Comparative Evaluation of Cooperative Multi-Agent Deep Reinforcement Learning Algorithms

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, Stefano V. Albrecht

Autonomous Agents Research Group
School of Informatics
University of Edinburgh

1. Introduction and Motivation
2. Algorithms
3. Evaluation Environments
4. Evaluation Metrics
5. Results
6. Discussion
7. Conclusion

Introduction and Motivation

- Lack of commonly used benchmarking environments
- Lack of consistent algorithm implementations

Introduction and Motivation

- Lack of commonly used benchmarking environments
- Lack of consistent algorithm implementations
- Lack of consistent evaluation
- Measuring progress in MARL research

Contributions

- Evaluate seven commonly used MARL algorithms
- We open-source two evaluation environments
- We discuss and analyse their performance with several metrics

1. Introduction and Motivation
2. Algorithms
3. Evaluation Environments
4. Evaluation Metrics
5. Results
6. Discussion
7. Conclusion

Algorithms

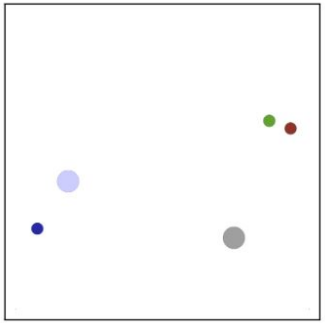
We evaluate three type of algorithms:

1. Independent Learners: IQL [11], IA2C [4, 6]
2. Centralised Multi-Agent Policy Gradient: MADDPG [7], COMA [5], Central-V
3. Value Decomposition: VDN [10], QMIX [8]

	Centr. Training	Off-/On-policy	Value-based	Policy-based
IQL	✗	Off	✓	✗
IA2C	✗	On	✓	✓
MADDPG	✓	Off	✓	✓
COMA	✓	On	✓	✓
Central-V	✓	On	✓	✓
VDN	✓	Off	✓	✗
QMIX	✓	Off	✓	✗

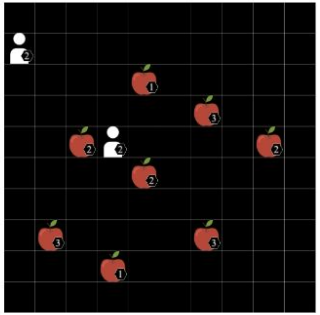
1. Introduction and Motivation
2. Algorithms
3. Evaluation Environments
4. Evaluation Metrics
5. Results
6. Discussion
7. Conclusion

Environments

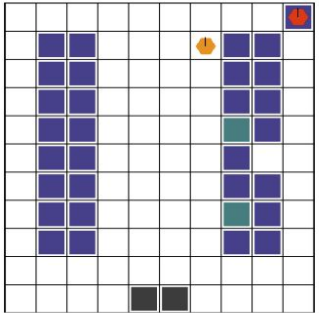


MPE [7]

SMAC [9]



LBF



RWARE

	Observability	Rew. Sparsity	Agents
MPE	Partial / Full	Dense	2-3
SMAC	Partial	Dense	2-9
LBF	Partial / Full	Sparse	2-3
RWARE	Partial	Sparse	2-4

Environments

+Matrix games [3]

$$\begin{bmatrix} 0 & 6 & 5 \\ -30 & 7 & 0 \\ 11 & -30 & 0 \end{bmatrix}$$

Climbing

$$\begin{bmatrix} k & 0 & 10 \\ 0 & 2 & 0 \\ 10 & 0 & k \end{bmatrix}$$

Penalty

	Observability	Rew. Sparsity	Agents
MPE	Partial / Full	Dense	2-3
SMAC	Partial	Dense	2-9
LBF	Partial / Full	Sparse	2-3
RWARE	Partial	Sparse	2-4

1. Introduction and Motivation
2. Algorithms
3. Evaluation Environments
4. Evaluation Metrics
5. Results
6. Discussion
7. Conclusion

Evaluation Protocol

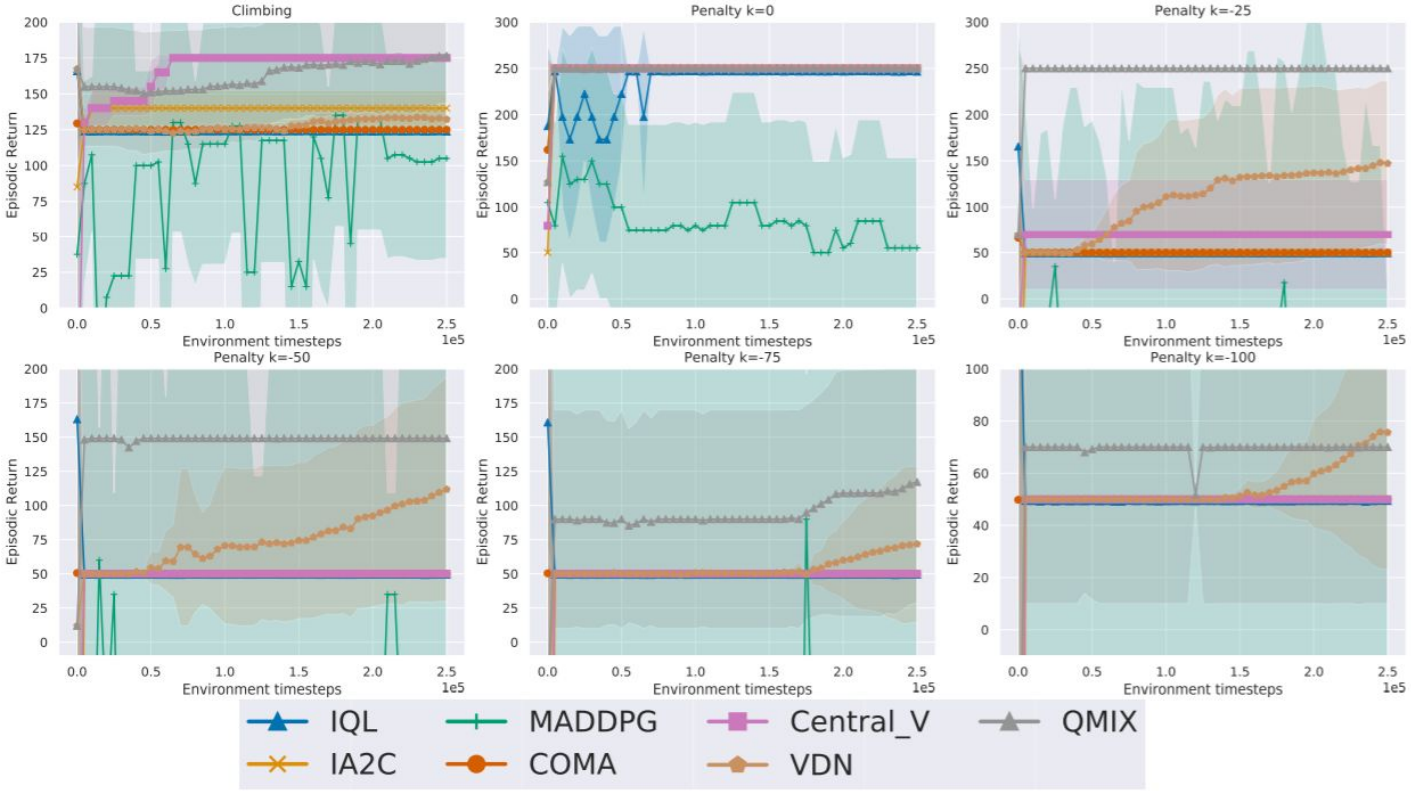
- In each algorithm and task we train several hyperparameter combinations for 10 seeds each
- In Matrix Games:
  - Train on-policy algorithms for 2.5M steps
  - Train off-policy algorithms for 250K steps
  - Evaluate 100 times during training for 100 episodes at each evaluation
- In the rest:
  - Train on-policy algorithms for 20M steps
  - Train off-policy algorithms for 2M steps
  - Evaluate 40 times during training for 100 episodes at each evaluation

Evaluation Metrics

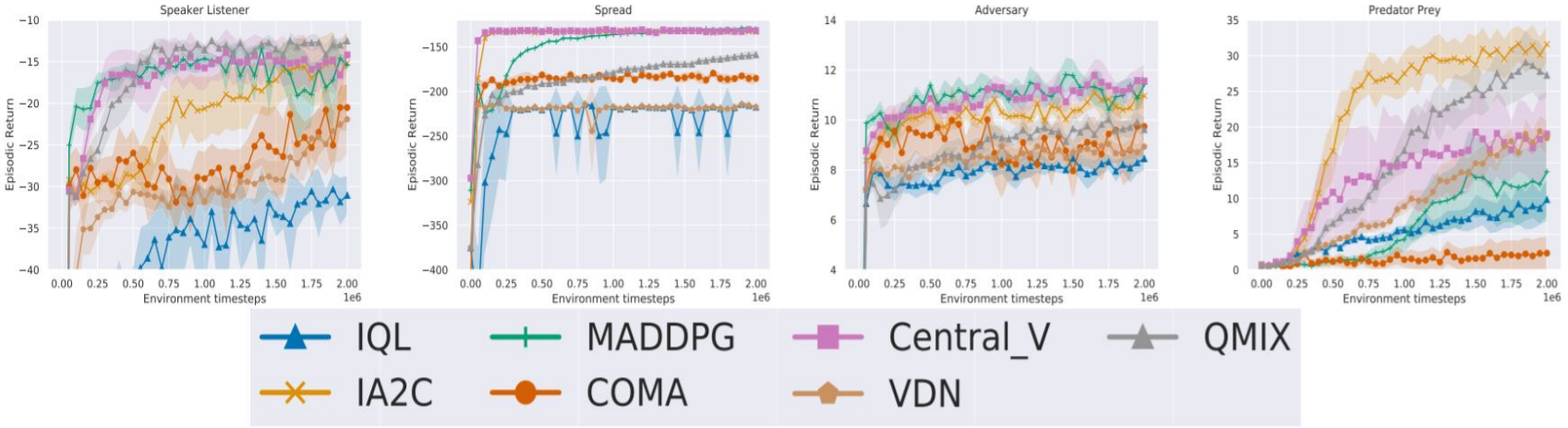
- Maximum returns (maximum evaluation over different hyperparameters and evaluation points)
- Average returns (average among all evaluation points)
- Reliability metrics [1]:
  - Dispersion across Time (variability across training evaluations)
  - Short-term Risk across Time (maximum drop between two evaluations)
  - Long-term Risk across Time (maximum drop between any evaluation and the so-far best evaluation)
  - Dispersion across Runs (variability across different seeds)

1. Introduction and Motivation
2. Algorithms
3. Evaluation Environments
4. Evaluation Metrics
5. Results
6. Discussion
7. Conclusion

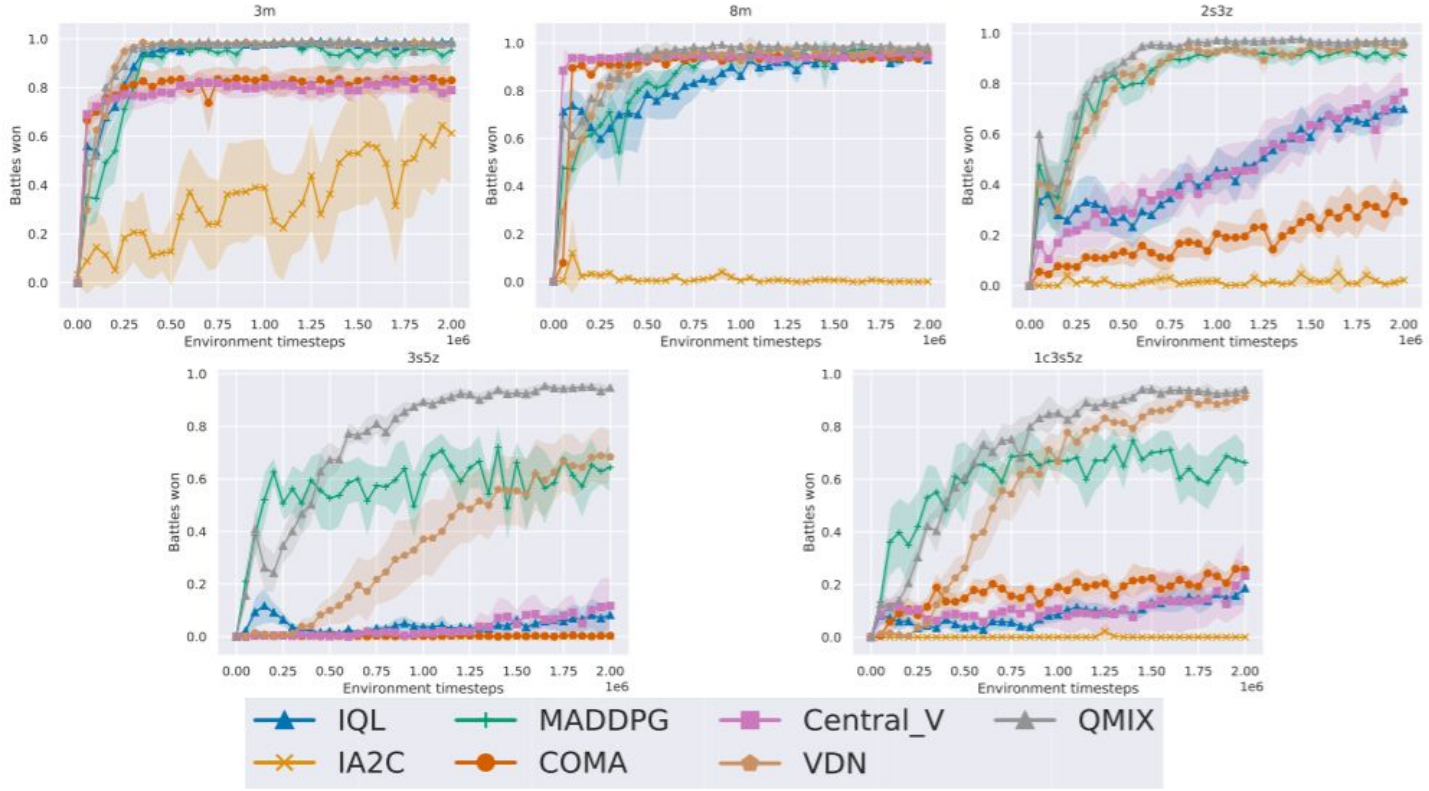
Results in Matrix Games



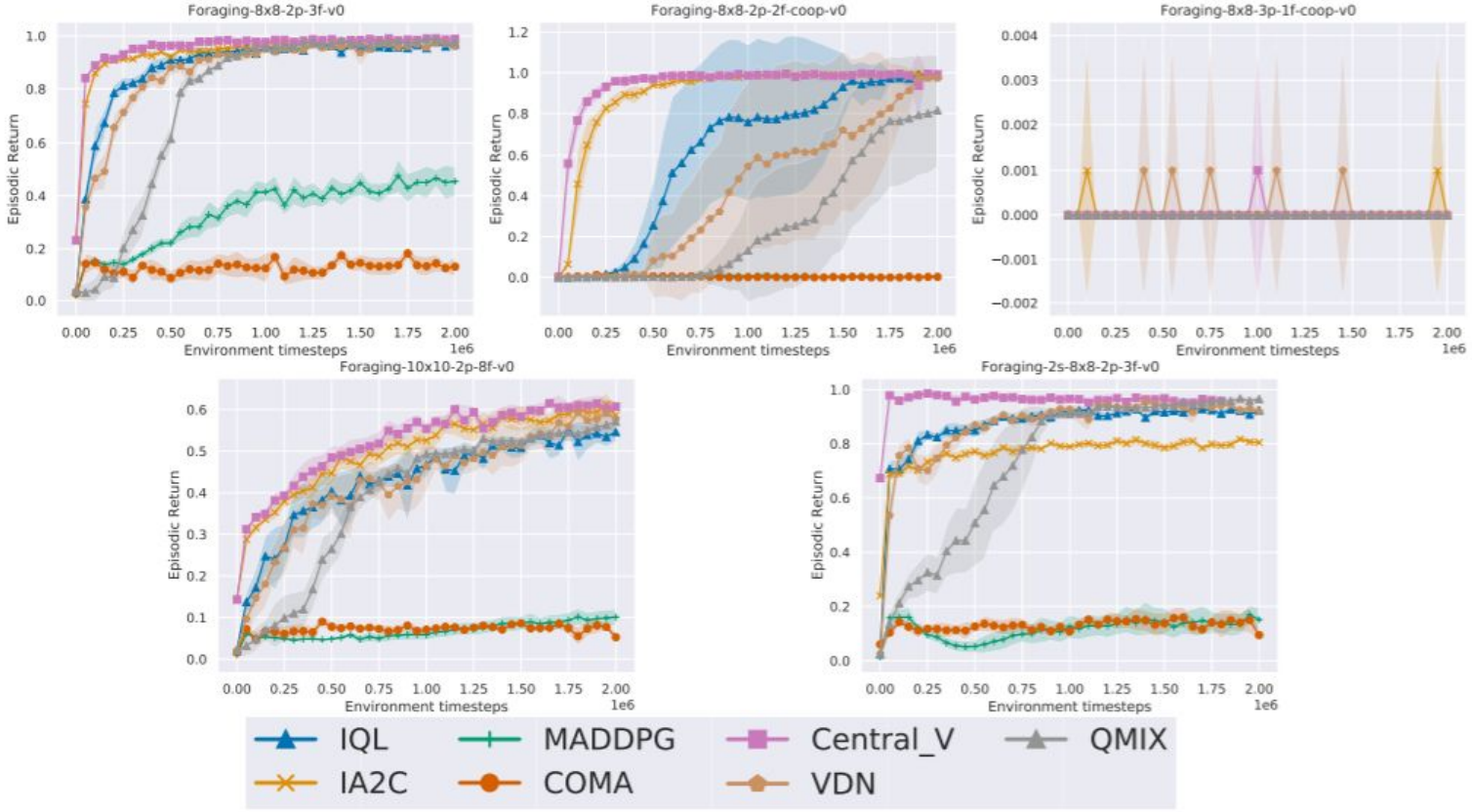
Results in MPE



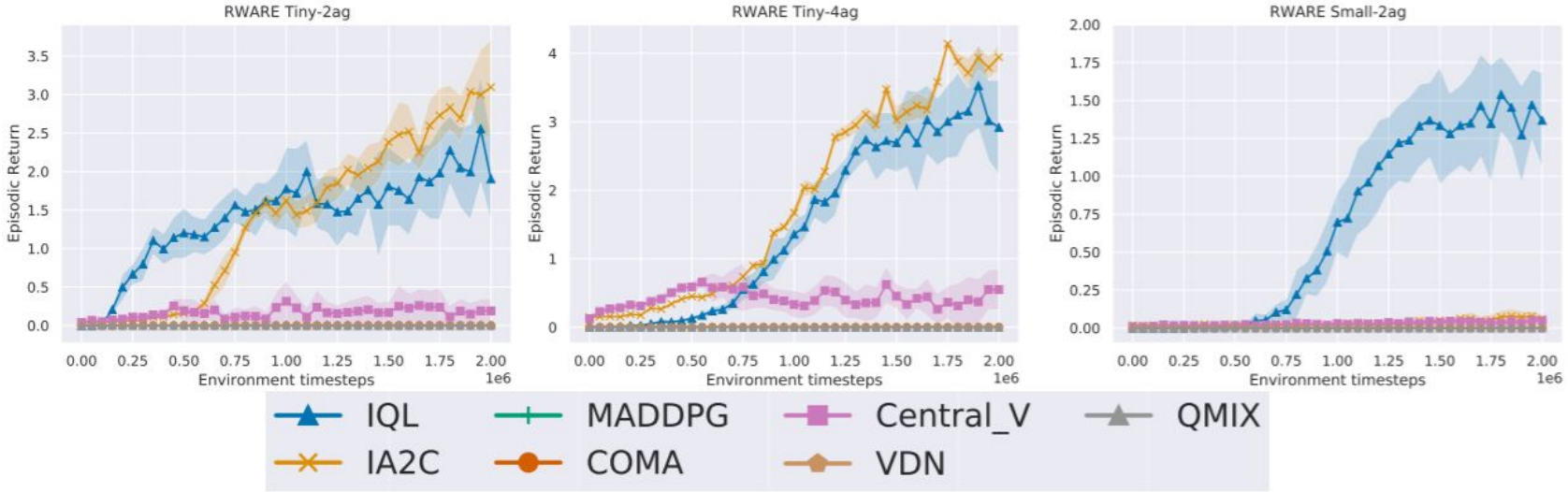
Results in SMAC



Results in LBF



Results in RWARE



1. Introduction and Motivation
2. Algorithms
3. Evaluation Environments
4. Evaluation Metrics
5. Results
6. Discussion
7. Conclusion

Independent Learners

- Achieve high returns in most fully-observable environments
- Struggle to coordinate in partially-observable environments
- Cannot reason about the joint action of all agents
- Experience replay hinders the learning of IQL
- IA2C achieves better returns in coordination tasks because it learns on-policy

Centralised Multi-Agent Policy Gradient

- Most effective in partially-observable environments
- MADDPG suffers from the biased sampling (Gumbel-Softmax)
- COMA suffers from high variance in the baseline computation
- Central-V is unable to reason about joint actions
- Centralised training can complicate the training procedure

Value Decomposition

- Achieve high returns in the majority of environments
- Can reason about joint trajectories
- VDN is bounded by the linear decomposition
- Require highly informative reward signal

1. Introduction and Motivation
2. Algorithms
3. Evaluation Environments
4. Evaluation Metrics
5. Results
6. Discussion
7. Conclusion

Future Work

- Include more algorithms such as IPPO, MAPPO
- Include more environments
- Evaluate implementation details, such as parameter sharing [2], reward standardisation

Comparative Evaluation of Cooperative Multi-Agent Deep Reinforcement Learning Algorithms

https://arxiv.org/abs/2006.07869

Contributions:

1. We evaluate and compare seven MARL algorithms in 23 tasks using several evaluation metrics

2. We discuss and analyse the main benefits and limitations of the evaluated algorithms

References

[1] Stephanie CY Chan, Sam Fishman, John Canny, Anoop Korattikara, and Sergio Guadarrama. 2020. Measuring the Reliability of Reinforcement Learning Algorithms. International Conference on Learning Representations (2020).

[2] Filippos Christianos, Georgios Papoudakis, Arrasy Rahman, and Stefano V. Albrecht. "Scaling Multi-Agent Reinforcement Learning with Selective Parameter Sharing." ALA Workshop (2021).

[3] Caroline Claus and Craig Boutilier. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. AAAI Conference on Artificial Intelligence (1998).

[4] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. 2017. OpenAI Baselines. https://github.com/openai/baselines.

[5] Jakob N Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. AAAI Conference on Artificial Intelligence (2018).

[6] 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. International Conference on Machine Learning (2016).

[7] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. Neural Information Processing Systems (2017).

[8] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. International Conference on Machine Learning (2018).

[9] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft multi-agent challenge. (2019).

[10] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2018. Value-Decomposition networks for cooperative multi-agent learning. International Conference on Autonomous Agents and Multi-Agent Systems (2018).

[11] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. International Conference on Machine Learning (1993).