Evaluating Generalization and Transfer Capacity of Multi-Agent Reinforcement Learning Across Variable Number of Agents

Bengisu Güresti, Nazım Kemal Üre

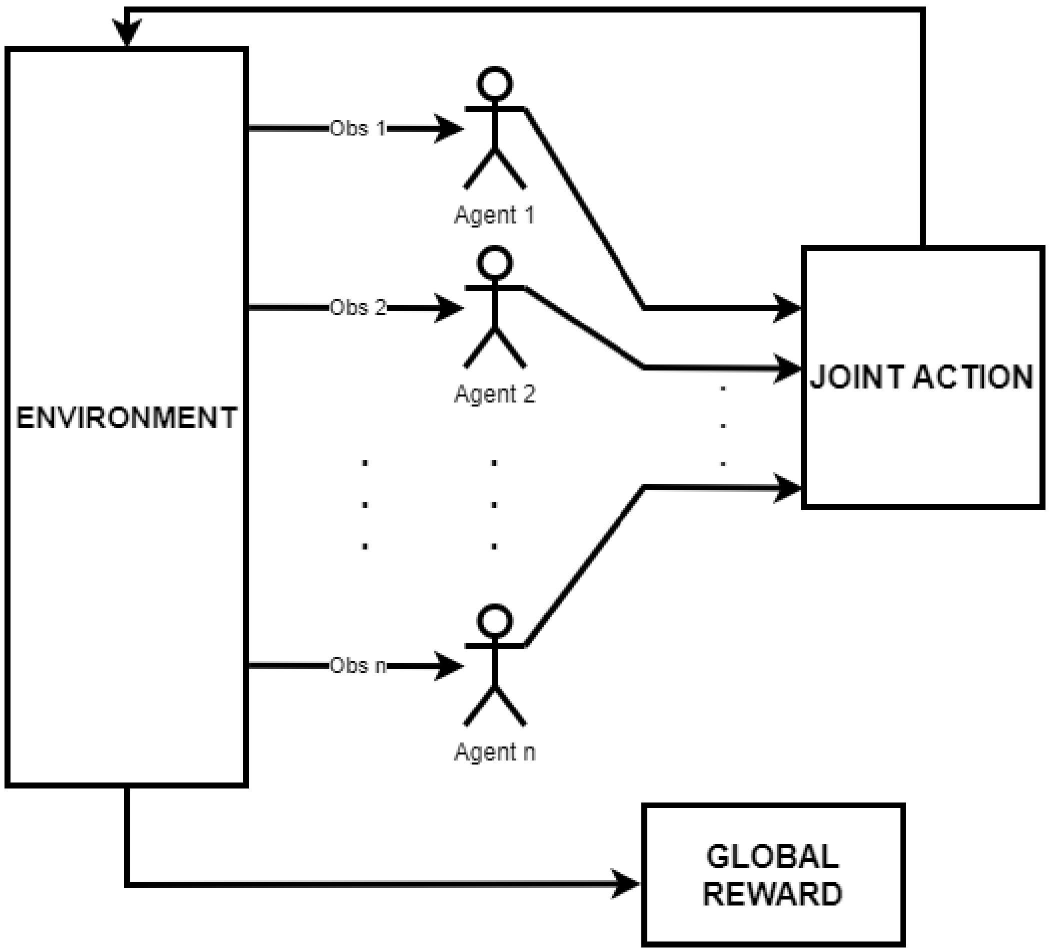
March 23, 2021

Outline

Table of Contents

Overview

Cooperative Multi-Agent Reinforcement Learning

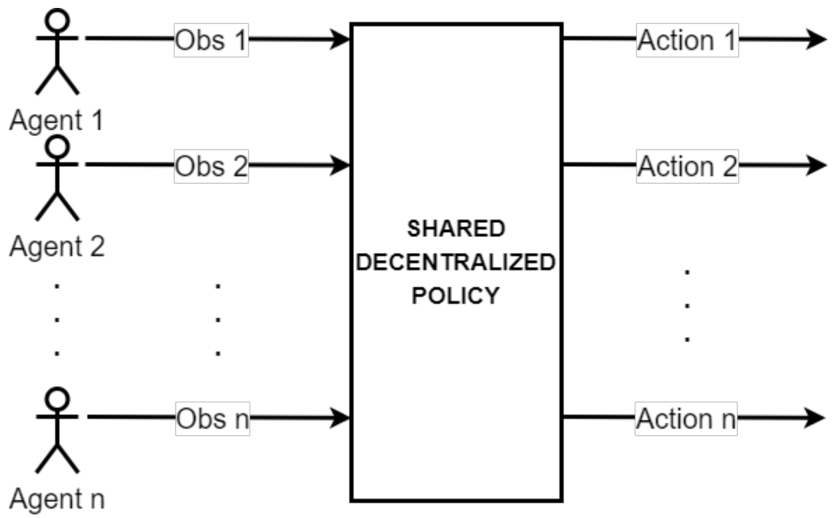


- Common goal
- Global reward
- Local observations

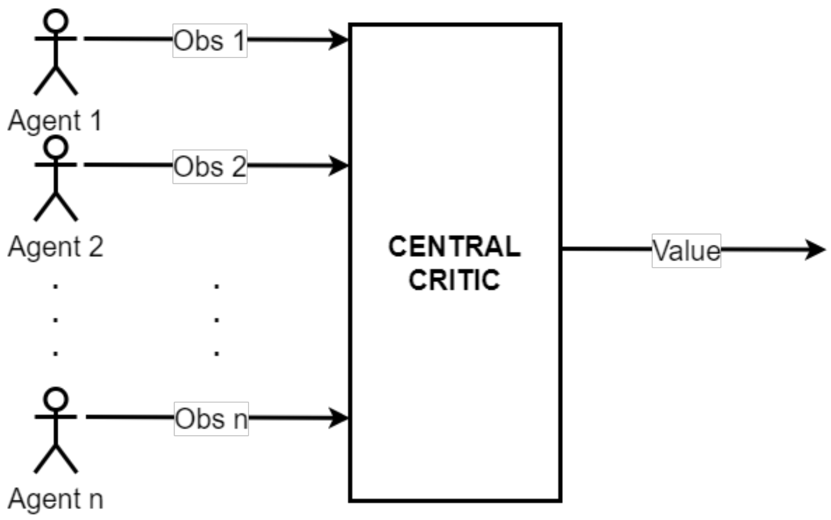
Issues: Non-stationarity, partial observability, restricted communication
↓
Centralized Training Decentralized Execution Paradigm (CTDE)

Overview

Centralized Training Decentralized Execution



Number of agents in training is fixed!

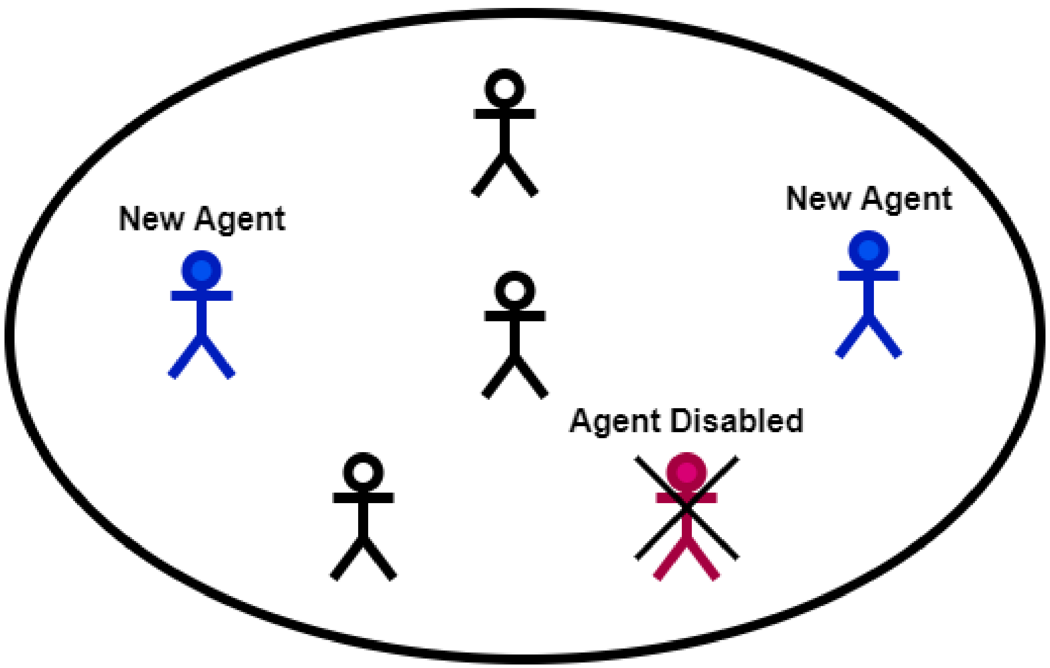


In real world scenarios:

- Number of agents vary
- Unguaranteed communication necessitate decentralized policies

Motivation and Approach

Environment with Variable Number of Agents



As number of agents ↑ scalability becomes an issue.

We investigate:

- Can learned decentralized policy work for settings with more / less agents?
- Are resulting policies good enough for use in systems with many more agents?

We show:

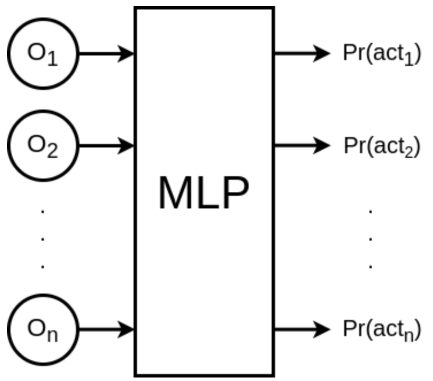
- Environment is key and a sweet spot exists for the optimal number of agents to train,
- Optimal agent count to train is different than target.
- Transfer across large number of agents can be a more efficient solution to scaling up in some environments

Table of Contents

Algorithm and Network Architecture

Decentralized Policy Network:

Multilayer Perceptrone

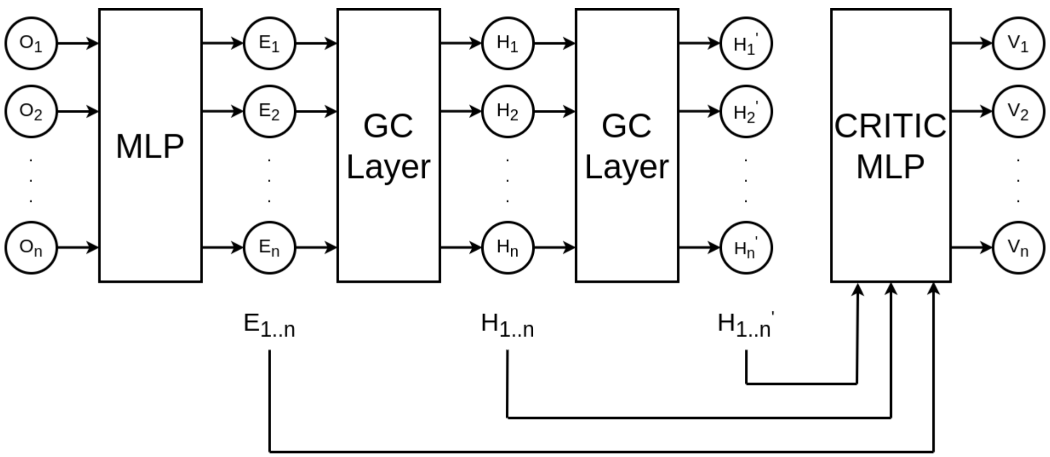


Algorithm: PPO
(Proximal Policy Optimization) [1]

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}.$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\Big[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\Big]$$

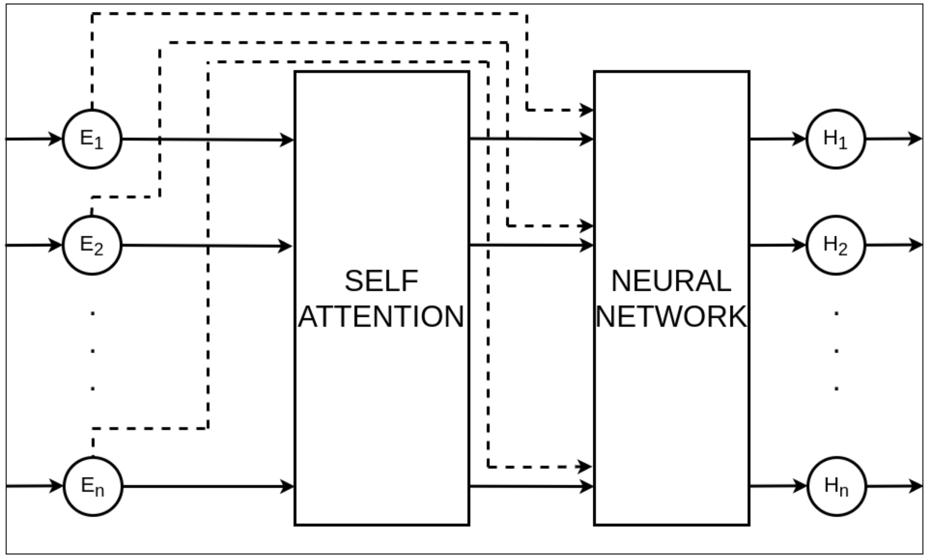
Centralized Critic Network



GC Layer: Graph Convolutional Layer with Self Attention Modules
o_i: observation of agent i - single time-step partial observation

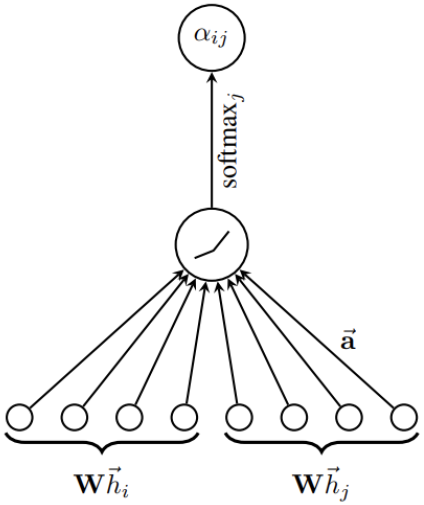
Algorithm and Network Architecture

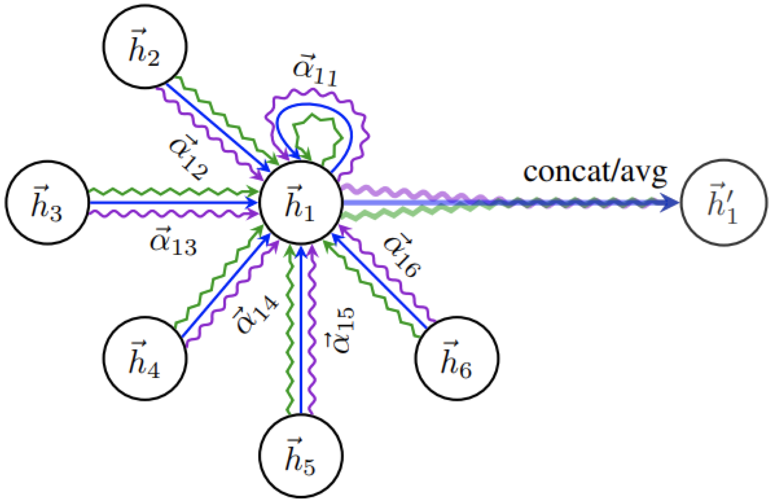
Graph Convolutional Layer with Self Attention



Self attention is used - only 1 attention head is used.

Graph Attention Networks [2]





Evaluation Method

1. Agent capacity of environment is determined. Number of agents to train and evaluate system performance is determined.
2. For each determined training number, system is trained at that fixed count until performance converges.
3. For each trained model, system is evaluated for all determined agent counts
4. Results are grouped per number of agents in evaluation. Performances of training for each agent count are analyzed and compared.

Table of Contents

Predator Prey Environment

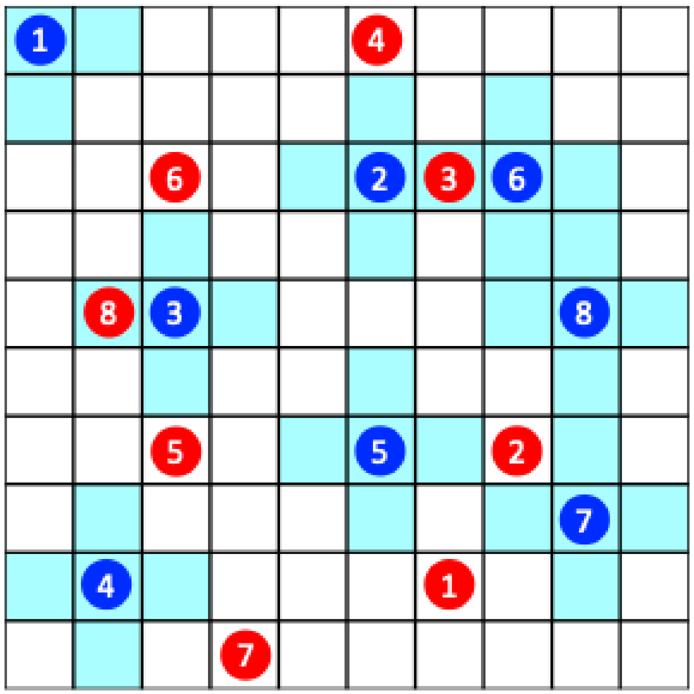


Figure 1: Predator Prey Environment [3] example grid size: 10×10

- Blue: Predators - Red: Preys
- Preys act $\rightarrow$ Predetermined rules + randomness
- Used grid size: 20×20
- Uncoordinated captures penalized
- Max agent capacity determined : 80 predators 80 preys
- Train and Evaluation agent counts determined: 2 - 5 - 10 - 20 - 50 - 80

Traffic Junction Environment

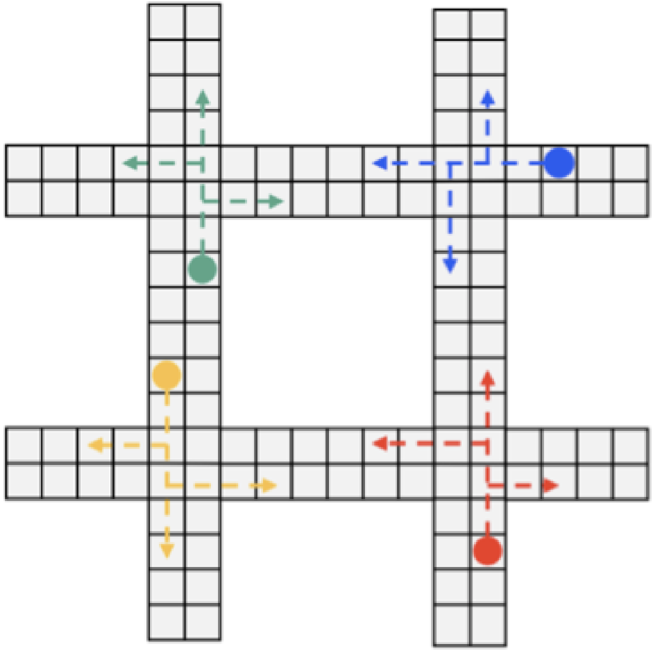


Figure 2: Traffic Junction Environment [3]
mode: hard

- Environment mode: hard - 4 junctions
- Goal: reach destination without accident
- Max agent capacity determined : 20 agents
- Train and Evaluation agent counts determined: 3 - 5 - 10 - 15 - 20

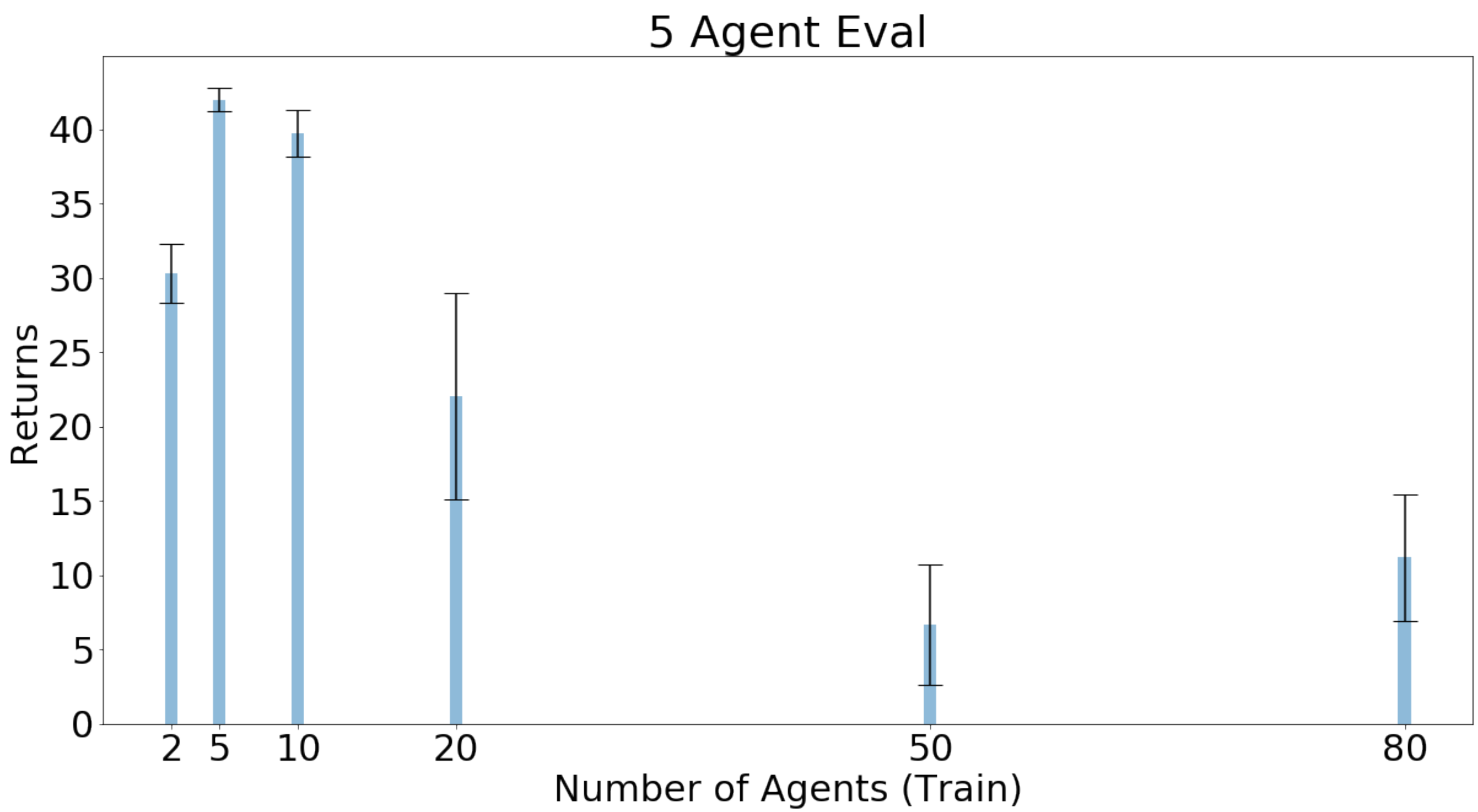
Predator Prey Environment Evaluation Results

	2	5	10	20	50	80
2	-2.01 ± 1.56	30.30 ± 1.95	86.48 ± 2.69	192.38 ± 1.88	496.58 ± 0.56	797.30 ± 0.59
5	$+4.62 \pm 1.46$	41.98 ± 0.77	94.60 ± 0.78	196.37 ± 0.58	497.96 ± 0.40	798.07 ± 0.82
10	-2.93 ± 1.10	39.71 ± 1.56	95.35 ± 0.21	196.95 ± 0.07	498.18 ± 0.37	798.28 ± 0.76
20	-11.75 ± 2.09	22.05 ± 6.95	84.60 ± 7.86	194.08 ± 2.48	496.80 ± 0.47	794.74 ± 2.30
50	-16.52 ± 1.68	6.68 ± 4.03	63.39 ± 4.61	182.00 ± 2.86	494.68 ± 1.00	795.90 ± 1.45
80	-13.96 ± 1.00	11.17 ± 4.29	62.57 ± 7.94	168.27 ± 8.47	484.42 ± 4.33	789.59 ± 2.09

Table 1: Mean of Total Rewards for Predator Prey
Columns: number of agents in evaluation, rows: number of agents in training.

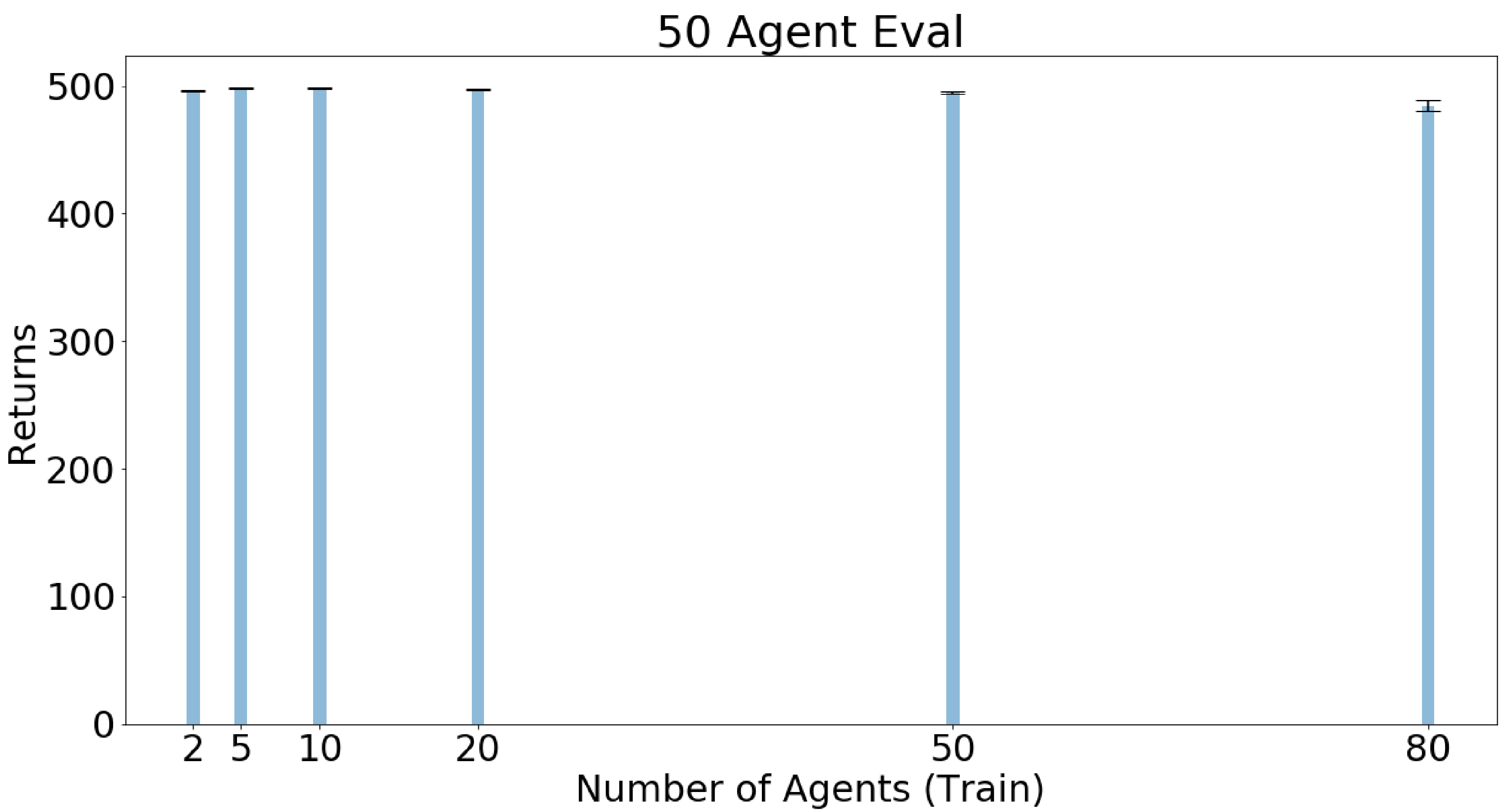
- Models trained with few number of agents have high generalization and transfer capacity for execution with high number of agents.
- The reverse is not true.
- Models trained with high number of agents have low generalization and transfer capacity for execution with low number of agents.
- Choosing number of agents to train from the range [5, 10] would be the better choice for transfer to system with any number of agent count

Predator Prey Environment Evaluation Results



For 5 agent evaluation case: Training with 5 agents gives the best evaluation result with 10 agent case following it. Models trained with large number of agents such as 50-80 have very poor performance.

Predator Prey Environment Evaluation Results



For 50 agent evaluation case: Training with 10 agents gives the best evaluation results with 5 agent case following it. 50 agent training case has the worst performance. (performance differences are marginal)

Traffic Junction Evaluation Results

	3	5	10	15	20
3	0.99 ± 0	0.92 ± 0.04	0.56 ± 0.10	0.26 ± 0.14	0.23 ± 0.15
5	0.99 ± 0	0.97 ± 0.01	0.77 ± 0.09	0.58 ± 0.16	0.58 ± 0.18
10	1.00 ± 0	0.99 ± 0.00	0.95 ± 0.01	0.84 ± 0.07	0.73 ± 0.09
15	1.00 ± 0	0.99 ± 0.01	0.94 ± 0.01	0.85 ± 0.03	0.79 ± 0.05
20	0.99 ± 0	0.99 ± 0.00	0.90 ± 0.02	0.83 ± 0.04	0.79 ± 0.03

Table 2: Mean of Success Rates for Traffic Junction
Columns: number of agents in evaluation, rows:number of agents in training.

- Models trained with few number of agents such as 3-5 get evaluation results with much lower success rate compared to the evaluation results of models that are trained with large number of agents such as 15-20.
- Models that are trained with 15-20 agents have very high success rate for the evaluation cases where there are 3-5 agents in the environment.
- Models trained with few number of agents can not sufficiently transfer for execution with high number of agents.
- Environment dynamic is key for transfer.

Traffic Junction Environment Evaluation Results

Models that are trained with 15-20 agents have very high success rate for the evaluation case with 5 agents in the environment

Traffic Junction Environment Evaluation Results

Figure 3: 20 Agents

Models that are trained with 3-5 agents have very low success rate for the evaluation cases where there are 15-20 agents in the environment.

References I

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, 2017. arXiv: 1707.06347 [cs.LG].

[2] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018. arXiv: 1710.10903 [stat.ML].

[3] S. Li, J. K. Gupta, P. Morales, R. Allen, and M. J. Kochenderfer, *Deep implicit coordination graphs for multi-agent reinforcement learning*, 2021. arXiv: 2006.11438 [cs.LG].