{W}_{\mathrm{com}}$	weights for $\mathrm{h}_{\mathrm{com}}$ in the weighted mixture
$\mathrm{W}_{\mathrm{CNN}}$	weights for o in the weighted mixture
$\mathbb{G}$	*Straight-Through Gumbel Softmax*
f_{action}	action-value function
pmi	PMI matrix ($\mathrm{pmi} \in \mathrm{pmi} \in \mathbb{R}^{\lvert\mathcal{A}_i\rvert \times 256}$)
$\mathrm{U}, \mathrm{S}, \mathrm{V}$	SVD output ($\mathrm{U} \in \mathbb{R}^{\lvert\mathcal{A}_i\rvert \times k}$, $\mathrm{S} \in \mathbb{R}^{k \times k}$, $\mathrm{V} \in \mathbb{R}^{k \times 256}$ ($k \leq \lvert\mathcal{A}_i\rvert$)

A.2 STRAIGHT-THROUGH GUMBEL-SOFTMAX

Human communication is discrete in nature and can, in general, be represented by categorical variables. Additionally, discrete variables are more interpretable which makes it well suited for real life problems like traffic management, where one needs transparency. However, the use of discrete latent variables render the neural network non-differentiable. The Gumbel Softmax gives a differentiable sample from a discrete distribution by approximating the hard one-hot vector into a soft version.

The Gumbel distribution has two parameters: μ and β. The standard Gumbel distribution where μ and β are 0,1 respectively has probability density function: $\mathrm{G}(0, 1) = e^{-(z+e^{-z})}$. Suppose, for a given agent, our model (Communicator) outputs a $\mathrm{Multinomial}$ distribution of message bits with logits : $\mathbf{p} = (p^1, \ldots, p^d)$ where $d = 8$. These logits are functions of inputs and weights which need to be trained. A simple way to draw samples from a discrete distribution would be to use the Gumbel-max trick (Jang et al., 2016) as shown,

$$\tilde{\mathrm{m}}^i = \mathrm{one\_hot}\left(\arg\max_i [\mathrm{z}^i + \log p^i]\right) \tag{2}$$

Here, the noise in form of z^i is independently sampled from standard Gumbel distribution and are obtained as $\mathrm{z}^i = -\log(-\log(u^i))$, u_i being i.i.d samples from $\mathrm{Uniform}(0, 1)$. It turns out

that m_i is distributed according to p^i. However, gradients still cannot propagate through $\arg\max$, hence a $\mathrm{softmax}$ is used as a differentiable alternative, to generate a d-dimensional sample vector $\tilde{\mathbf{m}} = (\tilde{\mathrm{m}}^1, \ldots, \tilde{\mathrm{m}}^d)$, where $\tilde{\mathrm{m}}^i$ is given as,

$$\tilde{\mathrm{m}^i} = \frac{e^{(\mathrm{z}^i + \log p^i)/\beta}}{\sum_{j=1}^{d} e^{(\mathrm{z}^j + \log p^j)/\beta}} \tag{3}$$

β is the temperature parameter, which, in our experiments, is set to 0.5. As $\beta > 0$, we obtain well-defined gradients $\partial \mathrm{m}_i / \partial p_i$ with respect to parameters p_i. Gumbel-Softmax can produce a 0.9999-hot vector instead of a hard one-hot vector depending on β. Since we want the communication to be discrete, we employ the *Straight-Through* version of the Gumbel-Softmax estimator with a simple reformulation of the form,

$$\mathrm{m}^i = (\hat{\mathrm{m}^i} - \tilde{\mathrm{m}^i}).\mathrm{detach}() + \tilde{\mathrm{m}^i} \tag{4}$$

Here, $\hat{\mathrm{m}^i}$ is the one-hot vector of $\tilde{\mathrm{m}^i}$ i.e. such that $\hat{\mathrm{m}^i} = \mathbf{I}\{\arg\max_i \tilde{\mathrm{m}^i} = k, k \leq k^{'}\}$. We use binary 8-bit messages, therefore $k^{'} = 1$ and k is fixed during the training process. Now, detach prevents the gradients from flowing through that node, hence, $\nabla_p{}^i \mathrm{m}^i = \nabla_p{}^i \tilde{\mathrm{m}^i}$. This makes the communication discrete in the forward pass even as the gradients flow is smooth during the backward pass. The final output message of agent is given as $\mathbf{m} = (\mathrm{m}^1, \ldots, \mathrm{m}^d)$.

A.3 COMMUNICATION

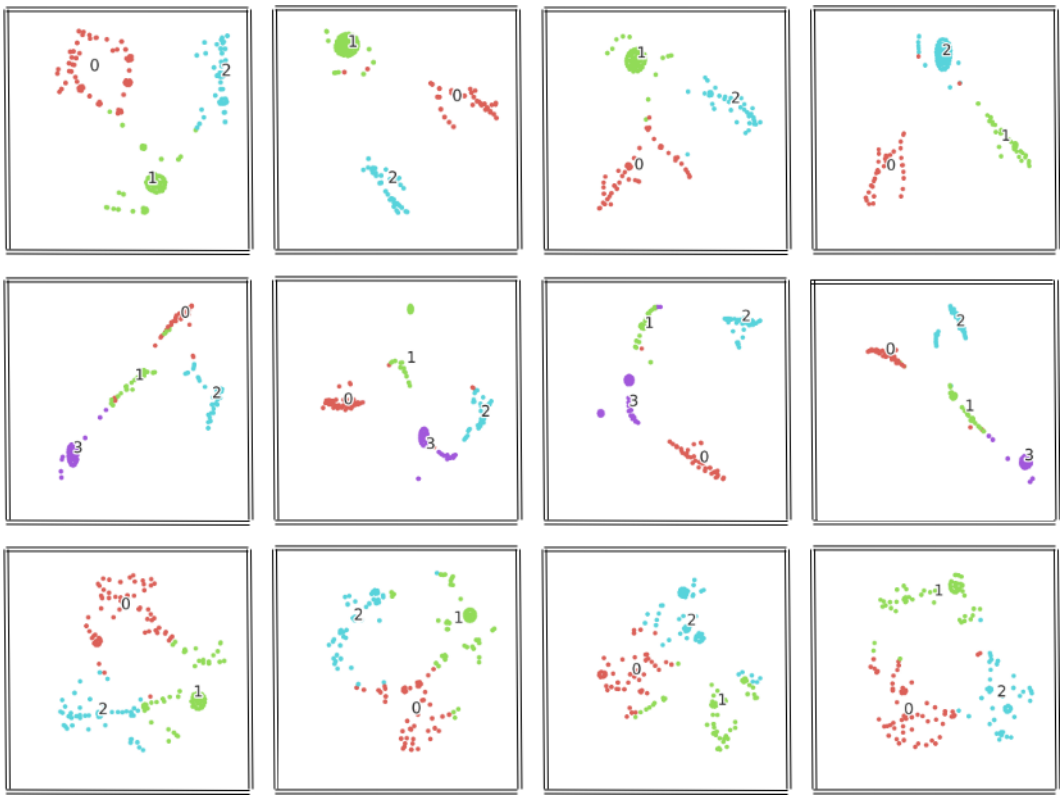

Figure 9: **[Best viewed in colour]** Word embeddings of neighbour j ($j \in \mathcal{U}_i$) corresponding to actions of agent i for a 3-way junction (indexed by $\{0, 1, 2\}$) and for a 4-way junction (indexed by $\{0, 1, 2, 3\}$). The clusters can be interpreted as distinct words used to mean different actions. 3-way junctions (Rows 1,3) have an action space of size 3 and 4-way junctions (Row 2) have an action space of size 4.

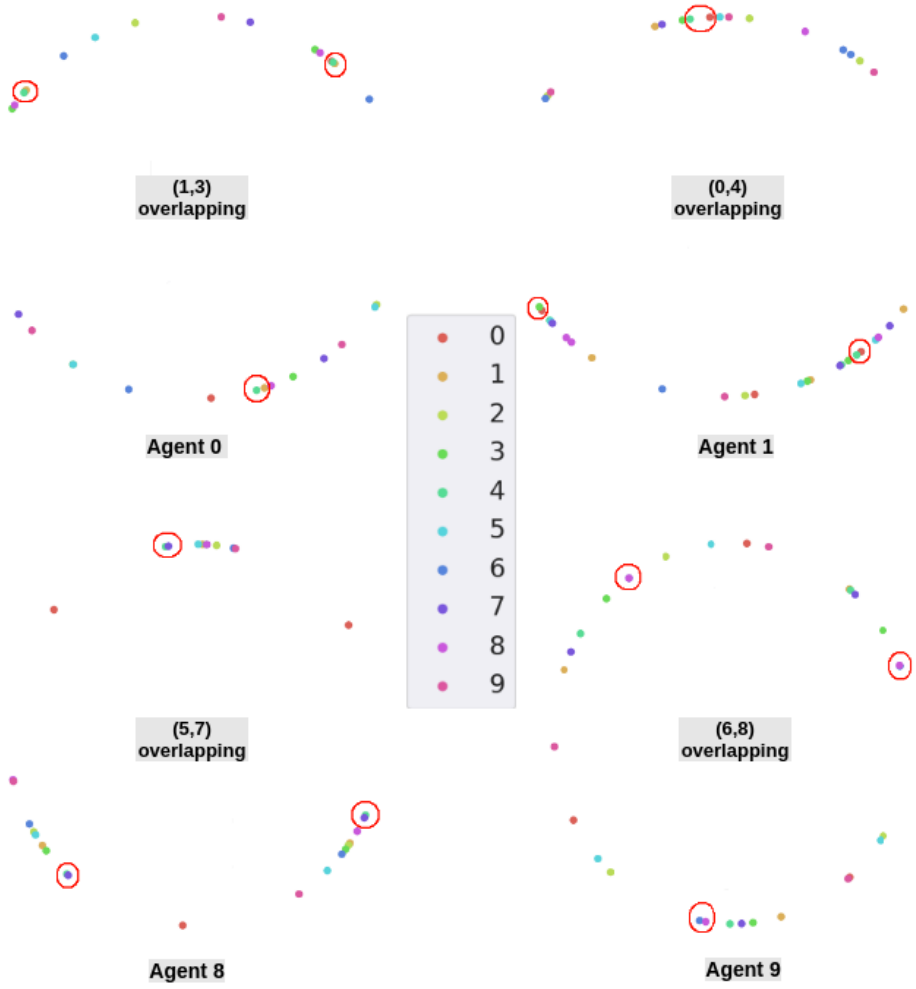

Figure 10: **[Best viewed in colour]**. Action embeddings from matrix U (orthonormal matrix) of agent 0 corresponding to messages from all agents. As highlighted by the red circles, the action embeddings of agent i in response to messages from neighbours $\mathcal{U}_i$ (as highlighted in gray at the centres of the plots) are overlapped. The color bar represents different agents. Here, we plots are for agents $0, 1, 8, 9$. Similar trends are observed for rest of the agents as well.