SOCIAL NETWORK STRUCTURE SHAPES INNOVATION: EXPERIENCE-SHARING IN RL WITH SAPIENS

Anonymous authors
Paper under double-blind review

ABSTRACT

The human cultural repertoire relies on innovation: our ability to continuously explore how existing elements can be combined to create new ones. Innovation is not solitary, it relies on collective accumulation and merging of previous solutions. Machine learning approaches commonly assume that fully connected multi-agent networks are best suited for innovation. However, human laboratory and field studies have shown that hierarchical innovation is more robustly achieved by dynamic social network structures. In dynamic settings, humans oscillate between innovating individually or in small clusters, and then sharing outcomes with others. To our knowledge, the role of multi-agent topology on innovation has not been systematically studied in machine learning. It remains unclear a) which social network structures are optimal for which innovation tasks, and b) which properties of experience sharing improve multi-level innovation. Here we use a multi-level hierarchical problem setting (WordCraft), with three different innovation tasks. We systematically design networks of DQNs sharing experiences from their replay buffers in varying structures (fully connected, small world, dynamic, ring). Comparing the level of innovation achieved by different experience-sharing social network structures across different tasks shows that, first, consistent with human findings, experience sharing within a dynamic structure achieves the highest level of innovation across tasks. Second, experience sharing is not as helpful when there is a single clear path to innovation. Third, two metrics we propose, conformity and diversity of shared experience, can explain the success of different social network structures on different tasks. These contributions can advance our understanding of optimal AI-AI, human-human, and human-AI collaborative networks, inspiring future tools for fostering collective innovation in large organizations.

1 INTRODUCTION

Unlike herds or swarms, human social networks solve different tasks with different social network structures (Momennejad, 2022). Human and computational studies show that, to optimize the performance of collective learning by choosing who should communicate with whom, one needs to take into account the inter-play between task and network properties: networks with high connectivity are better suited for quick convergence in problems with clear global optima (Coman et al., 2016; Momennejad et al., 2019) while partially-connected structures perform best in *deceptive* tasks, where acting greedily in the short-term leads to missing the optimal trajectory (Derex & Boyd, 2016; Lazer & Friedman, 2007; Cantor et al., 2021). Despite this evidence, works in decentralized reinforcement learning (RL) have not to date considered the effect of group connectivity, with most solutions adopting a fully-connected architecture with a single learner and multiple actors (Mnih et al., 2016; Horgan et al., 2018; Espeholt et al., 2018; Nair et al., 2015; Christianos et al., 2020; Schmitt et al., 2019). Here, we test the performance of different social network structures in groups of RL agents that share their individual experiences. We refer to such entities as multi-agent topologies, introduce SAPIENS, an algorithm for Structuring multi-Agent toPology for Innovation through ExperieNce Sharing[1], and evaluate it in a type of deceptive task that models collective innovation.

[1]We provide an implementation of SAPIENS and code to reproduce the simulations we report in `https://anonymous.4open.science/r/SAPIENS-859E`.

Innovations represent the expansion of an agent's behavioral repertoire with new problem-solving abilities and are, therefore, a necessary ingredient of continuous learning and exploration (Leibo et al., 2019). The solitary genius narrative often obscures the collective nature of human innovations, which arise from tinkering, recombination and adoption of existing innovations (Solé et al., 2013; Derex & Boyd, 2016). Cognitive and ecological studies of human technological innovation characterize it as a type of combinatorial search constrained by semantics that describe which combinations of innovations are feasible (Solé et al., 2013; Derex & Boyd, 2016). Here, we adopt this definition, so that innovation tasks: a) exhibit a multi-level search space, where innovations arise out of recombination of existing ones(Hafner, 2021) b) are characterized by rewards that increase monotonically with the level of innovation. This property captures the human intrinsic motivation for progress (Solé et al., 2013) and contributes to the deceptive nature of these tasks, attracting agents toward potentially sub-optimal paths.

Laboratory and field studies of human groups have shown that collective innovation is highly contingent on the social network structure (Momennejad, 2022; Migliano & Vinicius, 2022; Derex & Boyd, 2016). The reason for this lies in the exploration versus exploitation dynamics of social networks: partial connectivity leads to high clustering and long shortest path, which helps maintain diversity in the collective at the benefit of exploration. In contrast, high-connectivity quickly leads to conformity which benefits exploitation (Lazer & Friedman, 2007). Of particular interest are social network structures that achieve a balance in this trade-off: small-worlds are static graphs that, due a modular structure with long-range connections, achieve both high clustering and small shortest path Watts & Strogatz (1998). Another example are dynamic structures, where agents are assigned to distinct groups but are able to periodically visit other groups (Volz & Meyers, 2007). These graphs have the attractive property that they both locally protect innovations, thus maintaining diversity, and allow sharing so that agents can recombine the solutions of the collective and discover the global optimum quickly in large search spaces (Derex & Boyd, 2016).

In this paper we study the effect of social network structure in multi-agent topologies that solve innovation tasks. We propose SAPIENS, an algorithm connecting DQN learners (Mnih et al., 2013) that independently explore the environment and share experience tuples from their replay buffers with their neighbors in different static and dynamic social network structures. We employ Wordcraft (Jiang et al., 2020) as a test-bed and design three custom tasks (Figures 1 and 2) covering innovation challenges of different complexity: (i) a task with a single innovation path to an easy-to-find global optimum. This type of task can be used to model a linear innovation structure, such as the evolution of the fork from knife to having, two-, three- and, eventually four tines. (Solé et al., 2013) (ii) a task with two paths that individually lead to local optima, but when combined, can merge toward the global optimum. With this task we can capture innovations that were repurposed after their invention, such as Gutenberg's screw press leading to the print press (Solé et al., 2013). (iii) a task with ten paths, only one of which leads to the global optimum, which captures search in vast spaces.. In addition to Wordcraft, we also evaluate SAPIENS on a deceptive task implemented in a grid world.

We empirically show that the performance of SAPIENS depends on the inter-play between social network structure and task demands. Dynamic structures perform most robustly, exhibiting an impressive ability to converge quickly in the easy task, avoid local optima in the second task, and explore efficiently in the third task. We propose and compute measures of conformity and diversity (Momennejad et al., 2019) to interpret these findings and compare the performance of SAPIENS with other decentralized RL algorithms (Mnih et al., 2016; Nair et al., 2015).

Contributions Our contributions are two-fold. **From a neuroscience and cognitive science perspective**, SAPIENS is, to our knowledge, the first computational study that attempts at reproducing results observed in human groups on the effect of social network structure on collective innovation, that employs deep RL as the individual learning mechanism. Compared to previous computational studies that employed simple agent-based models and evolutionary search (Lazer & Friedman, 2007; Cantor et al., 2021; Mason & Watts, 2012), deep RL provides three main advantages : i) it enables empirical experiments with more complex test-beds and larger search spaces ii) agents can share their experience by simply exchanging transitions from their respective replay buffers, without requiring ad-hoc mechanisms for copying the behaviors of other agents iii) by using the replay buffer as a proxy of the memories of agents, we can directly measure properties such as diversity and alignment of experiences that are challenging to estimate with human studies. **From an RL perspective**, our work extends upon the decentralized RL paradigm by systematically analyzing the

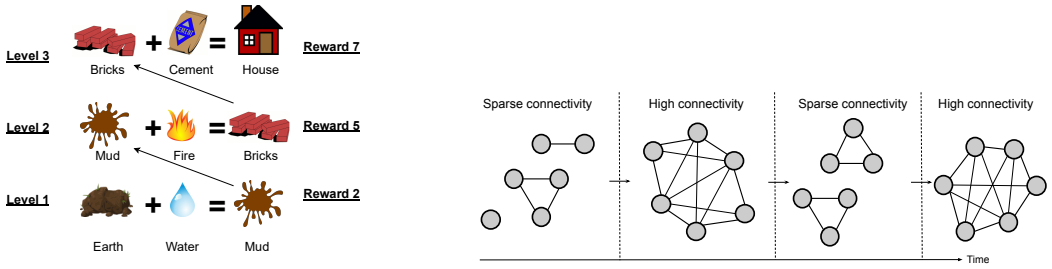

Figure 1: (Left) Illustration of an innovation task, which consists of an initial set of elements (Earth, Water) and a recipe book that determines which element combinations create new elements. Upon creating a new element the player moves up an innovation level and receives a reward that increases monotonically with levels (Right) Dynamic social network structures oscillate between phases of low connectivity, where experience sharing takes place within clusters, and high connectivity, where experiences spread between clusters

effect of different social network structures on different class of tasks. We propose novel metrics enabling a thorough analysis of the learning dynamics. In summary, our contributions provide a better understanding of results originally obtained in human experiments and build upon these insights to propose a novel algorithm for distributed RL.

Related work Experience sharing among RL agents can significantly improve both convergence rate and performance (Horgan et al., 2018; Mnih et al., 2016; Schmitt et al., 2019; Nair et al., 2015). Shared experience may take the form of experience tuples or gradients, the former being common among off-policy algorithms (Souza et al., 2019; Horgan et al., 2018; Mnih et al., 2016) and baring benefits over sharing gradients in actor-critic architectures, where instability and latency are concerns (Schmitt et al., 2019; Christianos et al., 2020). For instance, the Gorila framework (Nair et al., 2015), introduced a general multi-learner architecture with both local and global replay buffers, but focused on parameter sharing rather than experience sharing. A common architecture for experience sharing is that of multiple actors and a single learner (Mnih et al., 2016; Horgan et al., 2018; Schmitt et al., 2019). Another approach included experience sharing between multiple actors/learners but the groups only included two agents (Souza et al., 2019).

2 METHODS

2.1 WORDCRAFT: A TEST-BED FOR INNOVATION

We perform experiments on Wordcraft (Jiang et al., 2020), an environment inspired from the game Little Alchemy 2 [2]. Tasks start with an initial set of elements and the player explores combinations of two elements in order to create new ones. Figure 1 (Left) provides an illustrative example.

In Wordcraft one can create different types of tasks using a *recipe book* ($\mathcal{X}_{valid}$), which is a dictionary containing the valid element combinations with entries in the form $(x, y) \rightarrow z$, where x and y indicate the two chosen elements and z the newly crafted one. When queried with a non-valid combination X_{valid} returns no element. In addition to the recipe book, a task requires a reward function R_{valid} that returns a scalar reward associated with crafting element z. A valid combination returns a new element and reward only the first time it is chosen. When queried with a non-valid combination, $\mathcal{R}_{valid}$ returns a reward of zero. Thus, a task can be be described by a tuple $(\mathcal{X}_0, \mathcal{X}_{valid}, \mathcal{R}_{valid}, T)$, where $\mathcal{X}_0$ denotes the initial set of elements and T is the number of time steps available to an agent before the environment resets. The environment can therefore be modeled as a fully-observable Markov Decision Process (MDP) that we define in detail in Appendix A.1.

In addition to Wordcraft, we also show the applicability of our method to another domain: a deceptive game implemented in a grid world (Appendix A.5).

[2] https://littlealchemy2.com/.

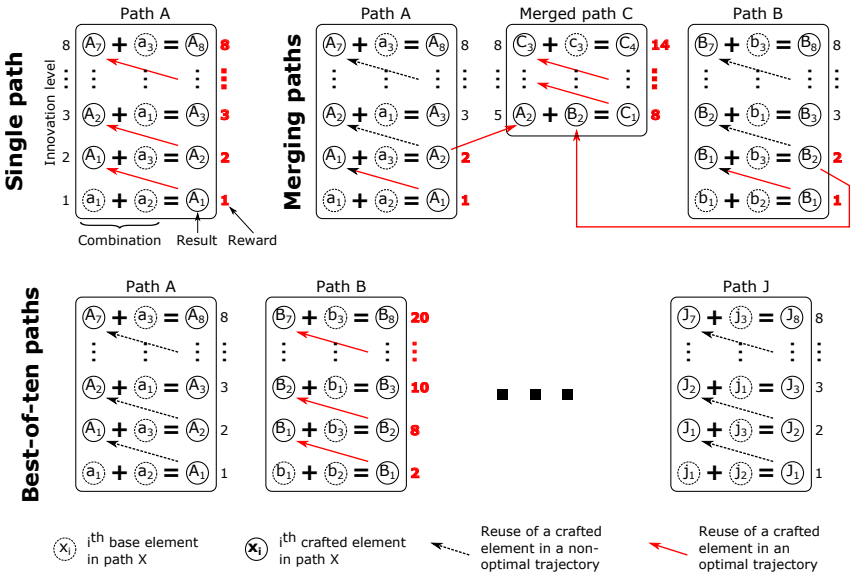

Figure 2: We introduce three innovation tasks called single path, merging paths and best-of-ten paths, described in Section 2.2. Each task contains one or more paths, labeled by an uppercase letter (A to J). Each path X has its own initial set of three base elements $\{x_1, x_2, x_3\}$, which are represented in dashed circles. Crafted elements in path X are represented in upper case (X_i) in solid circles. Optimal trajectories for each tasks are represented by solid red arrows, with their corresponding reward in bold red.

2.2 Innovation tasks

We design three innovation tasks, shown in Figure 2, that pose different challenges:

Single innovation path The recipe book consists of an initial set of 3 base elements ($\mathcal{X}_{valid} = \{a_1, a_2, a_3\}$) and 8 innovation levels. An agent needs to first combine two of them to create the first element and then progresses further by combining the most recently created element with an appropriate one from the initial set. This optimization problem contains a single global optimum.

Merging paths There are two paths, A and B, and at level 2 there is a cross-road that presents the player with three options: moving forward on path A, moving forward on path B (until level 8), or combining elements from path A and B to progress on path C. The latter path is more rewarding and is, thus, the optimal choice. This task is particularly challenging: to explore path C the player needs to first go to lower innovation levels instead of just progressing on a single path. It thus exhibits two local optimum and one global optimum

Best-of-ten paths Here, one of the ten paths is the most rewarding. The optimal strategy is to move only on the most rewarding path but, to do so, the player must first explore and reject the other paths. This optimization task is characterized by a single global optimum and aims at evaluating the ability of agents to explore effectively in large search spaces.

We introduce the following concepts to characterize the structure of innovation tasks. An innovation task can contain one or more paths, which can potentially be connected to each other (as in the merging paths task). In our proposed tasks, an **innovation path** X is defined as a sequence of elements $[X_1, ..., X_n]$, where crafting an element X_i ($i > 1$) requires to combine the previously crafted element X_{i-1} and a base element from the initial set. The first element X_1 in the sequence requires combining elements from the initial set or from other paths. The **innovation level** of an element corresponds to the length of the path it belongs to, plus the sum of the path lengths required to craft its first item (0 if the first item is crafted from base elements). For example, the innovation level of A_3 in the single path task is 3, while the innovation level of element C_1 in the merging paths task is 5 (1 on C + 2 on A + 2 on B). Within an innovation task, the **trajectory** of an agent is defined

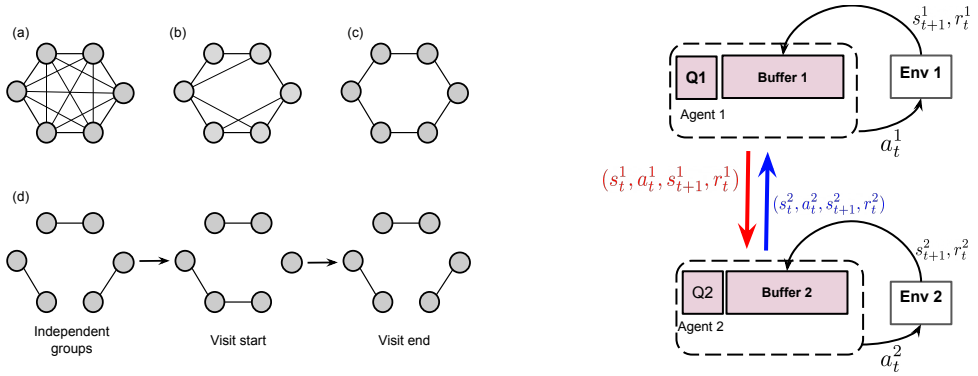

Figure 3: (Left) Social network structures (a) fully-connected (b) small-world (c) ring (d) dynamic. (Right) Schematic of two neighboring DQNs sharing experiences: agent 1 shares experiences from its own replay buffer to that of agent 2 (red arrow) and vice versa (blue arrow) while both agents are independently collecting experiences by interacting with their own copy of the environment.

as the sequence of crafted elements it produces. Finally, the **optimal trajectory** of an innovation task is the trajectory that returns the highest cumulative reward within the problem horizon T.

2.3 Learning algorithm

SAPIENS considers a group of K DQN agents, where each agent interacts with its own copy of the environment. An agent can share experiences with others by sampling transition tuples from its own replay buffer and directly inserting them to the replay buffer of another agent. An undirected graph $\mathcal{G}$ determines who shares information with whom, i.e. the multi-agent topology, and consists of a set of nodes representing agents and a set of edges indicating that two agents share experiences with each other. We define the neighborhood of agent k as the set of nodes connected to it and denote it as $\mathcal{N}_k$. At the end of each episode the agent shares experiences with each of its neighbors with probability p_s: sharing consists of sampling a subset of experiences of length L_s from its own buffer B_k and inserting it in the buffers of all neighbors $B_n, n \in \mathcal{N}_k$. Thus, an agent communicates distinct experiences with each of its neighbors. We present a schematic of two DQN agents sharing experiences on the right of Figure 3 and the pseudocode of our learning algorithm in Appendix A.2. We also provide more implementation details about the DQN in Appendix A.1.

We visualize the multi-agent topologies studied in our work in Figure 3: fully-connected, small-world, ring and dynamic. We construct the latter topology by grouping the agents in pairs and then allowing agents to visit other groups with a pre-determined, static duration, T_v and probability p_v. This is a common type of dynamic topology that has been used in human laboratory and field studies Derex & Boyd (2016); Migliano & Vinicius (2022) and studied for its graph-theoretic properties (Volz & Meyers, 2007). We provide more information about dynamic topologies in Appendix A.3, where we study how its behavior changes with different values of its hyper-parameters $T_v, p + v$ and present an alternative type of dynamic topology.

2.4 Evaluation framework

During evaluation trials, which take place every E_{eval} training steps with experience sharing deactivated, we compute the following metrics (indexed by the training step t):

Performance-based metrics This family of metrics describes the quality of the final solution and the speed of convergence to it, which are the two main properties of interest when evaluating decentralized RL algorithms. Specifically we measure: (i) R_t^+: the maximum reward of the group at training step t; (ii) R_t^*: the average reward of the group at training step t; (iii) T^+, Time to first success: the first training step at which at least one of the agents collected the maximum achievable reward; (iv) T^*, Time to all successes: the first training step at which all of the agents found the optimal solution (v) $T^>$, Spread time: number of training steps required for the optimal solution to

spread to the whole group, once at least one member discovered it (equals $T^* - T^+$) (vi) $\mathcal{S}$, group success, a binary variable denoting whether at least one agent in a group found the optimal solution

To get a deeper understanding of why the performances of groups varies with different topologies, we propose to extend the classical performance-based metrics with *behavioral metrics* (characterizing the policies followed by the agents) and *mnemonic metrics* (characterizing the replay buffer of agents), either at an individual or group level.

Behavioral metrics This family of metrics characterizes the policies followed by the agents. Specifically: (i) conformity C_t is a group-level metric that denotes the percentage of agents in a group that followed the same trajectory in a given evaluation trial (ii) volatility V_t is an agent-level metric that denotes the cumulative number of changes in the trajectory followed by an agent.

Mnemonic metrics These metrics capture properties of the replay buffer. In particular: (i) diversity D_t^k is an agent-level metric that denotes the number of unique experiences in an agent's replay buffer; (ii) $D_t^{\mathcal{G}}$ is a group-level metric that captures the diversity of the aggregated group buffer. To compute it we concatenate the replay buffers of all agents into a single one and compute its diversity.

3 RESULTS

We now evaluate the social network structures presented in Figure 3 on the innovation tasks described in Section 2.2 and visualized in Figure 2. Specifically, methods ring, *small-world*, dynamic and fully-connected are instantiations of SAPIENS for different social network structures with 10 DQNs. In all experiments shared experiences are sampled randomly from the replay buffers. We benchmark SAPIENS against: a) no-sharing, a setting with 10 DQN agents without experience sharing b) single, a single DQN agent c) A2C, a decentralized policy gradient RL algorithm that employs a single learner and 10 workers sharing gradients (Mnih et al., 2016) and, finally d) Ape-X, a decentralized RL algorithm with 10 workers that share experience samples with a single DQN learner (Horgan et al., 2018). We perform 20 independent trials report statistical significance testing results conducted with the Welch test (Colas et al., 2018). We refer readers to Appendix A.4.1 for information about how we tuned each method A.4.1).

3.1 OVERALL COMPARISON

In Figure 4 we compare performance across tasks and methods (We refer readers to Appendix A.4 for plots of reward with time for all conditions). We observe that performance varies across tasks and topologies: the single path task is optimally solved by all multi-agent topologies with the fully-connected group being the slowest to converge and A2C being the quickest, in agreement with the expectation that policy-gradient methods are more sample efficient than value function methods. Failure to converge was observed in a few trials for a single agent and Ape-X due to slow convergence. In the merging paths task the performance of the dynamic structure is significantly better than all other baselines except for no-sharing (p-value 0.22) and small-world (p-value 0.07). In contrast, fully-connected, A2C and Ape-X perform significantly worse. In this case, A2C and Ape-X show a similar qualitative behavior to fully-connected: extensive experience sharing leads to consistent convergence to local optimal. This indicates that, while learners that do not share experiences manage to solve the task with relative success, learners that share experiences under social networks combining large clustering and small shortest path perform best. In the best-of-10 paths task, the dynamic topology is again the best-performing, with the single agent failing due to slow convergence and other methods failing due to convergence to local optima.

In additional experiments we have also: a) observed that these conclusions are consistent across network sizes, with the performance of partially-connected social network structures improving with increasing network size and the performance of the fully-connected decreasing (see Appendix A.4.5) b) Observed a drop in performance under prioritized experience sharing, where the DQNs employ prioritized replay buffers and experiences are selected for sharing proportionally to their priority (see Appendix A.4.6) . This behavior has also been observed in other works (Souza et al., 2019) and we hypothesize that it is due to the fact that priorities computed by an agent do not necessarily agree with the priorities of the other c) analyzed how the performance of the dynamic topology varies with its hyper-parameters and derived suggestions for tuning it (see Appendix A.3) d) monitored

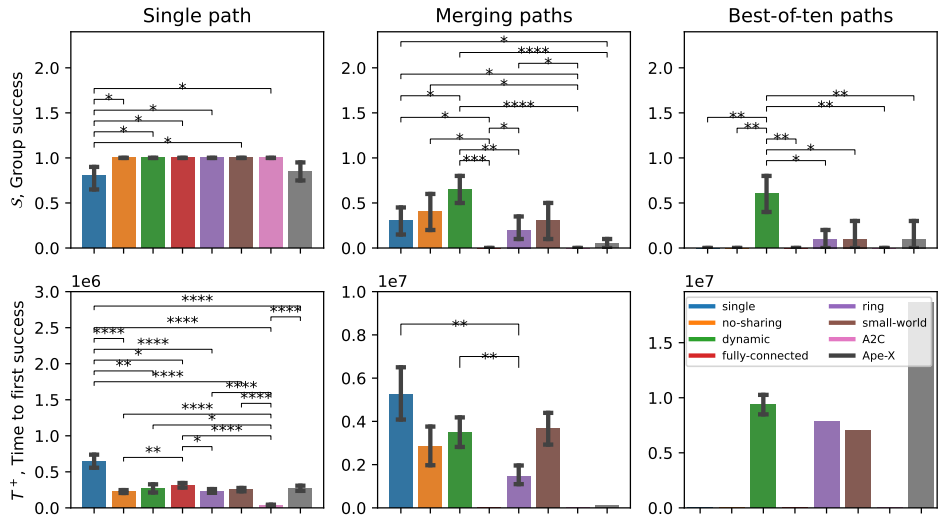

Figure 4: Overall performance comparison for the single path (first column), merging paths (second column) and best-of-ten paths (third column) task. We present two metrics: group success ($\mathcal{S}$) denotes whether at least one agent in the group found the optimal solution (top row) and T^+, Time to first success, is the number of training time steps required for this event (bottom row). Note that T^+ can be computed only for $\mathcal{S} > 0$ its error bars and significance tests can only be computer for $\mathcal{S} > 1$. We denote statistical significance levels with asterisks.)

the robustness of social network structures to different learning hyper-parameters and observed that the dynamic topology is more robust than fully-connected and no-sharing (see Section A.4.4) and e) computed measures of the alignment of experiences within and across groups that further support our hypothesis that the content of replay buffers is highly contingent on the multi-agent topology (see Appendix A.4.3) f) performed additional simulations in another deceptive test-bed and derived similar conclusions to what we observed in Wordcraft, namely that partially-connected structures are better at avoiding local optima (See Appendix A.5)

To explain the differences for SAPIENS under different structures we now study each task in isolation focusing on the behavioral and mnemonic metrics described in Section 2.4. We will focus on the key metrics here and refer readers for values of all metrics to Appendix A.4.2.

3.2 TASK: SINGLE PATH

Although all multi-agent topologies found the optimal solution, experience sharing did not improve performance in this task (see Appendix A.4.2 for values of all performance-based metrics). It even slows down learning when connectivity is high (fully-connected group). While it is not surprising that a single agent with epsilon-greedy exploration can efficiently solve this task, it is not clear at first glance why experience sharing harms performance. Notably this phenomenon has been observed in related works (Souza et al., 2019; Schmitt et al., 2019). Perhaps surprisingly, the fully-connected group does not have the highest conformity (Figure 5, Left). We therefore conclude that there is an upper threshold for connectivity, beyond which shared experiences destabilize learning. We can detect this destabilization when looking at volatility V_t, which is highest for the fully-connected setting (Figure 6, Top left). We believe that this instability is caused by the diversity in the replay buffer of each agent, captured by the average individual diversity metric $\bar{D}_t$ (Figure 5, Right): agents in the fully-connected network have the most diverse buffers while agents in the no-sharing setting have the least diverse buffers. As DQNs sample their replay buffer randomly and the task is simple, low diversity of individual buffers leads to quicker convergence.

Finally, when looking at the group diversity in Figure 6, fully-connected ranks last and the dynamic topology ranks first. This is rather surprising: the no-sharing setting does not rank first, but expe-

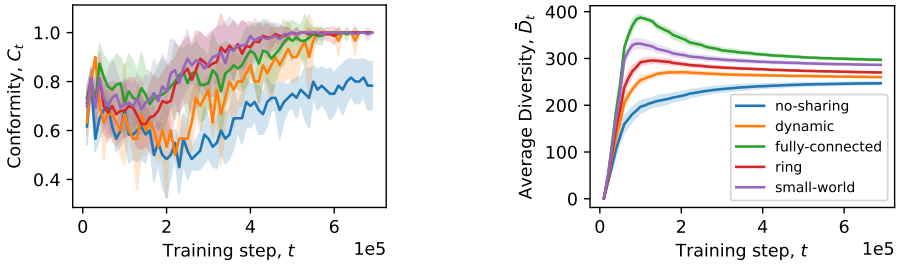

Figure 5: Analyzing group behavior in the single path task: (Left) Conformity C_t is a behavioral metric that denotes the percentage of agents in a group that followed the same trajectory in a given evaluation trial. (Right) Average Diversity $\bar{D}_t$ is a mnemonic metric that denotes the number of unique experiences in the replay buffer of an agent, averaged over all agents

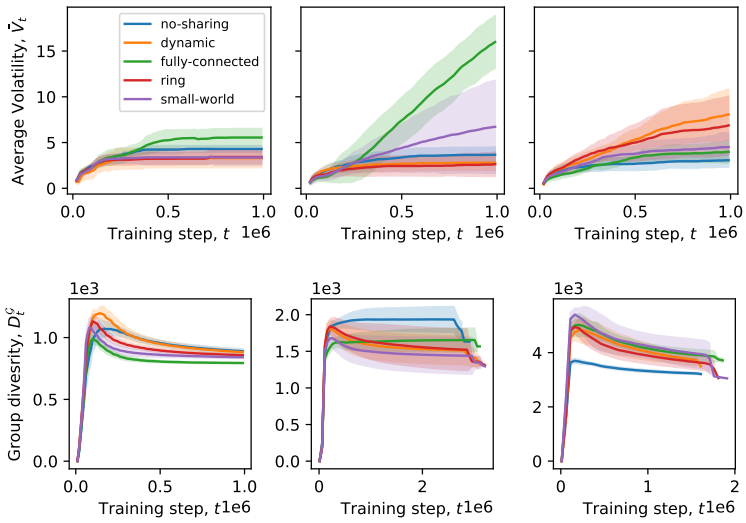

Figure 6: Analyzing group behavior in the single path task (first column), merging paths task (second column) and best-of-10 paths task (third column). On the top row, the Average volatility (V_t) is a behavioral metric indicating the cumulative number of changes in the trajectory followed by an agent, averaged by all agents. On the bottom row, Group Diversity $D_t^{\mathcal{G}}$ is a mnemonic metric that captures the diversity of the aggregated group buffer.

rience sharing increases the diversity of a group when connectivity is low, and, most prevalently, when the topology is dynamic. This indicates that shared experiences foster exploration.

3.3 Task: Merging paths

Here we observed that experience sharing improved performance only under the dynamic and small-world topologies. The other ones either find the global optimum quickly (if an agent discovers it by chance before being drawn to the local optimum by others, as happened with Ape-X) or miss it entirely. We can see this maintenance of exploration in the lower conformity of the dynamic topology (C_t remains around 0.6 for the dynamic, around 0.4 for the no-sharing setting and around 0.8 for the others). Finally, compared to the single-path task, we observe that methods rank in the same order in terms of Average Diversity $\bar{D}_t$ (see Figure 12 in Appendix A.4.2). The most notable difference between the two tasks is that diversity, both for individual agents and the group, persists longer in this task, due to the fact that finding the optimal solution requires longer exploration.

3.4 TASK: BEST-OF-TEN PATH

In this task we observed that groups with lower connectivity explore more efficiently, with dynamic performing best ($S^{\mathcal{G}} = 0.4$). No-sharing failed due to slow convergence while the other methods primarily converged to local optima (see Appendix A.4.2 for reward plots). Observing the metrics in Figure 6 we see that this task differs significantly from the previous two: a) volatility (Figure 6, Top right) is highest for dynamic and ring. As these have the highest success rate, we conclude that low connectivity is crucial for exploration of large spaces, as it ensures that the group does not converge prematurely; b) in the no-sharing setting, group diversity (Figure 6, Bottom right) is lowest. This comes in stark contrast to the two previous tasks, where this setting exhibited high diversity. Thus, similarly to the dynamic setting in the single path task, experience sharing fosters exploration.

4 DISCUSSION AND FUTURE WORK

From tool use and language to music and mathematics, human innovation is characterized by cumulative solutions. Human studies have shown that different communication networks are best suited to solve different innovation tasks. However, most decentralized RL approaches assume static, often fully-connected networks. Here we test the hypothesis that the topology of experience sharing in a group of deep RL agents can shape its performance using our proposed algorithm SAPIENS. In line with human experiments, SAPIENS experiments show that dynamic topologies of experience sharing are best suited to solve complex innovation tasks. Our work extends upon the classical decentralized RL paradigm by explicitly measuring diversity, which has been associated with improved exploration (Horgan et al., 2018; Christianos et al., 2020), and demonstrating that certain multi-agent topologies are good at maintaining it, without requiring agents with different hyper-parameters.

Our study shows that both multi-agent network topology and task structure affect the performance of SAPIENS. Based on our experimental results, we can provide general recommendations on which topology to use for which task class. The single-path task is an instance of a class of tasks with no strong local optima (similarly to long-horizon tasks (Gupta et al., 2019)). In this case, our experimental results show no benefit of experience sharing. The merging-path task exhibits strong local optima that have to be explored up to a certain point in order to discover the global optimum (in the spirit of hard exploration tasks as in (Baker et al., 2022; Ecoffet et al., 2021)). Our results show that topologies with low initial connectivity (such as no-sharing, small world and dynamic) performs best here by improving the exploration of different innovation paths. The dynamic topology shows the highest performance, allowing different groups to reach the merging innovation level in non-optimal paths before sharing their experience during visits to other groups to find the optimal one. Finally, the best-of-ten task is an instance of a class of tasks with a large search space, many local optima and a few global ones (in the spirit of combinatorial optimization tasks as in (Mazyavkina et al., 2021)). In this case, our results show that the dynamic topology performs best, allowing different groups to first explore different paths, then spread the optimal solution to other groups once discovered.

Limitations and future works We currently limit our experiments to symbolic innovation tasks implemented in Wordcraft. We believe that scaling up our study by applying SAPIENS in environments commonly employed by the multi-agent reinforcement learning community to study innovation (Leibo et al., 2019) is an important next step. We therefore provide results on a deceptive game task implemented in a grid world in Appendix ¡REF¿. While the present implementation considers groups of DQN agents, SAPIENS is in theory agnostic to the choice of off-policy algorithm. In future studies we plan to investigate SAPIENS networks with different types of RL agents. For instance, we could identify what network characteristics (e.g. centrality, brokering, bridging, etc) is best suited for which learning agents across different tasks. We also envision a meta-learning version of SAPIENS where an outer-loop optimizes the social network structure.

Societal implications. Such future studies can have wide societal implications, paving the way toward predicting optimal social network structures for human-AI, AI-AI human-human cooperation networks. In an era of pending climate catastrophes and global pandemics, better understanding collaboration networks appears relevant. We believe SAPIENS can shed light on how to solve a wide range of innovation tasks and lead to future tools for fostering human and machine innovation in large organizations. Particular attention will however be required to prevent potential negative impact, where controlling the dynamics of social network topologies could result in unequal access to information and privacy issues.

REFERENCES

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *arXiv preprint arXiv: Arxiv-2206.11795*, 2022.

Philip Bontrager, Ahmed Khalifa, Damien Anderson, Matthew Stephenson, Christoph Salge, and Julian Togelius. Superstition in the network: Deep reinforcement learning plays deceptive games. *CoRR*, abs/1908.04436, 2019. URL http://arxiv.org/abs/1908.04436.

Mauricio Cantor, Michael Chimento, Simeon Q. Smeele, Peng He, Danai Papageorgiou, Lucy M. Aplin, and Damien R. Farine. Social network architecture and the tempo of cumulative cultural evolution. *Proceedings of the Royal Society B: Biological Sciences*, 288(1946):20203107, March 2021. doi: 10.1098/rspb.2020.3107. URL https://royalsocietypublishing.org/doi/10.1098/rspb.2020.3107. Publisher: Royal Society.

Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 10707–10717. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/7967cc8e3ab559e68cc944c44b1cf3e8-Paper.pdf.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? statistical power analysis in deep reinforcement learning experiments. *CoRR*, abs/1806.08295, 2018. URL http://arxiv.org/abs/1806.08295.

Alin Coman, Ida Momennejad, Rae D Drach, and Andra Geana. Mnemonic convergence in social networks: The emergent properties of cognition at a collective level. *Proc. Natl. Acad. Sci. U. S. A.*, 113(29):8171–8176, July 2016.

Maxime Derex and Robert Boyd. Partial connectivity increases cultural accumulation within groups. *Proceedings of the National Academy of Sciences*, 113(11):2982–2987, March 2016. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1518798113. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1518798113.

Robin I. M. Dunbar. How conversations around campfires came to be. *Proceedings of the National Academy of Sciences*, 111(39):14013, September 2014. doi: 10.1073/pnas.1416382111. URL http://www.pnas.org/content/111/39/14013.abstract.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. Technical Report arXiv:1802.01561, arXiv, June 2018. URL http://arxiv.org/abs/1802.01561. arXiv:1802.01561 [cs].

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1025–1037. PMLR, 2019. URL http://proceedings.mlr.press/v100/gupta20a.html.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*, 2021.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed Prioritized Experience Replay. *arXiv:1803.00933 [cs]*, March 2018. URL http://arxiv.org/abs/1803.00933. arXiv: 1803.00933.

Minqi Jiang, Jelena Luketina, Nantas Nardelli, Pasquale Minervini, Philip H. S. Torr, Shimon Whiteson, and Tim Rocktäschel. WordCraft: An Environment for Benchmarking Commonsense Agents. *arXiv:2007.09185 [cs]*, July 2020. URL http://arxiv.org/abs/2007.09185. arXiv: 2007.09185.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

Michelle A. Kline and Robert Boyd. Population size predicts technological complexity in Oceania. *Proceedings of the Royal Society B: Biological Sciences*, 277(1693):2559–2564, August 2010. ISSN 0962-8452, 1471-2954. doi: 10.1098/rspb.2010.0452. URL https://royalsocietypublishing.org/doi/10.1098/rspb.2010.0452.

David Lazer and Allan Friedman. The Network Structure of Exploration and Exploitation. *Administrative Science Quarterly*, 52(4):667–694, December 2007. ISSN 0001-8392, 1930-3815. doi: 10.2189/asqu.52.4.667. URL http://journals.sagepub.com/doi/10.2189/asqu.52.4.667.

Joel Z Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *arXiv preprint arXiv:1903.00742*, 2019.

W. Mason and D. J. Watts. Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3):764–769, January 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1110069108. URL http://www.pnas.org/cgi/doi/10.1073/pnas.1110069108.

Winter A. Mason, Andy Jones, and Robert L. Goldstone. Propagation of innovations in networked groups. *Journal of Experimental Psychology: General*, 137(3):422–433, 2008. ISSN 1939-2222, 0096-3445. doi: 10.1037/a0012798. URL http://doi.apa.org/getdoi.cfm?doi=10.1037/a0012798.

Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 2021.

Andrea Bamberg Migliano and Lucio Vinicius. The origins of human cumulative culture: from the foraging niche to collective intelligence. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 377(1843):20200317, January 2022. doi: 10.1098/rstb.2020.0317. URL https://royalsocietypublishing.org/doi/10.1098/rstb.2020.0317. Publisher: Royal Society.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. *arXiv:1602.01783 [cs]*, June 2016. URL http://arxiv.org/abs/1602.01783. arXiv: 1602.01783.

Ida Momennejad. Collective minds: social network topology shapes collective cognition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 377(1843):20200315, January 2022. doi: 10.1098/rstb.2020.0315. URL https://royalsocietypublishing.org/doi/10.1098/rstb.2020.0315. Publisher: Royal Society.

Ida Momennejad, Ajua Duker, and Alin Coman. Bridge ties bind collective memories. *Nat. Commun.*, 10(1):1578, April 2019.

Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, Shane Legg, Volodymyr Mnih, Koray Kavukcuoglu, and David Silver. Massively Parallel Methods for Deep Reinforcement Learning. Technical Report arXiv:1507.04296, arXiv, July 2015. URL http://arxiv.org/abs/1507.04296. arXiv:1507.04296 [cs].

Simon Schmitt, Matteo Hessel, and Karen Simonyan. Off-Policy Actor-Critic with Shared Experience Replay. Technical Report arXiv:1909.11583, arXiv, November 2019. URL http://arxiv.org/abs/1909.11583. arXiv:1909.11583 [cs, stat].

Ricard V. Solé, Sergi Valverde, Marti Rosas Casals, Stuart A. Kauffman, Doyne Farmer, and Niles Eldredge. The evolutionary ecology of technological innovations. *Complexity*, 18(4):15–27, March 2013. ISSN 10762787. doi: 10.1002/cplx.21436. URL http://doi.wiley.com/10.1002/cplx.21436.

Lucas Oliveira Souza, Gabriel de Oliveira Ramos, and Celia Ghedini Ralha. Experience Sharing Between Cooperative Reinforcement Learning Agents. *arXiv:1911.02191 [cs]*, November 2019. URL http://arxiv.org/abs/1911.02191. arXiv: 1911.02191.

Erik Volz and Lauren Ancel Meyers. Susceptible–infected–recovered epidemics in dynamic contact networks. *Proceedings of the Royal Society B: Biological Sciences*, 274(1628):2925–2934, December 2007. ISSN 0962-8452, 1471-2954. doi: 10.1098/rspb.2007.1159. URL https://royalsocietypublishing.org/doi/10.1098/rspb.2007.1159.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. 393:3, 1998.

Polly W. Wiessner. Embers of society: Firelight talk among the Ju/'hoansi Bushmen. *Proceedings of the National Academy of Sciences*, 111(39):14027, September 2014. doi: 10.1073/pnas.1404212111. URL http://www.pnas.org/content/111/39/14027.abstract.

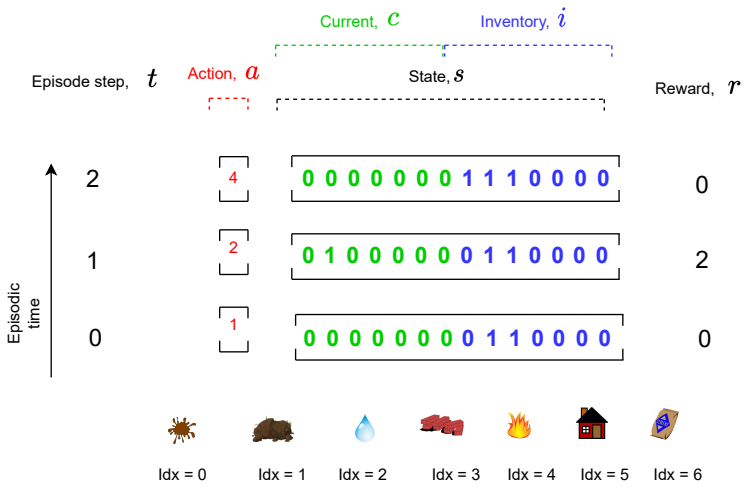

Figure 7: Visualizing actions and states in Wordcraft: we present the first 3 time steps of an episode corresponding to playing the example in Figure 1. This task contains 7 elements, so the action space is a integer with maximum value 7. In the components current c and inventory i, each digit in the vector corresponds to the element with the corresponding index. The initial set includes *Water* and *Earth* (their indexes at $\tau = 0$ in the inventory are non-zero). The agent first picks *Earth* (second index in the action vector). At $t = 1$, *Earth* becomes active in the *Current* vector of the state, the the agent selects *Water* and receives a positive reward. At $t = 2$, *Mud* is created and inserted in the inventory and c is cleared.

A APPENDIX

This supplementary material provides additional methods, results and discussion, as well as implementation details.

- Section A.1 describes in detail the MDP formulation of Wordcraft;
- Section A.2 contains the pseudocde of SAPIENS;
- Section A.3 explains how we model dynamic social network structures and how their performance varies with their hyper-paramaters;
- Section A.4 provides more information about our experimental setup and results (effect of group size, intra-group and inter-group alignment, robustness to learning hyper-parameters and effect of prioritized experience sharing). We also provide tables and figures for all metrics presented in Section 2.4 and reward plots.
- Section A.5 contains simulations with another testbed, the Deceptive Coins game.

A.1 DETAILS OF WORDCRAFT AS A MARKOV DECISION PROCESS

We consider the episodic setting, where the environment resets at the end of each episode and an agent is trained for E_{train} episodes. At each time step t, the agent observes the state s_t and selects an action a_t from a set of possible actions $\mathcal{A}$ according to its policy π^θ, where π^θ is a mapping from states to actions, parameterized by a neural network with weights θ. In return, the agent receives the next state s_{t+1} and a scalar reward r_t. Each DQN agent collects experience tuples of the form $[s_t, a_t, s_{t+1}, r_t]$ in its replay buffer.

Figure 7 offers a visualization of the states and actions encountered during an episode in Wordcraft, where the chosen actions and elements are chosen so as to reproduce the example of Figure 1. In order to solve the innovation task described in Section 2.1 we compute the maximum number of elements a player can craft within horizon T for recipe book $\mathcal{X}_{valid}$ and initial set $\mathcal{X}_0$, which we

denote as $|X|$. We, then, encode each element as an integer in $[0, |X|)$. Thus, the action space is $\mathcal{A} = [0, |X|)$, with action a_t indicating the index of the currently chosen element. The state s_t contains two sets of information: a binary vector of length $|X|$ with non-zero entries for elements already crafted by the agent within the current episode (we refer to this as inventory i) and another binary vector of length $|X|$ where an index is non-zero if it is currently selected by the agent (we refer to this as current c). An agent begins with an inventory having non-zero element only for the initial set $\mathcal{X}_0$ and an all-zero selection. With the first action a_0, the selected item becomes non-zero in the selection. With the second action, a_1, we check if the combination (a_1, c_0) is valid under the recipe book and, if so, return the newly crafted element (corresponding entry in i becomes non-zero) and the reward. This two-step procedure continues until the end of the episode.

A.2 PSEUDOCODE OF SAPIENS

We present the pseudocode of our proposed algorithm SAPIENS in Algorithm 1. SAPIENS works similarly to an off-policy reinforcement learning algorithm, with the difference that, after each episode, an experience sharing phase takes place between agents that belong in the same group.

Algorithm 1 SAPIENS (Structuring multi-Agent toPology for Innovation through ExperieNce Sharing)

Input: $\mathcal{G}, connectivity, R, p_s, LS$
$\mathcal{G}$.initializeGraph(connecticity)
$\mathcal{I}$.initializeAgent() ▷ Initialize agents
for $i \in \mathcal{I}$ **do**
  I.neighbors= I.formNeighborhood($\mathcal{G}$) ▷ Inform agent about its neighbors
  I.env = initEnv(R) ▷ Create agent's own copy of the environment based on the recipe book
end for
while training not done **do**
  for $i \in \mathcal{I}$ **do** ▷ Loop through each agent
    while episode not done **do**
      a = i.policy() ▷ Choose action
      r, s_{new} = env.step(a)
      i.B.insert($[s, r, a, s_{new}]$)
    end while
    ϵ =random()
    if $\epsilon < p_s$ **then** ▷ Share with probability p_s
      for $j \in$ i.neighbors **do**
        j.B.add(i.B.sample(L)) ▷ Sample random set of experiences of length L
      end for
    end if
    i.train() ▷ Train agent
  end for
end while

A.3 ANALYSIS OF DYNAMIC NETWORK TOPOLOGIES

In the main paper we presented results for a single type of dynamic topolgoy. Here we present another type and analyze how they both behave for different values of their hyper-parameters. The two dynamic topologies are:

- Inspired by graphs employed in human laboratory studies (Derex & Boyd, 2016), we designed graphs where the macro structure of the graph is constant but agents can randomly change their position. In particular, we divide a group of agents into sub-groups of two agents and, at the end of each episode, move an agent to another group with a probability p_v for a duration of T_v episodes (for a visualization see Figure 3). To reduce the complexity of the implementation, we assume that only one visit can take place at a time. In the main paper we employ $p_v = 0.01$ and $T_v = 10$ across conditions and present results with different values in Appendix A.3, where we refer to this topology as dynamic-Boyd.
- Human behavioral ecology emphasize the importance of periodic variation in human social networks encountered throughout our evolutionary trajectory Wiessner (2014); Dunbar

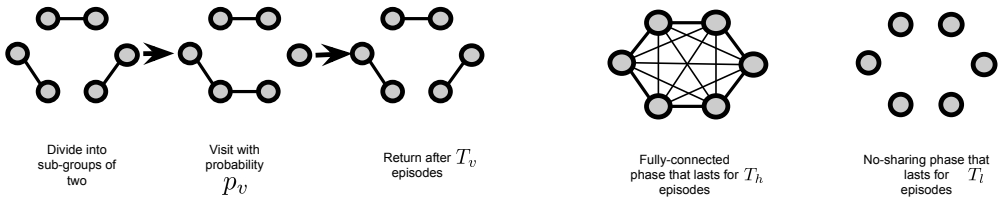

Figure 8: Two types of dynamic topologies: (Left) in the dynamic-Boyd topology the group is divided into sub-groups of two agents and a visit takes place with probability p_v and lasts T_v episodes (Right) In the dynamic-periodic topology the graph oscillates between a phase with a fully-connected topology that lasts for T_h episodes to a phase without sharing that lasts for T_l episodes.

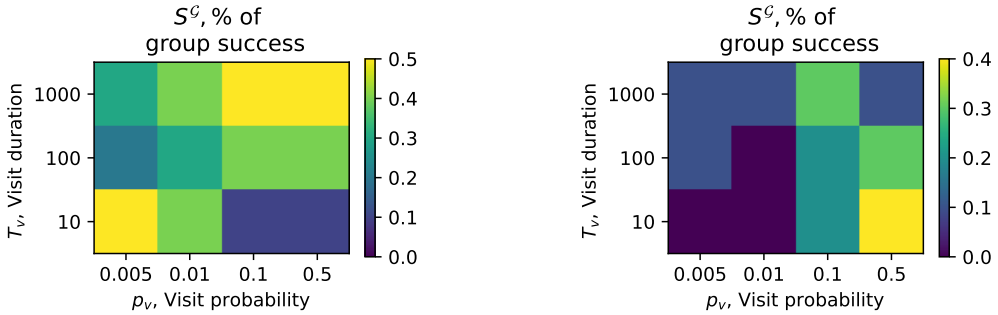

Figure 9: Examining the sensitivity of the dynamic-Boyd topology to its hyper-parameters: $\%$ of group success ($S^{\mathcal{G}}$) for the merging-paths task (left) and the best-of-ten paths task (right).

(2014). Due to ecological constraints human groups oscillate between phases of high and low connectivity: low-connectivity phases arise when individuals need to individually collected resources (e.g. day-time hunting) while high-connectivity phases arise when humans are idle and "forced" to be in proximity with others (e.g. fireside chats). Although these high-connectivity phases do not bare a direct evolutionary advantage, they may have played an important role by creating the conditions for the evolution of human language and culture. Inspired by this hypothesis, we have designed dynamic graphs that oscillate between a fully-connected topology that lasts for T_h episodes and a topology without sharing that lasts for T_l episodes. We present results for various values of T_h and T_d of this topology in Appendix A.3, where we refer to this topology as dynamic-periodic.

In Figure 9, we observe the $\%$ of group success ($S^{\mathcal{G}}$) with the dynamic-Boyd topology for different probabilities of visit (p_v) and visit duration T_v (the sub-group size is 2 in all cases). We note that, due to our implementation choice that a visit can take place only if no other agent is currently on a visit, the visit duration also affects the mixing of the group: longer visits mean that fewer visits will take place in total. **In the merging paths task (left), two hyper-parameter settings have a clear effect: (i) short visits with of high probability lead to bad performance. As such settings lead to a quick mixing of the population, they lead to convergence to the local optimum (ii) long visits with high probability work well. Due to the high visit probability, this setting effectively leads to topology where exactly one agent is always on a long visit. Thus, it ensures that sub-groups stay isolated for at least 1000 episodes, after which inter sub-group sharing needs to takes place to ensure that the sub-groups can progress quickly. In the best-of-ten paths task (right), this structure has a clear optimal hyper-parameterization: short visits with high probability are preferred, which maximizes the mixing of the group and makes early exploration more effective.**

In Figure 10, we observe the $\%$ of group success ($S^{\mathcal{G}}$) of the dynamic-periodic topology for various values of T_h and T_l. **In the merging paths task (left of Figure 10) medium values for the period of both phases works best, while there is some success when the low connectivity phase lasts long ($T_l = 1000$). In the best-of-ten paths task (rightof Figure 10), we observe the same**

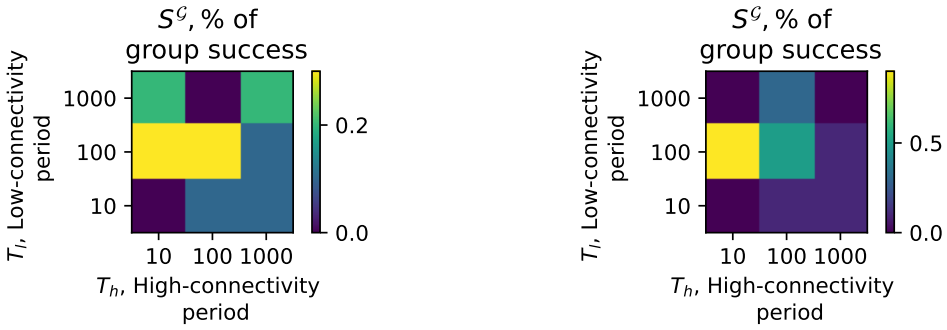

Figure 10: Examining the sensitivity of the dynamic-periodic topology to its hyper-parameters: % of group success ($S^{\mathcal{G}}$) for the merging-paths task (left) and the best-of-ten paths task (right).

medium values for the period of both phases work best: thus both the absolute value and their ratio is important to ensure that exploration is efficient. The optimal configuration is the same between the two tasks ($T_l = 100, T_h = 10$), which is a good indication of the robustness of this structure.

A.4 Empirical results

We employed the same budget across methods but varied it across tasks: the single path run for 5.5 million time steps, the merging paths task for 14 million steps and the best-of-ten paths for 20 million steps.

We presented the major results of our evaluation of SAPIENS in Section 3. We now present additional information regarding the implementation of the different components (Appendix A.4.1), the values of all performance metrics and additional plots for experiments discussed in 3 (Appendix A.4.2), results on intra-group and inter-group alignment (AppendixA.4.3), results for groups of varying sizes (Appendix A.4.5) and results on various dynamic topologies (Appendix A.3)

A.4.1 Implementation details

Implementation of DQN We employ the same hyper-parameter for each DQN across all studied tasks and topologies: discount factor $\gamma = 0.9$, the Adam optimizer with learning rate $\alpha = 0.001$ (Kingma & Ba, 2014; Dunbar, 2014), ϵ-greedy exploration with $\epsilon = 0.01$. We employ a feedforward network with two layers with 64 neurons each. We implemented SAPIENS by extending the DQN implementation in the stable-baselines3 framework [3].

Implementation of A2C We used the stable-baselines3 implementation of A2C [4] and tuned the hyper-parameters: learning rate, number of steps, discount factor, the entropy coefficient and the value function coefficient. This gave us the best-performing values 0.001, 5, 0.99, 0.1 and 0.25, respectively, that we also employed in the other tasks.

Implementation of Ape-X We used the ray implementation of Ape-X DQN [5] and tuned the hyper-parameters: learning rate, discount factor, replay buffer capacity and ϵ-greedy exploration. This gave us the best-performing values in the single path task 0.001, 0.9, 5000 and 0.02, respectively.

Implementation of graphs used as social network structures We construct small-worlds using the Watts–Strogatz model (`watts_strogatz_graph` method of the networkx package [6]). This model first builds a ring lattice where each node has n neighbors and then rewires an edge with

[3] https://stable-baselines3.readthedocs.io/en/master/
[4] https://stable-baselines3.readthedocs.io/en/master/modules/a2c.html
[5] https://docs.ray.io/en/latest/rllib/rllib-algorithms.html
[6] https://networkx.org/

Topology	R^+_∞	R^*_∞	T^+	T^*	$T^>$	$S^\mathcal{G}$	C_{avg}	$\bar{V}_{avg}$
no-sharing	0.9310	1	228333	830000	600000	1	0.7095	0.0383
dynamic	1	1	270000	500000	230000	1	0.8494	0.01383
fully-connected	1	1	310666	362000	51333	1	0.8912	0.052
ring	1	1	235333	305333	70000	1	0.8986	0.0280
small-world	1	1	253333	302666	49333	1	0.912	0.02976

Table 1: Evaluation metrics for the single path task

Topology	R^+_∞	R^*_∞	T^+	T^*	$T^>$	$S^\mathcal{G}$	C_{avg}	$\bar{V}_{avg}$
no sharing	0.6222	0.81	3176666	-	-	0.3	0.4253	0.01265
dynamic	0.7076	0.86	3124000	-	-	0.7	0.6444	0.0064
fully-connected	0.5531	0.768	2370000	-	-	0.2	a0.7456	0.1257
ring	0.6195	0.72	3540000	-	-	0.2	0.7779	0.012
small-world	0.5693	0.662	2800000	-	-	0.15	0.75772	0.0411

Table 2: Evaluation metrics for the merging paths task

probability β. Compared to other techniques used in previous works studying the effect of topology Mason et al. (2008), this way of constructing small-worlds ensures that the average path lengths is short and clustering is high. These two properties are what differentiates small-worlds from random (short average path length and small clustering) and regular (long average path length and high clustering) graphs. We employ $n = 4$ and $\beta = 0.2$ in our experiments, which we empirically found to lead to good values of average path length and clustering.

We have described the generation process of dynamic topologies in Appendix A.3. In the main paper we employ the dynamic-Boyd topology with $T_v = 10$ and $p_v = 0.001$ across tasks.

A.4.2 Overall comparison

Tables 1, 2 and 3 contain the values of all metrics discussed in Section 2.4 for the single path, merging paths and best-of-ten paths, respectively. We denote values computed after convergence of the group with underscore ∞ and values averaged over all training steps with underscore *avg* (note that we use $\bar{}$ over variables to denote averaging over agents in a single training step). Cells with a dash (-) indicate that we could not compute the corresponding metrics because a group failed to find a solution in all trials. We also provide the plots of volatility and average diversity for the merging paths and best-of-10 paths task (that were not included in Figure 6) due to page limit constraints).

Figure 11 presents the reward curves for all methods in the single path, merging paths and best-of-ten paths tasks respectively. Specifically, we plot the maximum reward of the group at training step t (R^+_t).

A.4.3 Measuring inter-group and intra-group alignment

We have so far captures the agreement between agents in a group through the behavioral metric of conformity. Here, we present a mnemonic metric for agreement, which we term alignment. Alignment is a complementary metric to the diversity (D^k_t) and group diversity ($D^\mathcal{G}_t$) metrics, that aims at capturing the effect of experience sharing on the replay buffers in a group. We propose a definition

Topology	R^+_∞	R^*_∞	T^+	T^*	$T^>$	$S^\mathcal{G}$	C_{avg}	V_{avg}
no sharing	0.1703	0.268	-	-	-	0	0.37	0.009
dynamic	0.4084	0.546	4806666	-	-	0.4	0.41	0.03
fully-connected	0.2183	0.238	4680000	5040000	360000	0.1	0.4784	0.0065
ring	0.4522	0.468	3010000	5030000	2440000	0.3	0.4607	0.02308
small-world	0.2921	0.328	2455000	3970000	1515000	0.2	0.4755	0.01418

Table 3: Evaluation metrics for the best-of-ten paths task

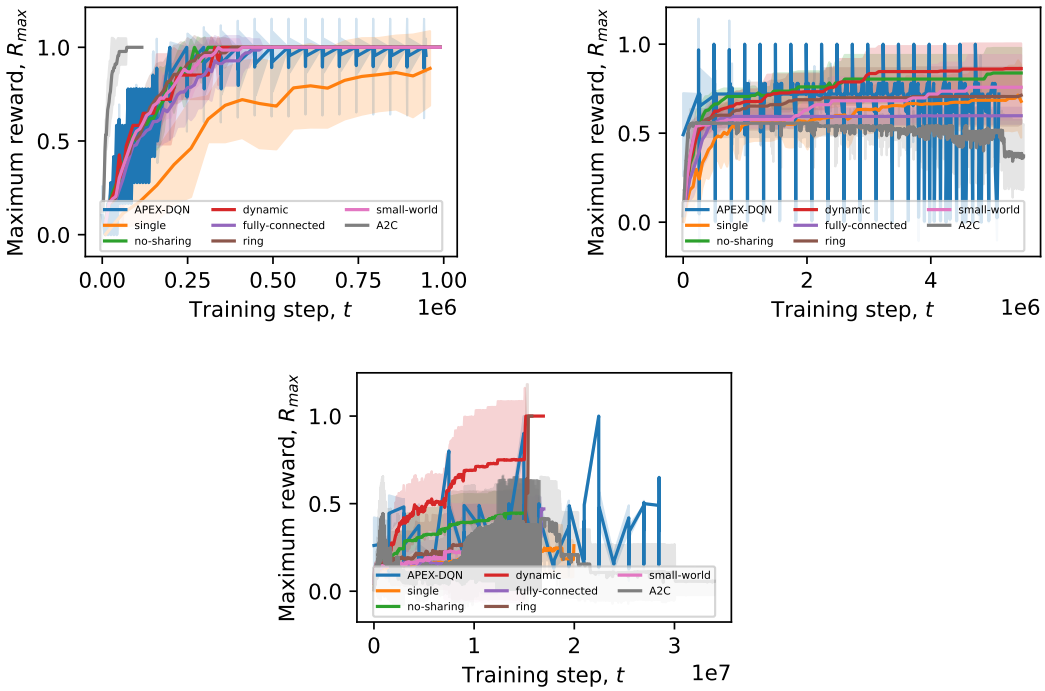

Figure 11: Maximum reward of the group at training step t (R_t^+) in the (left) single path task (middle) merging paths task (right) best-of-ten paths task

of alignment within a single group (intra-group alignment$A_t^{\mathcal{G}}$) and a definition of alignment between two different groups ($A_t^{\mathcal{G}_j,\mathcal{G}_j}$). Such metrics of mnemonic convergence have been linked to social network topology (Coman et al., 2016) and, as we show here, they can prove useful in analyzing groups of reinforcement learning agents.

Specifically: (i) $A_t^{\mathcal{G}}$ is the intra-group alignment. This metric captures the similarity in terms of content between the replay buffers of agents belonging to the same group. To compute this we compute the size of the common subset of experiences for each pair of agents and, then, average over all these pairs, normalizing in [0,1]. (ii) inter-group alignment $A_t^{\mathcal{G}_j,\mathcal{G}_j}$ is a similar notion of alignment but employed between different groups (e.g. how different is a group of fully-connected and a dynamic group of agents in terms of the content of their group replay buffers). To compute it we concatenate all replay buffers of a group into a single one and then compute the size of the common subset of the two replay buffers.

Figure 13 presents intra-group alignment in the three tasks. **We observe that, in all tasks, intra-group alignment increases with connectivity and that it reduces when the agents enter the exploitation phase. Thus, intra-group alignment can prove useful in characterizing the exploration behavior of a group.** In Figure 14, we present the inter-group alignment in the single path, merging paths and best-of-ten paths tasks. We observe that the topologies do not differ significantly in the single path task. **In the merging task, we observe that inter-group alignment is lower during the exploration phase, compared to other tasks, and that the small-world is the slowest to align with all other structures. Perhaps this explains why this topology finds the optimal solution with the least probability: by propagating information quickly, the group early on converges to the local optimum in this task. In the best-of-ten task, the no-sharing setting has the smallest alignment with all other structures. This reinforces our main conclusion in this work: experience sharing affects individuals and different topologies do so in different ways.**

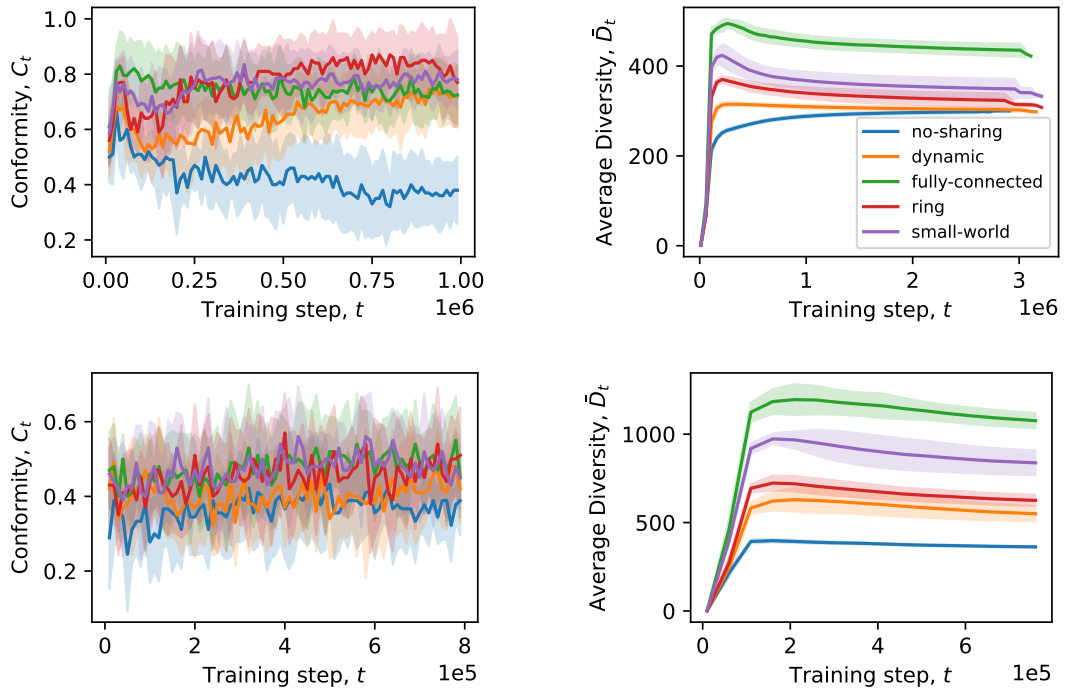

Figure 12: Analyzing group behavior in the merging paths task (top row) and best-of-10 paths task (bottom row). (left) Conformity C_t is a behavioral metric that denotes the percentage of agents in a group that followed the same trajectory in a given evaluation trial (right) Average Diversity $\bar{D}_t$ is a mnemonic metric that denotes the number of unique experiences in the replay buffer of an agent, averaged over all agents.

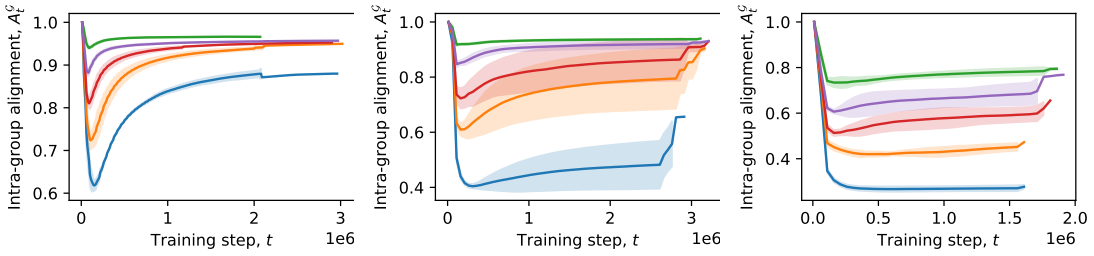

Figure 13: Intra-group alignment $A_t^{\mathcal{G}}$ in the single path task (left), merging paths task (middle) and best-of-ten paths task (right)

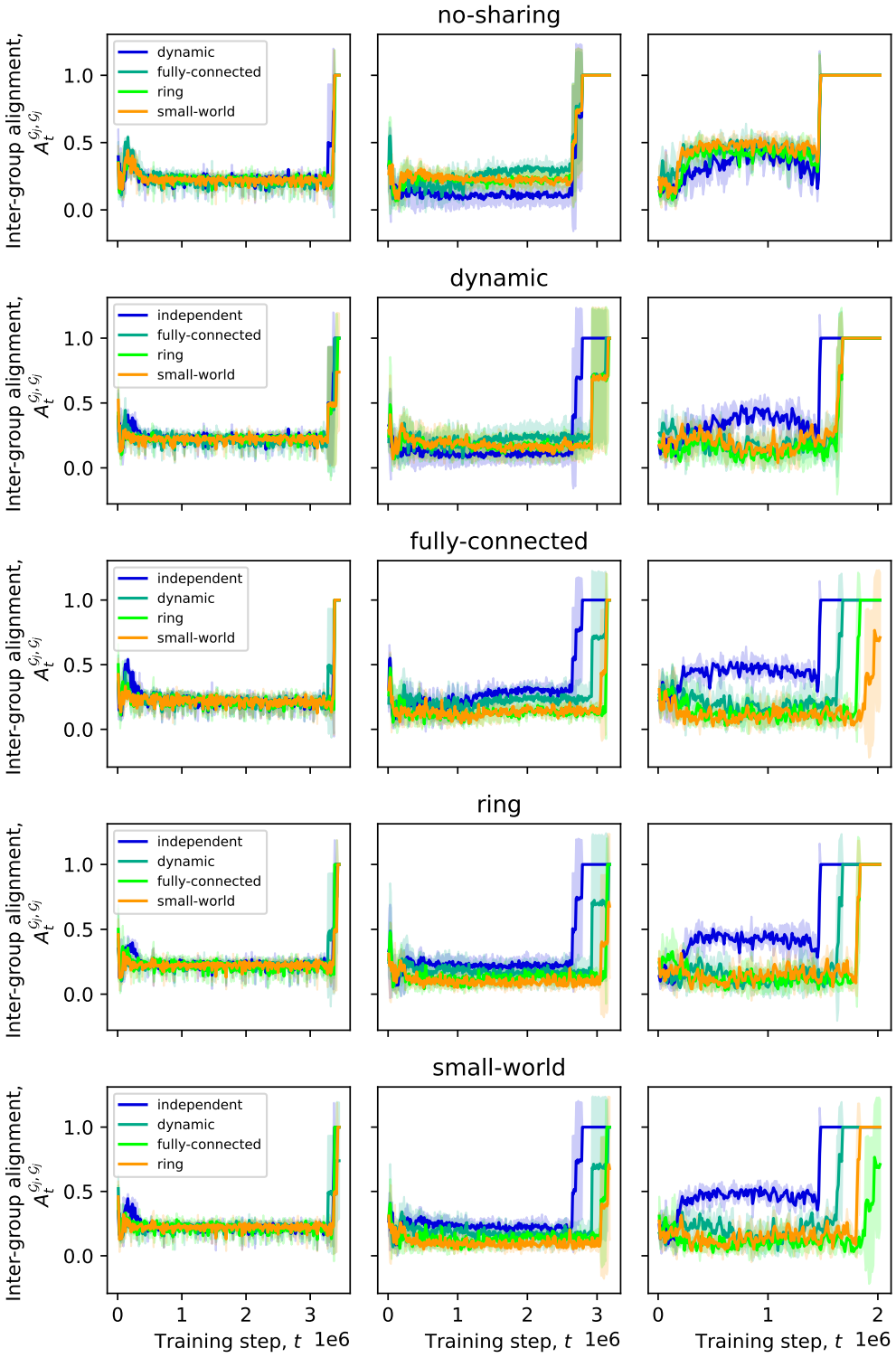

Figure 14: Inter-group alignment $A_t^{\mathcal{G}_j,\mathcal{G}_j}$ in the single path task (left), merging paths task (middle) and best-of-ten paths task (right). In each row we compare one topology with all the rest.

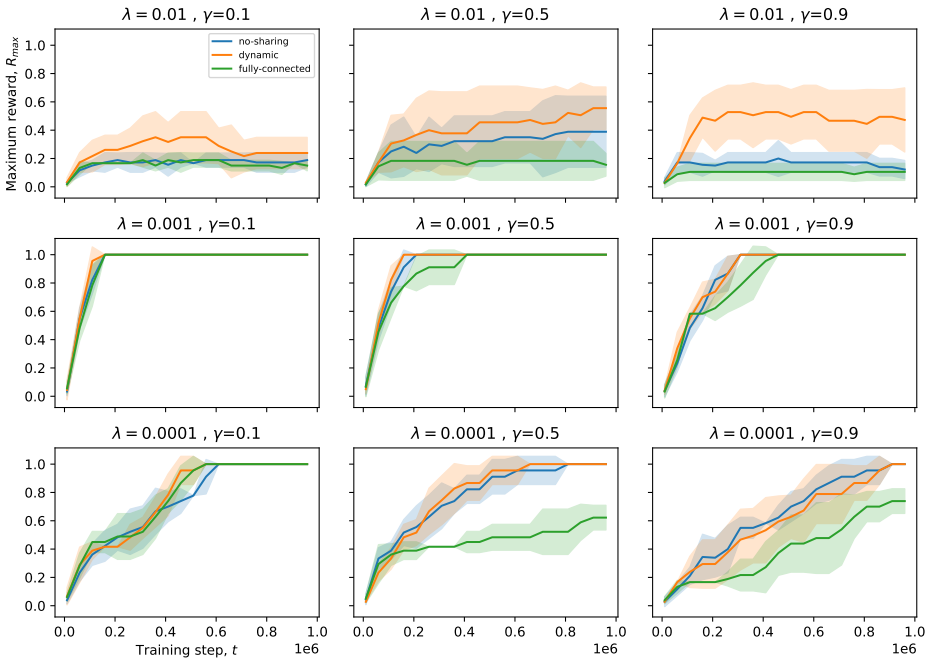

Figure 15: Varying the learning hyper-parameters learning rate (λ) and discount factor (γ) in different social network topologies in the single path task.)

A.4.4 Robustness to learning hyper-parameters

In Figure 15 we present how the performance of SAPIENS varies for different values of the learning hyperparameters learning rate and disocunt factor in the single path task under a fully-connected and a dynamic topolgoy, as well as the *no-sharing* condition. We observe that, although convergence to the optimal solution is not always achieved, the dynamic topology is at least as effective as the others either in terms of convergence rate and/or final performance in all conditions.

A.4.5 Effect of group size

We here examine the effect of the group size for all social network structures in the merging-paths and best-of-ten paths task. To visualize the progression of a group on the paths of the different tasks, we focus on specific elements in the tasks: (i) ($[A_8, B_8, C_2]$ in the merging-paths task. The first two correspond to reaching the end of the paths corresponding to the two local optima. To reduce the computational complexity of experiments, we do not study the last element of the optimal path (C_4), but focus on C_2 instead. This is sufficient to detect whether a group has discovered the optimum path. Here, we observe that the fully-connected topology fails to find the optimal path regardless of its size (with a small success probability for $N = 10$). We observe that the ability of the ring , small-world and dynamict topologies to avoid the local optima improves with the group size (ii) $[B_4, A_2, E_2]$ in the best-of-ten tasks. B_4 is the fourth element on the optimal path (again we do not study the last element to reduce complexity). To avoid cluttering the visualization we only present two of the nine sub-optimal paths. In this task, we again observe that the fully-connected network fails to discover the optimal task. Among all structures and group sizes, the large dynamic network performs best, while the performance of ring and small-world is also best for $N = 50$. We observe that small networks sizes ($N = 2, N = 6$) are slower at exploring (we can see that as they rarely find the second element of the sub-optimal paths, which is required to conclude that path B is the optimal choice).

Overall, **this scaling analysis indicates that increasing the group size in a fully-connected topology will not improve performance, while benefits are expected for low-connectivity structures, particularly for the dynamic topology.** We believe that this observation is crucial. In studies of

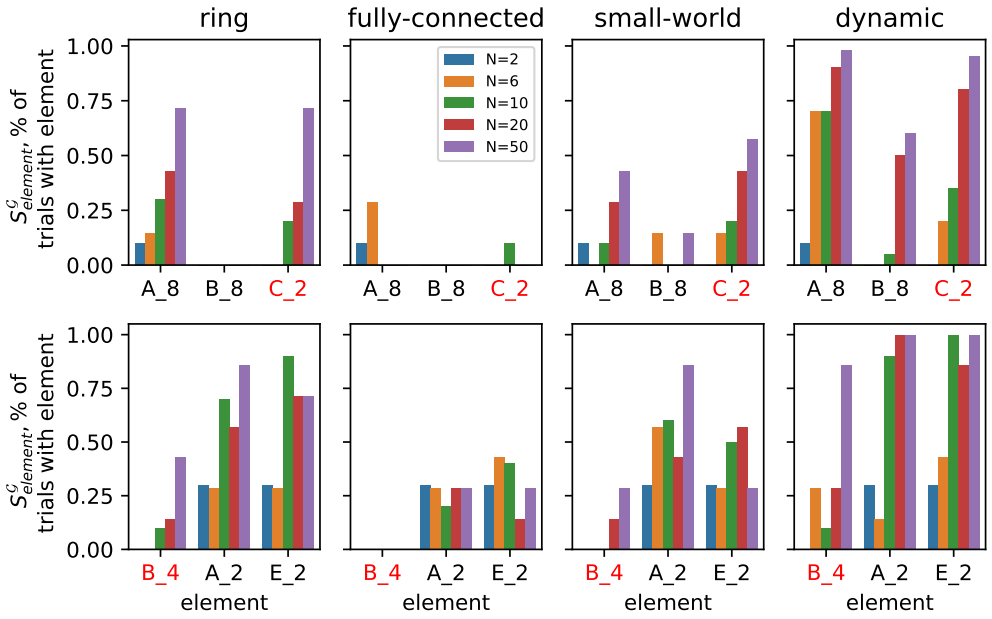

Figure 16: Scaling of different social network structures in the merging paths (top row) and best-of-ten paths tasks (bottom row). We highlight the element belonging to the optimal path in red.

groups of both human and artificial agents, we often encounter the conviction that, larger groups perform better and that size is a more important determinant than connectivity, the latter justifying why connectivity is often ignored Kline & Boyd (2010); Horgan et al. (2018); Mnih et al. (2016); Schmitt et al. (2019); Nair et al. (2015). Our results here point to the contrary.

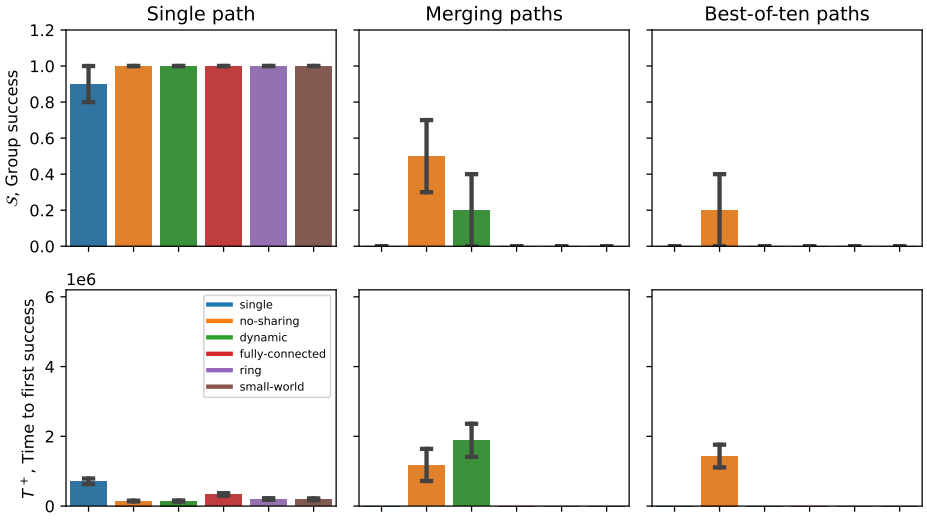

Figure 17: Examining the effect of prioritization in experience sharing. For more details about the setup, we refer the reader to Figure 4

A.4.6 Prioritized experience sharing

We now examine how sharing prioritized experiences instead of randomly sampled ones affects the performance of SAPIENS. In Figure 17 we repeat the same experiment with Figure 4, with the difference that all methods compute priorities, which they employ both for implementing a prioritized

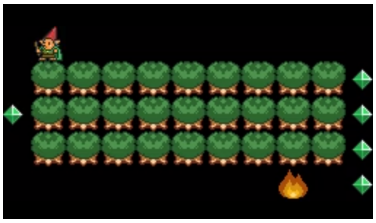

Figure 18: A screenshot of our implementation of the Deceptive Coins task. Collecting diamonds and touching the fire gives a positive reward.

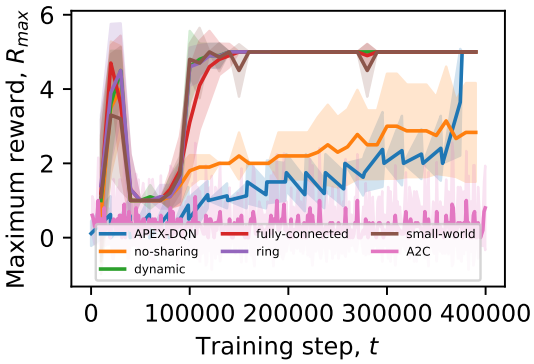

Figure 19: Performance for a group with 6 agents

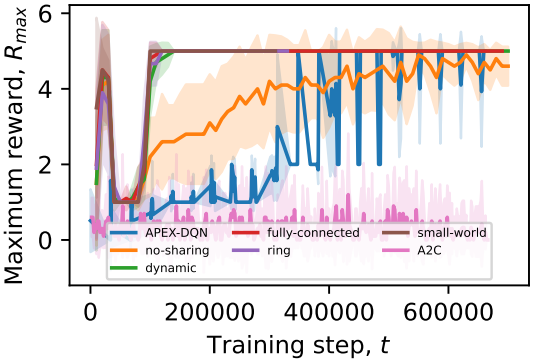

Figure 20: Performance for a group with 10 agents

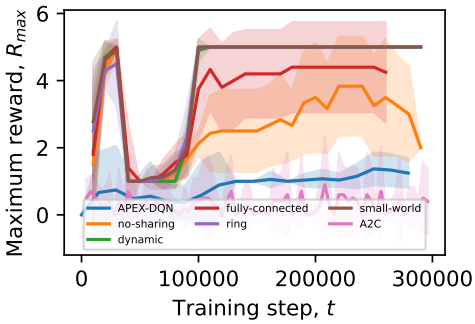

Figure 21: Performance for a group with 20 agents

replay buffer and sharing experiences by sampling them in proportion to their priorities. As we see, using priorities negatively impacts experience sharing, while it helps speed up the performance of the single agent in the single path task. This behavior has been observed in previous works Souza et al. (2019) and can be attributed to the fact that the priorities of the sender do not necessarily agree with the priorities of the receiver and, therefore, destabilize learning.

A.5 Additional test-bed: the deceptive coins game

Deceptive games are grid-world tasks introduced to test the ability of deep RL agents to avoid local optima. (Bontrager et al., 2019). Here, we perform prelimianry experiments with our own JAX-based implementation of one of the games: the first difficulty level of the deceptive coins game (see Figure 18 for an illustration). Here, the agent can navigate in the grid-world during an episode and collect diamonds, which give a unit of reward. The game finishes once the agent reaches the fire, which offers an additional reward, or when a timeout of 14 time steps is reached. There are two possible paths the agent can follow: moving left and reaching the fire will give a reward of two while moving right and reaching the fire will give a reward of five. The second path is more rewarding but is harder to complete because, once an agent discovers the easier-to-find diamond on the left, it is deceived into following the left path. Once an agent commits on a path (reaches the edge of the grid-world) a barrier is raised so that the agent cannot go back within that episode.

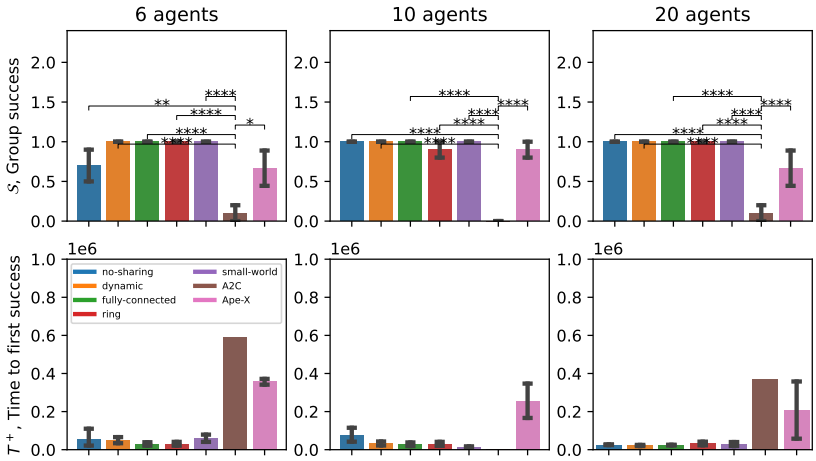

Figure 22: Overall performance comparison for a group with: 6 agents (first column), 10 agents (second column) and 20 agents (third column) task. We present two metrics: group success ($\mathcal{S}$) denotes whether at least one agent in the group found the optimal solution (top row) and T^+, Time to first success, is the number of training time steps required for this event (bottom row). Note that T^+ can be computed only for $\mathcal{S} > 0$ its error bars and significance tests can only be computer for $\mathcal{S} > 1$. We denote statistical significance levels with asterisks.)

We now examine the performance of SAPIENS under different social network structures (fully-connected, small-world, ring, dynamic), as well as the no-sharing, A2C and Ape-X baselines for three group sizes: 6 , 10 and 20 agents. We present the reward plots for the 3 sizes in Figures 19, 20 and 21, respectively, and present an overall comparison in Figure 22 (equivalent to Figure 4 for the Wordcraft tasks).

We observe that all conditions found either the local or the global optimum and that : a) A2C fails for all network sizes. This behavior has been observed in previous works (Bontrager et al., 2019) and can be attributed to the fact that policy-gradient methods are more susceptible to local minima b) no-sharing gets stuck in the local optimum in half of the trials when the group size is small. Increasing the group size increases the probability that at least one agent in the group will escape the local minima by ϵ-greedy exploration c) partially connectd structures find the global minima across network sizes d) fully-connected converges to the local optimum for the large group size, although the global optimum was discovered at the early exploration phase (see Figure 21). Thus, too much experience sharing is harmful e) Ape-X fails with high probability for all network sizes.

In general, our conclusions in this task are consistent with what we observe in Wordcraft, in particular the merging paths task that has a similar deceptive nature.