Leveraging Communication Topologies Between Learning Agents in Deep Reinforcement Learning

Paper # 1296

ABSTRACT

A common technique to improve learning performance in deep reinforcement learning and many other machine learning algorithms is to run multiple learning agents in parallel. A neglected component in the development of these algorithms has been how best to arrange the learning agents involved to improve distributed search. Here we draw upon results from the networked optimization literatures suggesting that arranging learning agents in communication networks other than fully connected topologies (the implicit way agents are commonly arranged in) can improve learning. We explore the relative performance of four popular families of graphs and observe that one such family (Erdos-Renyi random graphs) empirically outperforms the de facto fully-connected communication topology across several DRL benchmark tasks. Additionally, we observe that 1000 learning agents arranged in an Erdos-Renyi graph can perform as well as 3000 agents arranged in the standard fully-connected topology, showing the large learning improvement possible when carefully designing the topology over which agents communicate. We complement these empirical results with a theoretical investigation of why our alternate topologies perform better. Overall, our work suggests that distributed machine learning algorithms could be made more effective if the communication topology between learning agents was optimized[1].

KEYWORDS

Deep reinforcement learning; Evolutionary algorithms; Deep learning

1 INTRODUCTION

Implementations of deep reinforcement learning (DRL) algorithms have become increasingly distributed, running large numbers of parallel sampling and training nodes. For example, AlphaStar runs thousands of parallel instances of Stracraft II on TPU's [29], and OpenAI Five runs on 128,000 CPU cores at the same time [21].

Such distributed algorithms rely on an implicit communication network between the processing units being used in the algorithm. These units pass information such as data, parameters, or rewards between each other, often through a central controller. For example, in the popular A3C [14] reinforcement learning algorithm, multiple 'workers' are spawned with local copies of a global neural network, and they are used to collectively update the global network. These workers can either be viewed as implementing the parallelized form of an algorithm, or they can be seen as a type of multi-agent distributed optimization approach to searching the reward landscape for parameters that maximize performance.

In this work, we take the latter approach of thinking of the 'workers' as separate agents that search a reward landscape more or less efficiently. We adopt such an approach because it allows us to consider improvements studied in the field of multi-agent optimization [9], specifically the literatures of networked optimization (optimization over networks of agents with local rewards) [17–19] and collective intelligence (the study of mechanisms of how agents learn, influence and collaborate with each other) [32, 33].

These two literatures suggest a number of different ways to improve such multi-agent optimization, and, in this work, we choose to focus on one of main ways to do so: optimizing the topology of communication between agents (i.e. the local and global characterization of the connections between agents used to communicate data, parameters, or rewards with).

We focus on communication topology because it has been shown to result in increased exploration, higher reward, and higher diversity of solutions in both simulated high-dimensional optimization problems [12] and human experiments [4], and because, to the best of our knowledge, almost no prior work has investigated how the topology of communication between agents affects learning performance in distributed DRL.

Here, we empirically investigate whether using alternate communication topologies between agents could lead to improving learning performance in the context of DRL. The two topologies that are almost always used in DRL are either a complete (fully-connected) network, in which all processors communicate with each other; or a star network—in which all processors communicate with a single hub server, which is, in effect, a more efficient, centralized implementation of the complete network (e.g., [26]). Our hypothesis is that using other topologies than fully-connected will lead to learning improvements.

Given that network effects are sometimes only significant with large numbers of agents, we choose to build upon one of the DRL algorithms most oriented towards parallelizability and scalability: Evolution Strategies [23, 25, 30], which has recently been shown to scale-up to tens of thousands of agents [24].

We introduce Networked Evolution Strategies (NetES), a networked decentralized variant of ES. NetES, like many DRL algorithms and evolutionary methods, relies on aggregating the rewards from a population of processors that search in parameter space to optimize a single global parameter set. Using NetES, we explore how the communication topology of a population of processors affects learning performance.

Key aspects of our approach, findings, and contributions are as follows:

- We introduce the notion of communication network topologies to the ES paradigm for DRL tasks.

[1] Anonymized supplementary material available at www.bit.ly/2Dsk2OJ

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 2020, Auckland, New Zealand

© 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

https://doi.org/doi

- We perform an ablation study using various baseline controls to make sure that any improvements we see come from using alternate topologies and not other factors.
- We compare the learning performance of the main topological families of communication graphs, and observe that one family (Erdos-Renyi graphs) does best.
- Using an optimized Erdos-Renyi graph, we evaluate NetES on five difficult DRL benchmarks and find large improvements compared to using a fully-connected communication topology. We observe that our 1000-agent Erdos-Renyi graph can compete with 3000 fully-connected agents.
- We derive an upper bound which provides theoretical insights into why alternate topologies might outperform a fully-connected communication topology. We find that our upper bound only depends on the topology of learning agents, and not on the reward function of the reinforcement learning task at hand, which indicates that our results likely will generalize to other learning tasks.

2 PRELIMINARIES

2.1 Evolution Strategies for Deep RL

As discussed earlier, given that network effects are sometimes only significant with large numbers of agents, we choose to build upon one of the DRL algorithms most oriented towards parallelizability and scalability: Evolution Strategies.

We begin with a brief overview of the application of the Evolution Strategies (ES) [25] approach to DRL, following Salimans et al. [24]. Evolution Strategies is a class of techniques to solve optimization problems by utilizing a derivative-free parameter update approach. The algorithm proceeds by selecting a fixed model, initialized with a set of weights θ (whose distribution p_ϕ is parameterized by parameters ϕ), and an objective (reward) function $R(\cdot)$ defined externally by the DRL task being solved. The ES algorithm then maximizes the average objective value $\mathbb{E}_{\theta\sim p_\phi} R(\theta)$, which is optimized with stochastic gradient ascent. The score function estimator for $\nabla_\phi \mathbb{E}_{\theta\sim p_\phi} R(\theta)$ is similar to REINFORCE [31], given by $\nabla_\phi \mathbb{E}_{\theta\sim p_\phi} R(\theta) = \mathbb{E}_{\theta\sim p_\phi}[R(\theta)\nabla_\phi \log p_\phi(\theta)]$.

The update equation used in this algorithm for the parameter θ at any iteration $t+1$, for an appropriately chosen learning rate α and noise standard deviation σ, is a discrete approximation to the gradient:

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\alpha}{N\sigma^2}\sum_{i=1}^{N}\left(R(\theta^{(t)} + \sigma\epsilon_i^{(t)})\cdot\sigma\epsilon_i^{(t)}\right) \tag{1}$$

This update rule is implemented by spawning a collection of N agents at every iteration t, with perturbed versions of $\theta^{(t)}$, i.e. $\{(\theta^{(t)} + \sigma\epsilon_1^{(t)}), ..., (\theta^{(t)} + \sigma\epsilon_N^{(t)})\}$ where $\epsilon \sim \mathcal{N}(0, I)$. The algorithm then calculates $\theta^{(t+1)}$ which is broadcast again to all agents, and the process is repeated.

In summary, a centralized controller holds a global parameter θ, records the perturbed noise $\epsilon_i^{(t)}$ used by *all* agents, collects rewards from *all* agents at the end of an episode, calculates the gradient and obtains a new global parameter θ. Because the controller receives information from all agents, and then broadcasts a new parameter to all other agents, each agent is in effect communicating (through the controller) with all other agents.

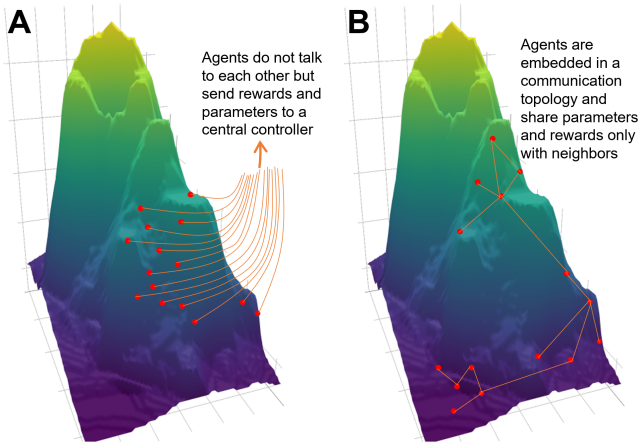

Figure 1: Learning in DRL can be visualized with agents (red dots) searching a reward landscape for the parameter set (location) that leads to the highest reward. A: In most DRL algorithms, including ES, agents are searching the same local area. Because the controller receives information from all agents, and then broadcasts a new parameter to all other agents, agents are, in effect communicating in a fully connected network. B: In NetES, the same number of agents are embedded in a communication topology over which they share data. This leads to a more distributed search where each cluster of agents focuses on a different part of the landscape.

This means that the de facto communication topology used in Evolution Strategies (and all other DRL algorithms that use a central controller) is a fully-connected network. Our hypothesis is that using alternate communication topologies between agents will lead to improved learning performance.

So far, we have assumed that all agents start with the same global parameter $\theta^{(t_0)}$. When each agent i starts with a different parameter $\theta_i^{(t_o)}$, Equation 1 has to be generalized. In the case when all agents start with the same parameter, Equation 1 can be understood as having each agent taking a weighted average of the differences (perturbations) between their last local parameter copy and the perturbed copies of each agent, (the differences being $\sigma\epsilon_i^{(t)} = ((\theta^{(t)} + \sigma\epsilon_i^{(t)}) - \theta^{(t)})$). The weight used in the weighted average is given by the reward at the location of each perturbed copy, $R(\theta^{(t)} + \sigma\epsilon_i^{(t)})$.

When agents start with different parameters, the same weighted average is calculated: because each agent now has different parameters, this difference between agent i and j's parameters is $((\theta_i^{(t)} + \sigma\epsilon_i^{(t)}) - \theta_j^{(t)})$. The weights are still $R(\theta^{(t)} + \sigma\epsilon_i^{(t)}))$. In this notation, Equation 1 is then:

$$\theta_j^{(t+1)} = \theta_j^{(t)} + \frac{\alpha}{N\sigma^2}\sum_{i=1}^{N}\left(R(\theta_i^{(t)} + \sigma\epsilon_i^{(t)})\cdot(\theta_i^{(t)} + \sigma\epsilon_i^{(t)} - \theta_j^{(t)})\right) \tag{2}$$

Type	Task	Fully-connected	Erdos	Improv. %
MuJoCo	Ant-v1	4496	4938	**9.8**
MuJoCo	HalfCheetah-v1	1571	7014	**346.3**
MuJoCo	Hopper-v1	1506	3811	**153.1**
MuJoCo	Humanoid-v1	762	6847	**798.6**
Roboschool	Humanoid-v1	364	429	**17.9**

Table 1: Improvements from Erdos-Renyi networks with 1000 nodes compared to fully-connected networks.

It is straightforward to show that Equation 2 reduces to Equation 1 when all agents start with the same parameter. As we will show, generalizing this standard update rule further to handle alternate topologies will be straightforward.

3 PROBLEM STATEMENT

The task ahead is to take the standard ES algorithm and operate it over new communication topologies, wherein each agent is only allowed to communicate with its neighbors. This would allow us to test our hypothesis that alternate topologies perform better than the de facto fully-connected topology.

An interesting possibility for future work would be to optimize over the space of all possible topologies to find the ones that perform best for our task at hand. In this work, we take as a more tractable starting point a comparison of four popular graph families (including the fully-connected topology).

3.1 NetES : Networked Evolution Strategies

We denote a network topology by $A = \{a_{ij}\}$, where $a_{ij} = 1$ if agents i and j communicate with each other, and equals 0 otherwise. A represents the *adjacency matrix* of connectivity, and fully characterizes the communication topology between agents. In a fully connected network, we have $a_{ij} = 1$ for all i, j.

Using adjacency matrix A, it is straightforward to allow equation 2 to operate over any communication topologies:

$$\theta_j^{(t+1)} = \theta_j^{(t)} + \frac{\alpha}{N\sigma^2} \sum_{i=1}^{N} a_{ij} \cdot \left(R(\theta_i^{(t)} + \sigma\epsilon_i^{(t)}) \cdot (\theta_i^{(t)} + \sigma\epsilon_i^{(t)} - \theta_j^{(t)}) \right) \tag{3}$$

Because equation 3 uses the same weighted average as in ES (equations 1 and 2), when fully-connected networks are used (i.e. $a_{ij} = 1$) and when agents start with the same parameters, equation 3 reduces to 1.

The only other change introduced by NetES is the use of periodic global broadcasts. We implemented parameter broadcast as follows: at every iteration, with a probability p_b, we choose to replace all agents' current parameters with the best agent's performing weights, and then continue training (as per Equation 3) after that. The same broadcast techniques have been used in many other algorithms to balance local vs. global search (e.g. the 'exploit' action in Population-based Training [11] replaces current agent weights with the weights that give the highest rewards).

The full description of the NetES algorithm is shown in Algorithm 1.

Algorithm 1 Networked Evolution Strategies

Input: Learning rate α, noise standard deviation σ, initial policy parameters $\theta_i^{(0)}$ where $i = 1, 2, \ldots, N$ (for N workers), adjacency matrix A, global broadcast probability p_b
Initialize: n workers with known random seeds, initial parameters $\theta_i^{(0)}$
for $t = 0, 1, 2, \ldots$ **do**
  for each worker $i = 1, 2, \ldots, N$ **do**
    Sample $\epsilon_j^{(t)} \sim \mathcal{N}(0, I)$
    Compute returns $R_i = R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)})$
  Sample $\beta^{(t)} \sim \mathcal{U}(0, 1)$
  if $\beta^{(t)} < p_b$ **then**
    Set $\theta_i^{(t+1)} \leftarrow \arg\max_{\theta_i^{(t)}} R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)})$
  else
    for each worker $i = 1, 2, \ldots, n$ **do**
      Set $\theta_i^{(t+1)} \leftarrow \theta_i^{(t)} + \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \left(R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \right)$

In summary, NetES implements three modifications to the ES paradigm: the use of alternate topologies through a_{ij}, the use of different starting parameters, and the use of global broadcast. In the following sections, we will run careful controls during an ablation study to investigate where the improvement in learning we observe come from. Our hypothesis is that they come mainly – or completely – from the use of alternate topologies. As we will show later, they do come from only the use of alternate topologies as shown in see Fig. 2B.

3.2 Updating parameters over alternate topologies

Previous work [4] demonstrates that the exact form of the update rule does not matter as long as the optimization strategy is to find and aggregate the parameters with the highest reward (as opposed to, for example, finding the most common parameters many agents hold). Therefore, although our update rule is a straightforward extension of ES, we expect that our primary insight—that network topology can affect DRL—to still be useful with alternate update rules.

Secondly, although Equation 3 is a biased gradient estimate, at least in the short term, it is unclear whether in practice we achieve a biased or an unbiased gradient estimate, marginalizing over time steps between broadcasts. This is because in the full algorithm (algorithm 1) we implement, we combine this update rule with a periodic parameter broadcast (as is common in distributed learning algorithms - we will address this in detail in a later section), and every broadcast returns the agents to a consensus position.

Future work can better characterize the theoretical properties of NetES and similar networked DRL algorithms using the recently developed tools of calculus on networks (e.g., [1]). Empirically and theoretically, we present results suggesting that the use of alternate topologies can lead to large performance improvements.

3.3 Communication topologies under consideration

Given the update rule as per equation 3, the goal is then to find which topology leads to the highest improvement. Because we are drawing inspiration from the study of collective intelligence and networked optimization, we use topologies that are prevalent in modeling how humans and animals learn collectively:

- **Erdos-Renyi Networks:** Networks where each edge between any two nodes has a fixed independent probability of being present [7], which are among the commonly used benchmark graphs for comparison in social networks [20].
- **Scale-Free Networks:** Scale-free networks, whose degree distribution follows a power law [6], are commonly observed in citation and signaling biological networks [3].
- **Small-World Networks:** Networks where most nodes can be reached through a small number of neighbors, resulting in the famous 'six degrees of separation' [28].
- **Fully-Connected Networks:** Networks where every node is connected to every other node.

We can randomly sample instances of graphs from each family which is parameterized by the number of nodes N, and their degree distribution. Erdos-Renyi networks, for example, are parameterized by their average density p ranging from 0 to 1, where 0 would lead to a completely disconnected graph (no nodes are connected), and 1 would lead back to a fully-connected graph. The lower p is, the sparser a randomly generated network is. Similarly, the degree distribution of scale-free networks is defined by the exponent of the power distribution. Because each graph is generated randomly, two graphs with the same parameters will be different if they have different random seeds, even though, on average, they will have the same average degree (and therefore the same number of links).

3.4 Predicted improved performance of NetES

Through the modifications to ES we have described, we are now able to operate on any communication topology. Due to previous work in networked optimization and collective intelligence which shows that alternate network structures result in better performance, we expect NetES to perform better on DRL tasks when using alternate topologies compared to the de facto fully-connected topology. We also expect to see differences in performance between families of topologies.

4 RELATED WORK

A focus of recent DRL has been the ability to be able to run more and more agents in parallel (i.e. scalability). An early example is the Gorila framework [16] that collects experiences in parallel from many agents. Another is A3C [14] that we discussed earlier. IMPALA [8] is a recent algorithm which solves many tasks with a single parameter set. Population Based Training [11] optimizes both learning weights and hyperparameters. However, in all the approaches described above, agents are organized in an implicit fully-connected centralized topology.

We build on the Evolution Strategies implementation of Salimans et al. [24] which was modified for scalability in DRL. There have been many variants of Evolution Strategies over the years, such as CMA-ES [2] which updates the covariance matrix of the Gaussian distribution, Natural Evolution strategies [30] where the inverse of the Fisher Information Matrix of search distributions is used in the gradient update rule, and, Again, these algorithms implicitly use a fully-connected topology between learning agents.

On the other hand, work in the networked optimization literature has demonstrated that the network structure of communication between nodes significantly affects the convergence rate and accuracy of multi-agent learning [17–19]. However, this work has been focused on solving global objective functions that are the sum (or average) of private, local node-based objective functions - which is not always an appropriate framework for DRL. In the collective intelligence literature, alternate network structures have been shown to result in increased exploration, higher overall maximum reward, and higher diversity of solutions in both simulated high-dimensional optimization [12] and human experiments [4].

The closest work to ours is from the multi-agent reinforcement learning literature. One recent study [13] investigated the effect of communication network topology, but only as an aside, and on very small networks - they also observe improvements when using not fully-connected networks. Another work focuses on the absence of a central controller, but uses on agents solving different tasks at the same time [34].

To the best of our knowledge, no prior work has focused on investigating how the topology of communication between agents affects learning performance in distributed DRL, for large networks and on popular graph families.

5 EXPERIMENTAL PROCEDURE

5.1 Goal of experiments

The main goal of our experiments is to test our hypothesis that using alternate topologies will lead to an improvement in learning performance. Therefore, we want to be able to generate communication topologies from each of the four popular random graph families, wire our agents using this topology and deploy them to solve the DRL task at hand. We also want to run a careful ablation study to understand where the improvements come from.

5.2 Procedure

We evaluate our NetES algorithm on a series of popular benchmark tasks for deep reinforcement learning, selected from two frameworks—the open source Roboschool [22] benchmark, and the MuJoCo framework [27]. The five benchmark tasks we evaluate on are: Humanoid-v1 (Roboschool and Mujoco), HalfCheetah-v1 (MuJoCo), Hopper-v1 (MuJoCo) and Ant-v1 (MuJoCo). Our choice of benchmark tasks is motivated by the difficulty of these walker-based problems.

To maximize reproducibility of our empirical results, we use the standard evaluation metric of collecting the total reward agents obtain during a test-only episode, which we compute periodically during training [5, 15, 24]. Specifically, with a probability of 0.08, we intermittently pause training, take the parameters of the best agent and run this parameter (without added noise perturbation) for 1000 episodes, and take the average total reward over all episodes—as in Salimans et al. [24]. When performance eventually stabilizes

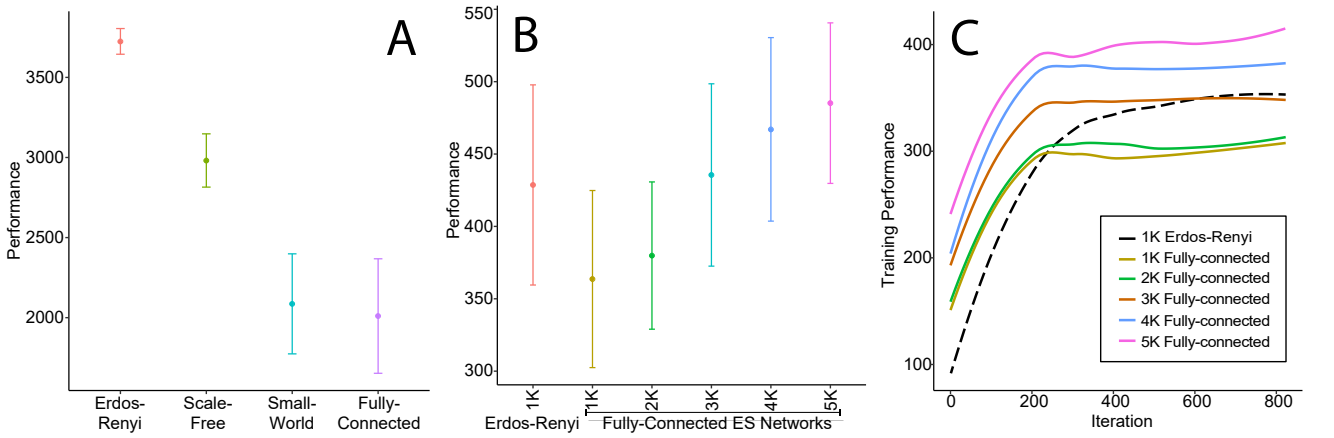

Figure 2: A: Learning performance on all network families: Erdos-Renyi graphs do best, fully-connected graphs do worst (MuJoCo Ant-v1 task with small networks of 100 nodes). B: Evaluation results for Erdos-Renyi graph with 1000 agents compared to fully-connected networks with varying network sizes (RoboSchool Humanoid-v1). C: Comparing Erdos-Renyi graph with 1000 agents to fully-connected networks with varying network sizes on training (not evaluation metric) performance (Roboschool Humanoid-v1). All: Error bars represent 95% confidence intervals.

to a maximum 'flat' line (determined by calculating whether a 50-episode moving average has not changed by more than 5%), we record the maximum of the evaluation performance values for this particular experimental run. As is usual [5], training performance (shown in Fig. 2C) will be slightly lower that the corresponding maximum evaluation performance (shown in Table 1). We observe this standard procedure to be quite robust to noise.

We repeat this evaluation procedure for multiple random instances of the same network topology by varying the random seed of network generation. These different instances share the same average density p (i.e. the same average number of links) and the same number of nodes N. We use a global broadcast probability of 0.8 (a popular hyperparameter value for broadcast in optimization problems). Since each node runs the same number of episode time steps per iteration, different networks with the same p can be fairly compared. For all experiments (all network families and sizes of networks), we use an average network density of 0.5 because it is sparse enough to provide good learning performance, and consistent (not noisy) empirical results.

We then report the average performance over 6 runs with 95% confidence intervals. We share the JSON files that fully describe our experiments and our anonymized code[2].

In addition to using the evaluation procedure of Salimans et al. [24], we also use their exact same neural network architecture: multilayer perceptrons with two 64-unit hidden layers separated by *tanh* nonlinearities. We also keep all the modifications to the update rule introduced by Salimans et al. to improve performance: (1) training for one complete episode for each iteration; (2) employing antithetic or mirrored sampling, also known as mirrored sampling [10], where we explore $\epsilon_i^{(t)}, -\epsilon_i^{(t)}$ for every sample $\epsilon_i^{(t)} \sim \mathcal{N}(0, I)$; (3) employing fitness shaping [30] by applying a rank transformation to the returns before computing each parameter update, and (4) weight decay in the parameters for regularization. We also use the exact same hyperparameters as the original OpenAI (fully-connected and centralized) implementation [24], varying only the network topology for our experiments.

[2]JSON expement files and code implementation can be found at www.bit.ly/2Dsk2OJ.

6 RESULTS

6.1 Empirical performance of network families

We first use one benchmark task (MuJoCo Ant-v1, because it runs fastest) and networks of 100 agents to evaluate NetES on each of the 4 families of communication topology: Erdos-Renyi, scale-free, small-world and the standard fully-connected network. As seen in Fig. 2A, Erdos-Renyi strongly outperforms the other topologies.

Our hypothesis is that using alternate topologies (other than the de facto fully-connected topologies) can lead to strong improvements in learning performance. We therefore decide to focus on Erdos-Renyi graphs for all other results going forward - this choice is supported by our theoretical results which indicate that Erdos-Renyi would do better on any task.

If Erdos-Renyi continues to outperform fully-connected topologies on various tasks and with larger networks, our hypothesis will be confirmed - as long as they are also in agreement with our ablation studies. We leave to future work the full characterization of the performance of other network topologies.

6.2 Empirical performance on all benchmarks

Using Erdos-Renyi networks, we run larger networks of 1000 agents on all 5 benchmark results. As can be seen in Table 1, our Erdos-Renyi networks outperform fully-connected networks on all benchmark tasks, resulting in improvements ranging from 9.8% on MuJoCo Ant-v1 to 798% on MuJoCo Humanoid-v1. All results are statistically significant (based on 95% confidence intervals).

We note that the difference in performance between Erdos-Renyi and fully-connected networks is higher for smaller networks (Fig. 2A and Fig. 3B) compared to larger networks (Table 1) for the same

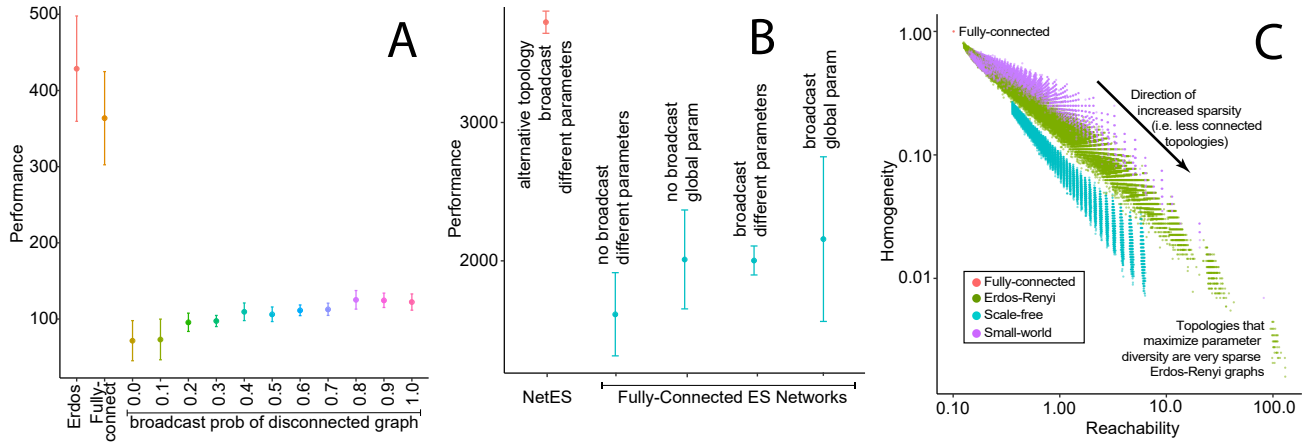

Figure 3: A: Agents with any amount of periodic broadcasting do not learn (RoboSchool Humanoid-v1 with 1000 agents). B: None of the control baselines with fully-connected networks learn, showing that the use of alternate topologies is what leads to learning (MuJoCo Ant-v1 with 100 agents). C: We generate instances of random networks from our four families of networks, and observe that sparser Erdos-Renyi graphs maximize the diversity of parameter updates.

benchmark, and we observe this behavior across different benchmarks. We believe that this is because NetES is able to achieve higher performance with fewer agents due to its efficiency of exploration, as supported in our empirical and theoretical results below.

6.3 Varying network sizes

So far, we have compared alternate network topologies with fully-connected networks containing the same number of agents. In this section, we investigate whether organizing the communication topology using Erdos-Renyi networks can outperform larger fully-connected networks. We choose one of the benchmarks that had a small difference between the two algorithms at 1000 agents, Roboschool Humanoid-v1. As shown in Fig. 2B and the training curves (which display the training performance, not the evaluation metric results which would be higher as discussed earlier) in Fig. 2C, an Erdos-Renyi network with 1000 agents provides comparable performance to 3000 agents arranged in a fully-connected network.

6.4 Ablation Study

To ensure that none of the modifications we implemented in the ES algorithm are causing improvements in performance, instead of just the use of alternate network topologies, we run control experiments on each modification: 1) the use of broadcast, 2) the fact that each agent/node has a different parameter set. We test all combinations.

6.4.1 Broadcast effect. We want to make sure that broadcast (over different probabilities ranging from 0.0 to 1.0) does not explain away our performance improvements. We compare 'disconnected' networks, where agents can only learn from their own parameter update and from broadcasting (they do not see the rewards and parameters of any other agents each step as in NetES). We compare them to Erdos-Renyi networks and fully-connected networks of 1000 agents on the Roboschool Humanoid-v1 task. As can be seen in Fig. 3A practically no learning happens with **just** broadcast and no network. These experiments show that broadcast does not explain away the performance improvement we observe when using NetES.

6.4.2 Global versus individual parameters. The other change we introduce in NetES is to have each agent hold their own parameter value $\theta_i^{(t)}$ instead of a global (noised) parameter $\theta^{(t)}$. We therefore investigate the performance of the following 4 control baselines: fully-connected ES with 100 agent running: (1) same global parameter, no broadcast; (2) same global parameter, with broadcast; (3) different parameters, with broadcast; (4) different parameters, no broadcast; compared to NetES running an Erdos-Renyi network. For this experiment we use MuJoCo Ant-v1. As shown in Fig. 3B, NetES does better than all 4 other control baselines, showing that the improvements of NetES come from using alternate topologies and not from having different local parameters for each agent.

7 THEORETICAL INSIGHTS

In this section, we present theoretical insights into why alternate topologies can outperform fully-connected topologies, and why Erdos-Renyi networks also outperform the other two network families we have tested. A motivating factor for introducing alternate connectivity is to search the parameter space more effectively, a common motivation in DRL and optimization in general. One possible heuristic for measuring the capacity to explore the parameter space is the diversity of parameter updates during each iteration, which can be measured by the variance of parameter updates:

Theorem 7.1. *In a NetES update iteration t for a system with N agents with parameters $\Theta = \{\theta_1^{(t)}, ..., \theta_N^{(t)}\}$, agent communication matrix $A = \{a_{ij}\}$, agent-wise perturbations $\mathcal{E} = \{\epsilon_1^{(t)}, ..., \epsilon_N^{(t)}\}$, and parameter update $u_i^{(t)} = \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \left(R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}\right) \cdot ((\theta_j^{(t)} +$*

$\sigma\epsilon_j^{(t)}) - (\theta_i^{(t)}))\big)$ *as per Equation 3, the following relation holds:*

$$\mathrm{Var}_i[u_i^{(t)}] \leq \frac{\max^2 R(\cdot)}{N\sigma^4}\Big\{\Big(\frac{\|A^2\|_F}{(\min_l |A_l|)^2}\Big)\cdot f(\Theta,\mathcal{E}) - \Big(\frac{\min_l |A_l|}{\max_l |A_l|}\Big)^2 \cdot \frac{\sigma^2}{N}\Big(\sum_{i,j}\epsilon_i^{(t)}\epsilon_j^{(t)}\Big)\Big\} \quad (4)$$

Here, $|A_l| = \sum_j a_{jl}$, *and* $f(\Theta,\mathcal{E}) = \sqrt{\big(\sum_{j,k,m}\big((\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_m^{(t)})\cdot(\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_m^{(t)})\big)^2\big)}$.

The proof for Theorem 8.1 is provided in the supplementary material.

In this work, our hypothesis is to test if some networks *could* do better than the de facto fully-connected topologies used in state-of-the-art algorithms. We leave to future work the important question of optimizing the network topology for maximum performance. Doing so would require a lower bound, as it would provide us the *worst-case* performance of a topology.

Instead, in this section, we are interested in providing insights into why some networks *could* do better than others, which can be understood through our upper-bound, as it allows us to understand the *capacity* for parameter exploration possible by a network topology.

By Theorem 8.1, we see that the diversity of exploration in the parameter updates across agents is likely affected by two quantities that involve the connectivity matrix A: the first being the term $(\|A^2\|_F/(\min_l |A_l|))^2$ (henceforth referred to as the *reachability* of the network), which according to our bound we want to maximize, and the second being $(\min_l |A_l|/\max_l |A_l|)^2$ (henceforth referred to as the *homogeneity* of the network), which according to our bound we want to be as small as possible in order to maximize the diversity of parameter updates across agents. Reachability and homogeneity are not independent, and are statistics of the degree distribution of a graph. It is interesting to note that the upper bound *does not depend on the reward landscape* $R(\cdot)$ *of the task at hand*, indicating that our theoretical insights should be independent of the learning task.

Reachability is the squared ratio of the total number of paths of length 2 in A to the minimum number of links of all nodes of A. Homogeneity is the squared ratio of the minimum to maximum connectivity of all nodes of A: the higher this value, the more homogeneously connected the graph is.

Using the above definitions for reachability and homogeneity, we generate random instances of each network family, and plot them in Fig. 3C. Two main observations can be made from this simulation.

- Erdos-Renyi networks maximize reachability and minimize homogeneity, which means that they likely maximize the diversity of parameter exploration.
- Fully-connected networks are the single worst network in terms of exploration diversity (they minimize reachability and maximize homogeneity, the opposite of what would be required for maximizing parameter exploration according to the suggestion of our bound).

These theoretical results agree with our empirical results: Erdos-Renyi networks perform best, followed by scale-free networks, while fully-connected networks do worse.

It is also important to note that the quantity in Theorem 8.1 is not the **variance of the value function gradient**, which is typically minimized in reinforcement learning. It is instead the **variance in the positions in parameter space** of the agents after a step of our algorithm. This quantity is more productively conceptualized as akin to a radius of exploration for a distributed search procedure rather than in its relationship to the variance of the gradient. The challenge is then to maximize the search radius of positions in parameter space to find high-performing parameters. As far as the side effects this might have, given the common wisdom that increasing the variance of the value gradient in single-agent reinforcement learning can slow convergence, it is worth noting that noise (i.e. variance) is often critical for escaping local minima in other algorithms, e.g. via stochasticity in SGD.

7.1 Sparsity in Erdos-Renyi Networks

We can approximate reachability and homogeneity for Erdos-Renyi networks as a function of their density (a derivation can be found in the supplementary):

Lemma 7.2. *For an Erdos-Renyi graph* $\mathcal{G}$ *with* N *vertices, density* p *and adjacency matrix* A*, the following approximation can be made on its Reachability* $\rho(\mathcal{G})$*.*

$$\rho(\mathcal{G}) \approx (pN)^{-1/2}.$$

Similarly, its homogeneity $\gamma(\mathcal{G})$ *can be approximated as follows.*

$$\gamma(\mathcal{G}) \approx 1 - 8\sqrt{(1-p)/(Np)}.$$

As can be interpreted from these approximations, the sparser an Erdos-Renyi network is (i.e. the lower is p), the larger is the reachability and the lower is the homogeneity. The approximations and the actual reachability and homogeneity (computed directly from the graph adjacency matrix) are plotted in Fig. 4.

In addition to providing insights as to why some families of network topologies do better than others (as shown in Fig. 3C), Theorem 8.1 also predicts that as Erdos-Renyi networks become sparser (less dense) – because their reachability increases and their homogeneity decreases – the diversity of parameter updates during each iteration would increase leading to more effective parameter search, and therefore increased performance.

By running a last number of experiments where we vary the density of Erdos-Renyi networks (keeping the number of agents at 1000) and use these topologies on the RoboSchool Humanoid-v1 DRL benchmark, we can test if sparser networks actually perform better. As can be seen in Figure 5, when the density of Erdos-Renyi networks decreases, learning performance increases significantly.

8 CONCLUSION

In our work, we extended ES, a DRL algorithm, to use alternate network topologies and empirically showed that the de facto fully-connected topology performs worse in our experiments. We also performed an ablation study by running controls on all the modifications we made to the ES algorithm, and we showed that the

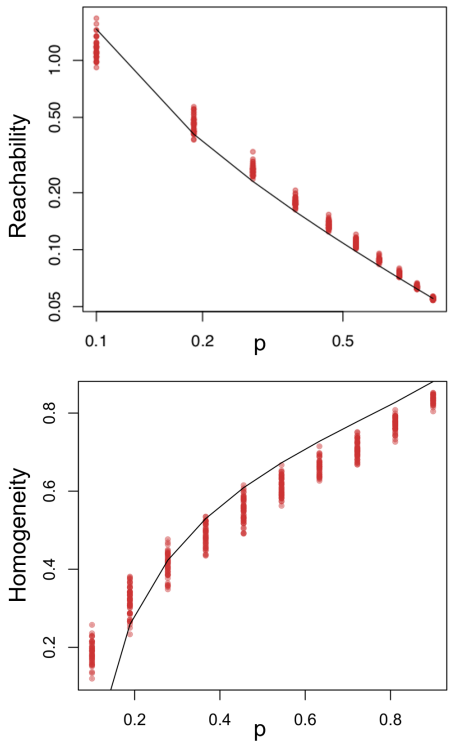

Figure 4: Reachability and homogeneity in the Erdos-Renyi case for different densities p. Points correspond to the real data, while the lines are the approximations.

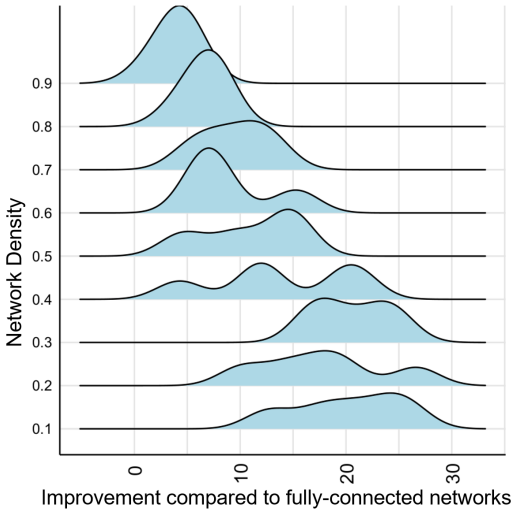

Figure 5: The distributions of reward improvements (compared to the fully-connected topologies) with network density in Erdos-Renyi networks for RoboSchool Humanoid-v1. As predicted by Theorem 8.1, as density decreases, performance increases. Note that a density of 1.0 would result in a fully-connected network.

improvements we observed are not explained away by modifications other than the use of alternate topologies. Finally, we provided theoretical insights into why alternate topologies may be superior, and observed that our theoretical predictions are in line with our empirical results. Future work could explore the use of dynamical topologies where agent connections are continuously rewired to adapt to the local terrain of the research landscape.

Anonymized supplementary material is available at https://gofile.io/?c=6nNZU5.

REFERENCES

[1] Daron Acemoglu, Munther A Dahleh, Ilan Lobel, and Asuman Ozdaglar. 2011. Bayesian learning in social networks. *The Review of Economic Studies* 78, 4 (2011), 1201–1236.

[2] Anne Auger and Nikolaus Hansen. 2005. A restart CMA evolution strategy with increasing population size. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, Vol. 2. IEEE, 1769–1776.

[3] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[4] Daniel Barkoczi and Mirta Galesic. 2016. Social learning strategies modify the effect of network structure on group performance. *Nature communications* 7 (2016).

[5] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47 (2013), 253–279.

[6] Krzysztof Choromański, Michał Matuszak, and Jacek Miekisz. 2013. Scale-free graph with preferential attachment and evolving internal vertex structure. *Journal of Statistical Physics* 151, 6 (2013), 1175–1183.

[7] P ERDdS and A R&WI. 1959. On random graphs I. *Publ. Math. Debrecen* 6 (1959), 290–297.

[8] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. 2018. IMPALA: Scalable distributed Deep-RL with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561* (2018).

[9] Jacques Ferber and Gerhard Weiss. 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Vol. 1. Addison-Wesley Reading.

[10] John Geweke. 1988. Antithetic acceleration of Monte Carlo integration in Bayesian inference. *Journal of Econometrics* 38, 1-2 (1988), 73–89.

[11] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. 2017. Population Based Training of Neural Networks. *arXiv preprint arXiv:1711.09846* (2017).

[12] David Lazer and Allan Friedman. 2007. The network structure of exploration and exploitation. *Administrative Science Quarterly* 52, 4 (2007), 667–694.

[13] Sergio Valcarcel Macua, Aleksi Tukiainen, Daniel García-Ocaña Hernández, David Baldazo, Enrique Munoz de Cote, and Santiago Zazo. 2017. Diff-DAC: Distributed Actor-Critic for Multitask Deep Reinforcement Learning. *arXiv preprint arXiv:1710.10363* (2017).

[14] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. (2016), 1928–1937.

[15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[16] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. 2015. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296* (2015).

[17] Angelia Nedic. 2011. Asynchronous broadcast-based convex optimization over a network. *IEEE Trans. Automat. Control* 56, 6 (2011), 1337–1351.

[18] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. 2017. Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization. *arXiv preprint arXiv:1709.08765* (2017).

[19] Angelia Nedic and Asuman Ozdaglar. 2010. 10 cooperative distributed multi-agent. *Convex Optimization in Signal Processing and Communications* 340 (2010).

[20] Mark Newman. 2010. Networks: An Introduction. (2010).

[21] OpenAI. [n. d.]. OpenAI Five. https://blog.openai.com/openai-five/. ([n. d.]).

[22] OpenAI. 2017. Roboschool. https://github.com/openai/roboschool. (2017). Accessed: 2017-09-30.

[23] Ingo Rechenberg. 1973. Evolution Strategy: Optimization of Technical systems by means of biological evolution. *Fromman-Holzboog, Stuttgart* 104 (1973), 15–16.

[24] Tim Salimans, Jonathan Ho, Xi Chen, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864* (2017).
[25] Hans-Paul Schwefel. 1977. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: mit einer vergleichenden Einführung in die Hill-Climbing-und Zufallsstrategie.* Birkhäuser.
[26] Steven L Scott, Alexander W Blocker, Fernando V Bonassi, Hugh A Chipman, Edward I George, and Robert E McCulloch. 2016. Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management* 11, 2 (2016), 78–88.
[27] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on.* IEEE, 5026–5033.
[28] Jeffrey Travers and Stanley Milgram. 1977. An experimental study of the small world problem. In *Social Networks.* Elsevier, 179–197.
[29] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/. (2019).
[30] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. 2014. Natural evolution strategies. *The Journal of Machine Learning Research* 15, 1 (2014), 949–980.
[31] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning.* Springer, 5–32.
[32] David H Wolpert and Kagan Tumer. 1999. An introduction to collective intelligence. *arXiv preprint cs/9908014* (1999).
[33] Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W Malone. 2010. Evidence for a collective intelligence factor in the performance of human groups. *science* 330, 6004 (2010), 686–688.
[34] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. 2018. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757* (2018).

APPENDIX 1 : DIVERSITY OF PARAMETER UPDATES

Here we provide proofs Theorem 1 from the main paper concerning the diversity of the parameter updates.

Theorem 8.1. *In a multi-agent evolution strategies update iteration t for a system with N agents with parameters* $\Theta = \{\theta_1^{(t)}, ..., \theta_N^{(t)}\}$, *agent communication matrix* $A = \{a_{ij}\}$, *agent-wise perturbations* $\mathcal{E} = \{\epsilon_1^{(t)}, ..., \epsilon_N^{(t)}\}$, *and parameter update* $u_i^{(t)}$ *given by the sparsely-connected update rule:*

$$u_i^{(t)} = \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \left(R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) \cdot ((\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) - (\theta_i^{(t)}))\right)$$

The following relation holds:

$$\mathrm{Var}_i[u_i^{(t)}] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \Big\{ \Big(\frac{\|A^2\|_F}{(\min_l |A_l|)^2} \Big) \cdot f(\Theta, \mathcal{E}) - \Big(\frac{\min_l |A_l|}{\max_l |A_l|} \Big)^2 \cdot g(\mathcal{E}) \Big\} \quad (5)$$

Here, $|A_l| = \sum_j a_{jl}$, $f(\Theta, \mathcal{E}) = \Big(\sum_{j,k,m}^{N,N,N} \big((\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_m^{(t)}) \cdot (\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_m^{(t)})\big)^2 \Big)^{\frac{1}{2}}$, *and* $g(\mathcal{E}) = \frac{\sigma^2}{N} \Big(\sum_{i,j}^{N,N} \epsilon_i^{(t)} \epsilon_j^{(t)} \Big)$.

Proof. From Equation 8.1, the update rule is given by:

$$u_i^{(t)} = \frac{\alpha}{N\sigma^2} \sum_{j=1}^{N} a_{ij} \cdot \left(R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) \cdot ((\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) - (\theta_i^{(t)}))\right) \quad (6)$$

The variance of $u_i^{(t)}$ can be written as:

$$\mathrm{Var}_i[u_i^{(t)}] = \mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})^2] - (\mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})])^2 \quad (7)$$

Expanding $\mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})^2]$:

$$= \frac{1}{N} \sum_{i \in \mathcal{A}} \Big\{ \frac{\gamma}{N\sigma^2} \sum_{j=1} a_{ij} \cdot R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \Big\}^2 \quad (8)$$

Simplifying:

$$= \frac{1}{N\sigma^4} \sum_{i,j,k} \Big(\frac{a_{ij}a_{ik}}{|A_i|^2} R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) R(\theta_k^{(t)} + \sigma\epsilon_k^{(t)}) \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \cdot (\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_i^{(t)}) \Big) \quad (9)$$

Since $R(\cdot) \leq \max R(\cdot)$, therefore:

$$\leq \frac{\max^2 R(\cdot)}{N\sigma^4} \sum_{i,j,k} \frac{a_{ij}a_{ik}}{|A_i|^2} \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \cdot (\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_i^{(t)}) \quad (10)$$

$$\leq \frac{\max^2 R(\cdot)}{N\sigma^4} \sum_{i,j,k} \frac{a_{ij}a_{ik}}{\min_l |A_l|^2} \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \cdot (\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_i^{(t)}) \quad (11)$$

By the Cauchy-SchDwarz Inequality:

$$\mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})^2] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \Big(\sum_{i,j,k} \frac{(a_{ij}a_{ik})^2}{\min_l |A_l|^4} \Big)^{\frac{1}{2}} \cdot \Big(\sum_{i,j,k} \big((\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \cdot (\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_i^{(t)})\big)^2 \Big)^{\frac{1}{2}} \quad (12)$$

Since $a_{ij} \in \{0, 1\} \forall\ (i, j), (a_{ij}a_{ik})^2 = a_{ij}a_{ik}\ \forall (i, j, k)$. Additionally, we know that $a_{ij} = a_{ji}$, since A is symmetric. Therefore, $\sum_i a_{ij}a_{ik} = \sum_i a_{ji}a_{ik} = A^2_{jk}$. Using this:

$$\mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})^2] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \cdot \Big(\frac{|A^2|^{\frac{1}{2}}}{\min_l |A_l|^2} \Big) \cdot \Big(\sum_{i,j,k} \big((\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \cdot (\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_i^{(t)})\big)^2 \Big)^{\frac{1}{2}} \quad (13)$$

Replacing $\Big(\sum_{i,j,k} \big((\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \cdot (\theta_k^{(t)} + \sigma\epsilon_k^{(t)} - \theta_i^{(t)})\big)^2 \Big)^{\frac{1}{2}} = f(\Theta, \mathcal{E})$, where $\Theta = \{\theta_i^{(t)}\}_{i=1}^N, \mathcal{E} = \{\epsilon_i\}_{i=1}^N$ for compactness, we obtain:

$$\mathbb{E}_{i \in \mathcal{A}}[(u_i^{(t)})^2] \leq \frac{\max^2 R(\cdot)}{N\sigma^4} \cdot \Big(\frac{|A^2|^{\frac{1}{2}}}{\min_l |A_l|^2} \Big) \cdot f(\Theta, \mathcal{E}) \quad (14)$$

Similarly, the squared expectation of $(u_i^{(t)})$ over all agents can be given by:

$$(\mathbb{E}_{i \in \mathcal{A}}[u_i^{(t)}])^2 = \Big(\frac{1}{N} \sum_{i \in \mathcal{A}} \Big\{ \frac{\gamma}{N\sigma^2} \sum_{j=1} a_{ij} \cdot R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \Big\} \Big)^2 \quad (15)$$

$$= \frac{1}{N^2\sigma^4} \Big(\sum_{i \in \mathcal{A}} \Big\{ \frac{1}{|A_i|} \sum_{j=1} a_{ij} \cdot R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \Big\} \Big)^2 \quad (16)$$

$$= \frac{1}{N^2\sigma^4} \Big(\sum_{i,j} \Big\{ \frac{a_{ij}}{|A_i|} \cdot R(\theta_j^{(t)} + \sigma\epsilon_j^{(t)}) \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \Big\} \Big)^2 \quad (17)$$

Since $R(\cdot) \geq \min R(\cdot)$, therefore:

$$\geq \frac{\min^2 R(\cdot)}{N^2\sigma^4} \Big(\sum_{i,j} \Big\{ \frac{a_{ij}}{|A_i|} \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \Big\} \Big)^2 \quad (18)$$

$$\geq \frac{\min^2 R(\cdot)}{N^2\sigma^4 \max_l |A_l|^2} \Big(\sum_{i,j} \big\{ a_{ij} \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) \big\} \Big)^2 \quad (19)$$

Since A is symmetric, $\sum_{i,j}^{N,N} a_{ij} \cdot (\theta_j^{(t)} + \sigma\epsilon_j - \theta_i^{(t)}) = \sum_{i,j}^{N,N} a_{ij} \cdot (\theta_i^{(t)} + \sigma\epsilon_i - \theta_j^{(t)})$. Therefore:

$$= \frac{\min^2 R(\cdot)}{N^2\sigma^4 \max_l |A_l|^2} \Big(\sum_{i,j} \frac{1}{2} \big\{ a_{ij} \cdot (\theta_j^{(t)} + \sigma\epsilon_j^{(t)} - \theta_i^{(t)}) + a_{ij} \cdot (\theta_i^{(t)} + \sigma\epsilon_i^{(t)} - \theta_j^{(t)}) \big\} \Big)^2 \quad (20)$$

Therefore,

$$(\mathbb{E}_{i\in\mathcal{A}}[u_i^{(t)}])^2 = \frac{\min^2 R(\cdot)}{N^2\sigma^2 \max_l |A_l|^2}\Big(\sum_{i,j}\frac{1}{2}\{a_{ij}\cdot(\epsilon_j^{(t)}+\epsilon_i^{(t)})\}\Big)^2 \quad (21)$$

Using the symmetry of A, we have that $\sum_{i,j}^{N,N} a_{ij}\epsilon_i = \sum_{i,j}^{N,N} a_{ij}\epsilon_j$. Therefore:

$$= \frac{\min^2 R(\cdot)}{N^2\sigma^2 \max_l |A_l|^2}\Big(\sum_{i,j} a_{ij}\cdot\epsilon_j^{(t)}\Big)^2 \quad (22)$$

$$= \frac{\min^2 R(\cdot)}{N^2\sigma^2 \max_l |A_l|^2}\Big(\sum_{j} |A_j|\cdot\epsilon_j^{(t)}\Big)^2 \quad (23)$$

$$\geq \frac{\min^2 R(\cdot)\min_l |A_l|^2}{N^2\sigma^2 \max_l |A_l|^2}\Big(\sum_{i,j}\epsilon_i^{(t)}\epsilon_j^{(t)}\Big) \quad (24)$$

Combining both terms of the variance expression, and using the normalization of the iteration rewards that ensures $\min R(\cdot) = -\max R(\cdot)$, we can obtain (using $g(\mathcal{E}) = \frac{\sigma^2}{N}\Big(\sum_{i,j}\epsilon_i^{(t)}\epsilon_j^{(t)}\Big)$):

$$\mathrm{Var}_{i\in\mathcal{A}}[u_i^{(t)}] \leq \frac{\max^2 R(\cdot)}{N\sigma^4}\Big\{\Big(\frac{|A^2|^{\frac{1}{2}}}{\min_l |A_l|^2}\Big)\cdot f(\Theta,\mathcal{E}) - \Big(\frac{\min_l |A_l|^2}{\max_l |A_l|^2}\Big)\cdot g(\mathcal{E})\Big\} \quad (25)$$

□

APPENDIX 2 : APPROXIMATING REACHABILITY AND HOMOGENEITY FOR LARGE ERDOS-RENYI GRAPHS

Recall that a Erdos-Renyi graph is constructed in the following way

(1) Take n nodes
(2) For each pair of nodes, link them with probability p

The model is simple, and we can infer the following:

- The average degree of a node is $p(n-1)$
- The distribution of degree for the nodes is the Binomial distribution of $n-1$ events with probability p, $B(n-1,p)$.
- The (average) number of paths of length 2 from one node i to a node $j \neq i$ ($n_{ij}^{(2)}$) can be calculated this way: a path of length two between i and j involves a third node k. Since there are $n-2$ of them, the maximun number of paths between i and j is $n-2$. However, for that path to exists there has to be a link between i and k and k and j, an event with probability p^2. Thus, the average number of paths between i and j is $p^2(n-2)$

Estimating Reachability

We can then estimate Reachability:

$$Reachability = \frac{||A^2||_F}{(min_l|A_l|)^2} = \frac{\sqrt{\sum_{i,j} n_{ij}^{(2)}}}{k_{min}^2}$$

where $k_{min} = (min_l|A_l|)$ is the minimum degree in the network. Given the above calculations we can approximate

$$\sum_{i,j} n_{ij}^{(2)} = \sum_i n_{ii}^{(2)} + \sum_{i\neq j} n_{ij}^{(2)} \approx n\times[p(n-1)] + n(n-1)\times[p^2(n-2)]$$

where the first term is the number of paths of length 2 from i to i summed over all nodes, i.e. the sum of the degrees in the network. The second term is the sum of $p^2(n-2)$ for the terms in which $i \neq j$. For large n we have that

$$\sum_{i,j} n_{ij}^{(2)} \approx p^2n^3$$

and thus,

$$||A^2||_F \approx \sqrt{p^2n^3}. \quad (26)$$

For the denominator k_{min} we could use the distribution of the minimum of the binomial distribution $B(n-1,p)$. However, since it is a complicated calculation we can approximate this way: since the binomial distribution $B(n-1,p)$ looks like a Gaussian, we can say that the minimum of the distribution is closed to the mean minus two times the standard deviation:

$$k_{min} \approx p(n-1) - 2\sqrt{p(n-1)(1-p)} \quad (27)$$

Once again in the case of large n we have

$$k_{min} \approx pn$$

Thus

$$Reachability \approx \frac{\sqrt{p^2n^3}}{[p(n-1)-2\sqrt{p(n-1)(1-p)}]^2} \quad (28)$$

Assuming that n is large, we can approximate

$$Reachability \approx \frac{pn^{3/2}}{p^2n^2} = \frac{1}{pn^{1/2}}$$

Thus the bound decreases with increasing n and p. Note that the density of the Erdos-Renyi graph (the number of links over the number of possible links) is p. And thus for a fixed n more sparse networks $p \simeq 0$ have larger Reachability than more connected networks $p \simeq 1$.

Estimating Homogeneity

The Homogeneity is defined as

$$Homogeneity = \left(\frac{k_{min}}{k_{max}}\right)^2$$

As before we can approximate

$$k_{max} \approx p(n-1) + 2\sqrt{p(n-1)(1-p)}$$

And thus

$$Homogeneity \approx \left(\frac{p(n-1)-2\sqrt{p(n-1)(1-p)}}{p(n-1)+2\sqrt{p(n-1)(1-p)}}\right)^2$$

For large p we can approximate it to be

$$Homogeneity \approx 1 - 8\frac{\sqrt{1-p}}{\sqrt{np}} \quad (29)$$

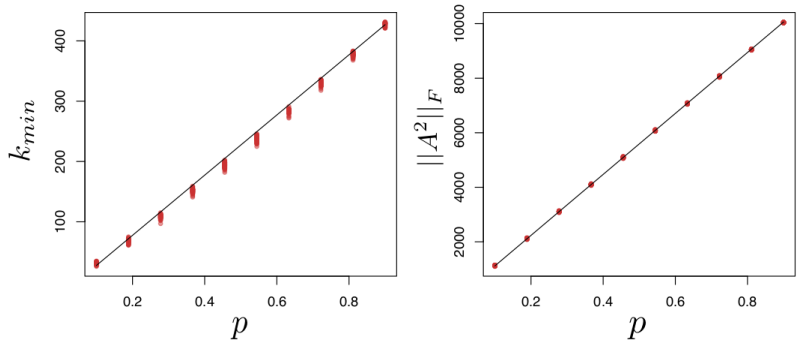

Figure 6: Comparison between the values of k_{min}, $||A^2||_F$, and Reachability as a function of p for different realizations of the Erdos-Renyi model (points) and their approximations given in Equations (27), (26) and (28) respectively (lines).

which shows that for $p \simeq 1$ we have that Homogeneity grows as a function of p. Thus for fixed number of nodes n, increasing p we get larger values of the Homogeneity.