UNDER REVIEW

RESCIENCE C

Replication / ML Reproducibility Challenge 2022

Reproducibility study of "Cooperative Multi-Agent Fairness and Equivariant Policies"

Anonymous[1], **Anonymous**[2], **Anonymous**[3]**Anonymous**[4]

[1,] Anonymous Affiliaton – [2]Anonymous Affiliaton – [3]Anonymous Affiliaton – [4]Anonymous Affiliaton

Edited by
(Editor)

Reviewed by
(Reviewer 1)
(Reviewer 2)

Received
03 February 2023

Published
–

DOI
–

Reproducibility Summary

Scope of Reproducibility – In this work, we study the reproducibility of the paper *Cooperative Multi-Agent Fairness and Equivariant Policy* written by Grupen et al. We aim to prove their main claims, while also verifying that they generalise to different settings (smaller world and different number of agents). The original claims are: (i) Mutual reward reinforcement learning leads to multi-agent coordination, (ii) the use of equivariant policy learning to achieve fair outcomes for individual members of a cooperative team, (iii) the utilization of the soft-constraint adaptation of this model, which achieves fairer outcomes than non - equivariant learning, and (iv) the fact that the magnitude of the fairness and utility trade-off depends on agent skill.

Methodology – We forked and adapted the Github repository provided by the authors to run the experiments. We had to create a working Python environment and re-implement some parts of the code. The repository of the code can be found on the footer of this page.

Results – We proved through empirical results that the main claims of the original paper, (i) mutual reward coordination leads to a higher utility, (ii) equivariant policy learning leads to fairer outcomes for individual members of a cooperative team, and (iv) the magnitude of the fairness-utility trade-off depends on agent skill. However, we were unable to reproduce the third claim (iii) which stated that Fair-ER leads to a higher utility than Fair-E and achieves fairer outcomes than non-equivariant learning. Finally, it was proved that the first claim (i), can be extended to a setting with four agents while the second claim (ii) can not.

What was easy – Main objectives in the original paper were clearly stated, which made the reproducibility easier. Codes related to scenario, simulation and modelling were well designed and easy to reproduce.

What was difficult – The most challenging issues that were faced is the environment setup, understanding of the code, and the extensive computation time.

Communication with original authors – Communication with the original authors was attempted through email two weeks before the deadline, but unfortunately, a response was not received until the submission deadline due to the short notice.

Correspondence should be addressed to ()
The authors have declared that no competing interests exists.
Code is available at https://anonymous.4open.science/r/multiagent_fairness_reproducibility-6934.

1 Introduction

Multi-Agent Reinforcement Learning (RL) has been of increasing importance when modelling social and economic challenges such as taxation and economic policy, as interactions between agents lead to actual actions towards accomplishing a common goal[1]. As these learning objectives can be efficiently resolved by teams of agents, it becomes crucial to understand the range of team behaviours that emerge in this situation.

Fairness is one of the scopes through which this team behaviour can be analyzed, as it was originally studied by Drew et al.[2]. This topic is also studied by Grupen et al. in the context of Cooperative multi-agent settings [3], where they seek to understand the fairness implications of emergent coordination learned by multi-agent teams that are bound by a shared reward and equivariant learning policies.

Our study's most important contributions are the following:

1. Proved which of the original paper's claims generalize to a smaller, more constrained world.
2. Extended the original experiments to an environment with an increased number of pursuers while allowing to add multiple evaders for further testing.
3. Improved scripts for model evaluation of Fair-E and Fair-ER.
4. Provided working environments for both Windows and Linux.

2 Scope of reproducibility

The development of multi-agent systems requires a thorough understanding of the impact that reward shaping can have on agents' behaviour. Hence, investigating the influence of mutual and individual rewards in fully cooperative multi-agent systems[4] was an initial step in the original paper [3]. The design of the agent's reward functions can impact the level of utility achieved, which is evaluated in relation to fairness within the team. Team fairness is a prevalent concept in the original research inspired by demographic parity concepts ([5][6]), which can be achieved by ensuring a uniform distribution of rewards among the agents within the team. The authors showed that it is possible to enforce team fairness by transforming the team's policies into an equivariant map. Two different methods are introduced to achieve this purpose, Fairness through Equivariance (Fair-E) and Fairness through Equivariance Regularisation (Fair-ER). The former imposes strict equivariance on the team's policies while the latter can be seen as a soft-constraint adaptation of Fair-E inspired by the work of Liu et al. [7]. Furthermore, the fairness-utility trade-off, a topic both prevalent in game-theoretic multi-agent settings([8], [9], [10]) and prediction-based settings[11], is studied and evaluated by the authors.

This study aims to reproduce the main claims of this paper while also analysing their robustness and building upon them. The original claims can be summarised as the following:

1. Mutual reward is critical to multi-agent coordination, favouring collaboration between agents and increasing utility. Conversely, individual rewards foster an environment of competitiveness between agents of the same team.
2. Fair-E achieves fair outcomes for individual members of a cooperative team.
3. Fair-ER leads to higher levels of utility than Fair-E and achieves fairer outcomes than Non-Equivariant learning.

4. The magnitude of the fairness-utility trade-off depends on agent skill. The higher the agent's skill, the easier it is to guarantee fairness.

Besides verifying the original claims of the paper, this study also focuses on proving their generalizability by scaling down the simulation world. The main motivation behind this was to provide a more competitive environment to verify whether fairness can still be achieved, while also reducing the amount of training time. Additionally, the number of evaders and pursuers was also incremented while maintaining the reduced world size for the same aforementioned reasons. It is expected, that conclusions of the original paper would hold even in the modified settings.

3 Fair-E and Fair-ER

Deep Deterministic Policy Gradients (DDPG) is a reinforcement learning algorithm based on the "actor-critic" technique [12] which follows the deterministic policy gradient theorem [13]. The actor is a policy network that will find the exact action based on the agent state as input. The critic is a Deep Q-learning network that evaluates the action by computing the value function with the action and state as input [14]. Unlike other policy-based methods, DDPG computes the action directly, resulting in a deterministic algorithm. Furthermore, it operates using off-policy data, meaning that it can learn from past experiences stored in a replay buffer rather than just the current episode [12],[15].

In the research paper we aim to reproduce, a novel multi-agent learning strategy based on Fairness Through Equivariance (Fair-E) is introduced. Applying fairness in multi-agent learning could help these systems become both more efficient and stable[16]. By enforcing parameter symmetry in each agent's policy, Grupen et al. [3] show that equivariance is established throughout the multi-agent RL problem, expanding on the work from Ravanbakhsh et al. [17]. Equivariance has been applied extensively in the recent works ([18][19]), where it has been proved that symmetry can increase robustness and efficiency in different applications. However, by equivariance in the context of Reinforcement Learning, it is implied that separate policies would take the same actions under permutations of state space. More specifically for an equivariant joint policy, applying a transformation σ to the state s_t (producing $\sigma \cdot s_t$) and then running the policy $\pi(\sigma \cdot s_t)$ is equivalent to running the joint policy first and then applying the transformation (resulting in the equality $\pi(\sigma \cdot s_t) = \sigma \cdot \pi(s_t)$).

Therefore, parameter sharing allows for equivariant policies, enabling multi-agent policies and trajectories, to yield exact team fairness. This hard-constraint method imposes team fairness in a rigid manner. However, it can be beneficial to tune the level of fairness enforced on the agent team. To allow for this, the authors propose Fairness Through Equivariance Regularisation (Fair-ER), which is a soft-constraint version of Fair-E. Fair-ER relies on an adapted cost function that includes an additional regularisation term:

$$J_{eqv}(\phi_1, ..., \phi_i, ..., \phi n) = \mathop{\mathbb{E}}_{s}[\mathop{\mathbb{E}}_{j \neq i}[1 - cos(\pi_{\phi_i}(s) - \pi_{\phi_j}(s))|_{s=s_t}]] \tag{1}$$

This term penalizes agents proportionally to the amount their actions differ from the actions of their teammates. Thus, it encourages equivariance. This is then added to the initial DDPG RL objective, resulting in the following:

$$J(\phi_i) + \lambda J_{eqv}(\phi_1, ..., \phi_i, ..., \phi n) \tag{2}$$

where the parameter λ is used to tune the applied fairness.

4 Methodology

The repository used by the authors to perform the experiments is publicly available on GitHub. Some of the experiments were not able to run successfully, thus, re-implementation of some parts was essential, including resolving important training issues like model checkpoint loading and modifying equivariance and collaboration settings for the different models. In addition, a script for plotting all of the figures of the original paper was added given the trajectories obtained after evaluation.

To speed up the process of training and to examine the paper's generalizability as mentioned in Section 2, different models were evaluated in a smaller environment with a varying number of agents. More specifically, the world size of the environment and the maximum amount of steps per episode were reduced by a factor of 3. Other hyperparameters were modified accordingly for consistency. Refer to Section 4.1 for a more detailed explanation of these changes.

4.1 Hyperparameters

As many of the hyperparameters used for the different runs were not mentioned in the original paper, it was assumed that the default hyperparameters present in the original code were the ones used to generate the different figures. As mentioned in Section 4, some hyperparameters were changed, including world size (reduced from 6.0 to 3.0) and the number of training episodes (reduced from 125000 to 42000). However, to ensure compatibility with these changes, we also made adjustments to other hyperparameters such as the distance start and end, and the number of steps per episode.

You can refer to Table 1 in Appendix 1 for a detailed explanation of the different hyperparameters used with their default values as found in the code and our changes made to them.

4.2 Experimental setup and code

The experimental setup of this work is largely based on the repository provided by the original paper. The training scripts have been optimized with the addition of command line arguments, allowing for more efficient customization of equivariance and collaboration settings. For both training and testing the models, separate scripts for Fair-E and Fair-ER have been provided. The goal was to reduce testing difficulty, as in the original code different parts of the script had to be manually changed to either evaluate the Fair-E or Fair-ER models.
The repositories with our code modifications have been structured in the same way as the original code repositories and can be found in Appendix 9.

In Section 5, the different measurements used to evaluate the experiments were the capture rate and the mutual information. The former is the rate of capture of the evader by the agents, while the latter is the mutual information between reward distributions R and sensitive variables Z, which is equal to zero for a perfectly fair team [3].

4.3 Computational requirements

Three computers geared with Intel®Core™i7 CPUs were used instead. The total training time required was of approximately 300 hours.

5 Results

This section will focus on explaining the different results that were obtained when trying to reproduce the paper's claims and building upon these.

5.1 Results reproducing original paper

Importance of Mutual Reward — For the first experiment, 2 different models using Non-Equivariant learning policies (DDPG model) with either shared or individual rewards were tested. In the mutual reward setting, each pursuer receives the total sum of the reward when capturing the evader, while in the individual reward setting, only the pursuer that catches the evader gets rewarded.

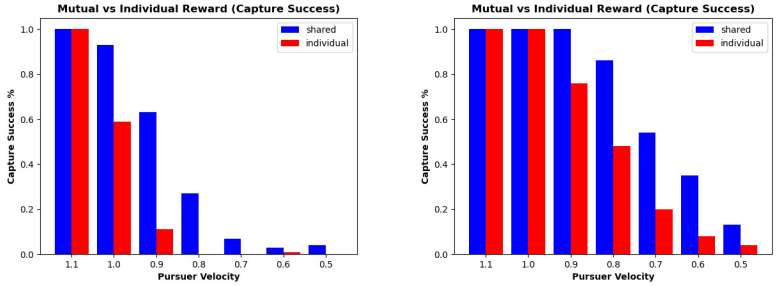

Figure 1. Performance of policies trained with individual vs mutual reward. It can be observed that as pursuer velocity decreases, the task of capturing the evader requires more sophisticated coordination. On the left, the set-up is trained with 3 agents, while on the right with 4 agents.

From the results, as seen in Figure 1 (left), the original paper's claim that mutual reward is crucial for increased utility is supported. The agents are incentivised to work together and thus have developed a coordination strategy that performs better than any individual strategy. This means that for pursuer velocities lower than the evader's (< 1.0), the capture success rate when using mutual reward will be higher as the agents are forced to collaborate to catch the evader more efficiently. Therefore, the first claim mentioned in Section 2 is supported in both original and smaller worlds.

Fair Outcomes with Fair-E — In this experiment, the claim that was verified was whether Fair-E enforces fairness on each agent within a team. This is demonstrated by analysing the distribution of the reward vector across different evaluation episodes.
From Figure 2 (top), it can firstly be said that the reward vector for the Non-Equivariant policy is not evenly distributed among individual agents ([r, 0, 0] = agent 0, [0, r, 0] = agent 1, [0, 0, r] = agent 2). The reason behind this is that role assignment takes place within the team, leading to an unfair reward distribution. Role assignment means that two pursuers take supporting roles guiding the evader towards the other agent which captures the evader more frequently. It can also be observed from the same Figure that [0, r, r] and [r, 0, r] have non-zero values, which corresponds to a double capture, which occurs when the evader is captured by two predators simultaneously. This happens due to the reduced world size, and it is not of significant importance in this analysis.
Additionally, in Figure 2 (bottom), when using the Fair-E policy, the rewards are more evenly distributed, which indicates increased amount of fairness. Although the reward distribution is not entirely uniformly distributed as in the original paper, it is believed that if the model was trained for more episodes it would have reached ideal fairness. In conclusion, the original paper's claim (second claim stated in Section 2) that Fair-E policy achieves fair outcomes for individual members of a cooperative team is also supported.

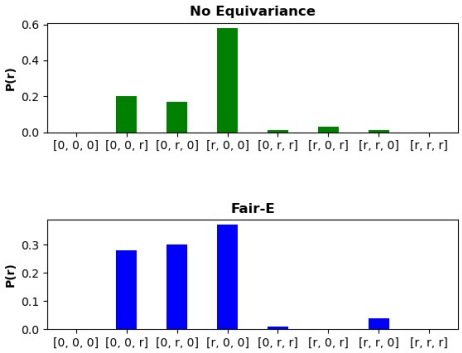

Figure 2. Distribution of reward vectors for both strategies at the pursuer velocity $|V_p| = 1.1$. Non-equivariant policy (top) has uneven reward distribution due to role assignment, while Fair-E policy (bottom) has a more uniform reward distribution.

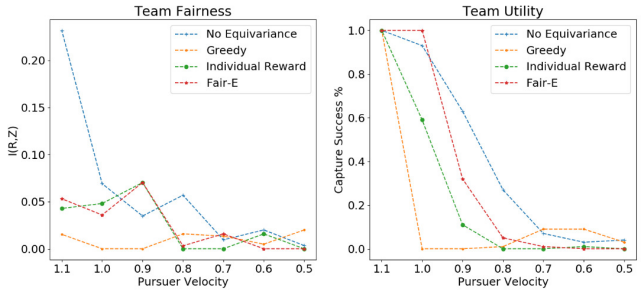

Figure 3. Team fairness score (lower score better fairness) for several strategies (left) across different speeds. It can be seen that the Non-Equivariant policy has the least amount of fairness. Team utility score (higher score better fairness) for several strategies across different speeds (right).

Regarding the team fairness score for different strategies, it can be said that agents trained with Fair-E are fairer than those trained without equivariant policies as seen in Figure 3 (left). Fair-E achieves a lower I(R; Z), and therefore, higher team fairness. However, the level of fairness reached by Fair-E does not match with the results of the original paper. For low velocities, Fair-E is indeed the fairer model but it is only because the agents do not succeed to capture the evader, as shown in Figure 3 (right). In this context, they receive no rewards, which is considered a fair outcome in terms of mutual information.

In addition, from Figure 3 (right) we can also observe how Non-Equivariant policies (No Equivariance) achieve a higher utility than fairer equivariant policies (Greedy, Individual Reward, Fair-E). The hard constraint that Fair-E places on each of the agent's policy create a commitment towards achieving fair outcomes at the expense of utility, which supports the idea of an existing fairness-utility tradeoff as stated in the fourth claim in Section 2.

Modulating fairness with Fair-ER — For the next experiment, 6 different models were trained using Fair-ER with different values of lambdas as an attempt to reproduce the Fair-ER fairness vs utility study from the original paper. All of these models were tested with 7 different pursuer velocities: 1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5.
We evaluated the effect of the equivariance control parameter λ on policy learning using Fair-ER in figure 4. The results deviate from those reported in the original study, where Fair-E showed the lowest capture rate. They also found that Fair-E performed better

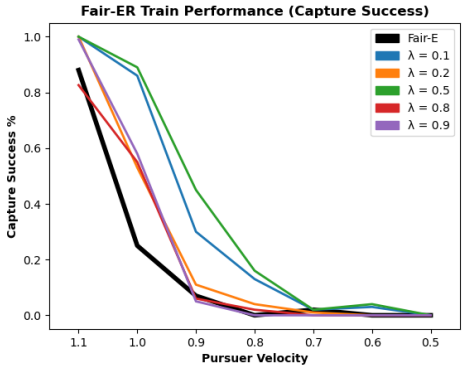

Figure 4. Effect of the equivariance control parameter λ on policy learning with Fair-ER. General trend can be seen where lower values of λ lead to higher capture success rate.

than Fair-ER for λ values of 0.9 and 0.8. However, the results provided in Figure 4 are not consistent with this expectation. For example, it can be seen that the agents trained with the Fair-ER model and $\lambda = 0.5$ outperformed those trained with $\lambda = 0.1$, while those trained with $\lambda = 0.9$ perform worse than both of these. A general trend can be observed where smaller values of λ lead to a higher capture success rate, but our results are not fully consistent across the whole testing range of values.

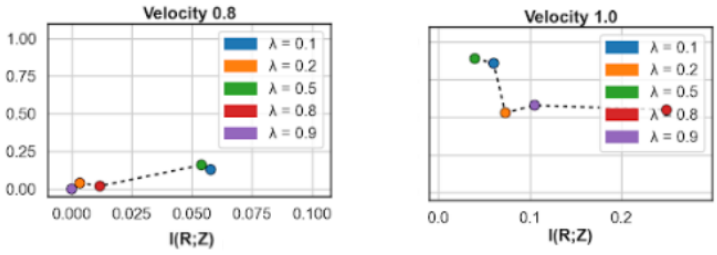

Figure 5. Fairness vs. utility comparisons for Fair-ER trained with various values of λ for different pursuer velocities.

In addition, comparisons between utility and fairness for different models of Fair-ER were plotted in Figure 5. No conclusion can be reached from the obtained results. In particular, it can be seen that the utility of the models for slower pursuers is lower than expected. Specifically, for all models, the capture rate is below 10% for a pursuer velocity of 0.7, while in the original paper, it was almost 90%. It is difficult to evaluate fairness through these results as for lower velocities the capture success approaches 0 for all configurations. This could be caused due to an underfitting of the models, which shows that training for more episodes could be interesting for further research. Regarding higher pursuer velocities, the capture rate remains low for velocities of 0.8, but increases for 1.0.

In conclusion, the third claim from Section 2 can not be fully validated. While Fair-ER does indeed provide higher utility than Fair-E, our results do not allow us to prove whether Fair-ER does indeed achieves fairer results than Non-Equivariant policies. The unexpected results may be attributed to the downscaling of the world size and the decreased number of training episodes. The reduction in the number of steps per episode may have also influenced these results since this increases the difficulty of catching the evader in the reduced number of steps. Consequently, the most affected results would be those with low pursuer velocities as more steps are required for catching the evader.

5.2 Results beyond original paper

Increasing the number of predators — Motivated by the low capture success rate, an additional experiment was performed by increasing the number of predators by one to make the task of capturing the evader easier. Given this modification to the original setting, we aim to find whether an increase in capture success rate does indeed occur, while still being able to support the first two claims from Section 2.
Figure 1 (right) shows how, as compared to Figure 1 (left), increasing the number of pursuers does indeed result in an increased capture success rate for the team. As expected, having more pursuers reduces the complexity of capturing the evader. Furthermore, it can also be observed how for pursuer velocities slower than the evader's, mutual reward is still essential to obtain a high capture success rate. Therefore, it can be shown that the first claim from Section 2 still holds.

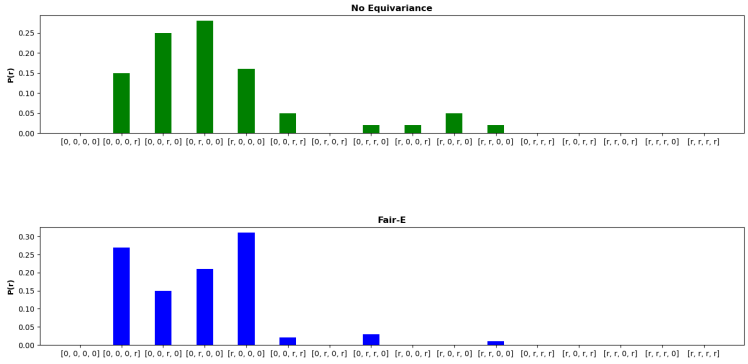

Figure 6. Distribution of reward vectors for both strategies at the pursuer velocity |Vp| = 1.0. Non-equivariant policy (top) and Fair-E (bottom). It can be seen that both strategies provide uneven reward distributions.

For proving the second claim stated in Section 2, an analysis equivalent to that shown in Figure 2 is performed. Observing Figure 6, it can be seen that for both Fair-E (bottom) and Non-Equivariant policies (top) the reward distribution remains unfair and no trend can be appreciated when comparing with the results from Figure 2. This shows how the second claim from Section 2 does not hold for an increased number of pursuers. One possible reason for this to happen is that the models could have gotten stuck in local minima. When observing this evaluation, the pursuers tended to oscillate around specific regions of the world, relying mainly on being initialised close to the prey for each episode to catch it.

This experiment shows that by increasing the number of pursuers the capture success rate improves without requiring the pursuers to be of high skill to obtain a high utility (as they learn to oscillate around fixed regions). Therefore, following claim four from Section 2, it was expected that in this setting it is much harder to guarantee fairness and achieve a uniform reward distribution as the agents are individually less skilled than in a smaller team.

6 Discussion

This study has focused on conducting experiments to support the claims of the paper written by Grupen et al [3]. Although we could not reproduce the exact same results, most of the main claims stated in Section 2 were supported while examining them in a different setting. Firstly, mutual reward can be crucial for an increased utility, as

agents are incentivized to cooperate, as elaborated in Section 5. Secondly, using Fair-E achieves fair outcomes as the reward distribution is more uniformly distributed when using equivariance as shown in Figure 2. Furthermore, Figure 3 supports the idea that Fair-ER achieves higher utility than Fair-E , but due to the inconsistency in the results a clear relationship between the values of lambda and the achieved fairness can not be established.

Interestingly, the only claim that could not be verified to its full extent is the way that different values of λ, correlate with the fairness-utility trade-off. We believe that these non-conclusive results are due to the decreased world size and decreased number of training episodes (changes explained in Section 4.1). Therefore, our results show how Fair-ER did not generalize to this new setting.

Last but not least, motivated by the low capture success rate with three pursuers, the first two claims were examined in a setting with an additional pursuer to verify whether these still hold. Although the first claim was verified successfully, which stated that mutual reward is still essential to obtain a higher capture success rate, the second claim could not be proved. Using Fair-E did not lead to a more equitable reward distribution than using a Non-Equivariant policy, supporting the fourth claim mentioned in Section 2 which states that it is much harder to obtain fairness when the agents are individually less skilled.

All in all, the original papers' contribution in the field of reinforcement learning is of significant importance, and thus, further research should be done to investigate the implication of fairness in different multi-agent scenarios. Reproducing these tasks can be quite challenging, and therefore, the code implementation of our study is publicly available.

6.1 What was easy

The task of identifying reproducibility objectives was straightforward, as the authors clearly outlined the contributions of the paper. Additionally, the scenario and simulation were successfully created and ran without any issues. The learning models were also implemented correctly, and any challenges encountered were primarily related to bugs in the hyperparameter settings and weight loading.

6.2 What was difficult

The initial challenge we faced was configuring the environment to work on the provided repository. The selected package versions and outdated version of Python used caused dependency issues when installing the environment. However, once the environment was set up, the next challenge was understanding the authors' code as the provided documentation was not sufficient. Additionally, some modifications to the code were needed to run the experiments without errors. Lastly, the extended training time was a limitation given our restricted computational resources.

6.3 Communication with original authors

We have tried to communicate with the original authors two weeks before the deadline to get information about the code implementation and technical details about the model's architecture. Unfortunately, an answer was not received but we understand that it is due to the short notice.

7 Acknowledgements

We would like to express our sincere gratitude to Dimitris Michailidis, our Teaching Assistant, for his invaluable guidance and support throughout the project. His expertise and dedication were crucial in helping us complete this project.
We also acknowledge the FACT course taught in University of Amsterdam for providing us with the opportunity to work on this project and gain hands-on experience in the field.

References

1. S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher. **The AI Economist: Taxation policy design via two-level deep multiagent reinforcement learning**. 2022, p. 2607.
2. D. S. Drew. "Multi-agent systems for search and rescue applications." In: **Current Robotics Reports** 2.2 (2021), pp. 189–200.
3. N. A. Grupen, B. Selman, and D. D. Lee. "Cooperative Multi-Agent Fairness and Equivariant Policies." In: **Proceedings of the AAAI Conference on Artificial Intelligence**. Vol. 36. 9. 2022, pp. 9350–9359.
4. J. Hao and H.-f. Leung. **Interactions in Multiagent Systems: Fairness, Social Optimality and Individual Rationality**. Jan. 2016, pp. 1–178.
5. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. "Fairness through awareness." In: **Proceedings of the 3rd innovations in theoretical computer science conference**. 2012, pp. 214–226.
6. M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. "Certifying and removing disparate impact." In: **proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining**. 2015, pp. 259–268.
7. L. T. Liu, S. Dean, E. Rolf, M. Simchowitz, and M. Hardt. "Delayed Impact of Fair Machine Learning." In: **Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19**. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 6196–6200.
8. D. Bertsimas, V. F. Farias, and N. Trichakis. "The price of fairness." In: **Operations research** 59.1 (2011), pp. 17–31.
9. C. Joe-Wong, S. Sen, T. Lan, and M. Chiang. "Multiresource allocation: Fairness–efficiency tradeoffs in a unifying framework." In: **IEEE/ACM Transactions on Networking** 21.6 (2013), pp. 1785–1798.
10. D. Bertsimas, V. F. Farias, and N. Trichakis. "On the efficiency-fairness trade-off." In: **Management Science** 58.12 (2012), pp. 2234–2250.
11. A. K. Menon and R. C. Williamson. "The cost of fairness in binary classification." In: **Conference on Fairness, accountability and transparency**. PMLR. 2018, pp. 107–118.
12. R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch. "Multi-agent actor-critic for mixed cooperative-competitive environments." In: **Advances in neural information processing systems** 30 (2017).
13. D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. "Deterministic policy gradient algorithms." In: **International conference on machine learning**. PMLR. 2014, pp. 387–395.
14. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. "Playing Atari with Deep Reinforcement Learning." In: (2013). cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.
15. J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson. "Stabilising experience replay for deep multi-agent reinforcement learning." In: **International conference on machine learning**. PMLR. 2017, pp. 1146–1155.
16. J. Jiang and Z. Lu. "Learning Fairness in Multi-Agent Systems." In: **Proceedings of the 33rd International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2019.
17. S. Ravanbakhsh, J. Schneider, and B. Poczos. "Equivariance through parameter-sharing." In: **International conference on machine learning**. PMLR. 2017, pp. 2892–2901.
18. D. Chen, J. Tachella, and M. E. Davies. "Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements." In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. 2022, pp. 5647–5656.
19. V. G. Satorras, E. Hoogeboom, and M. Welling. "E (n) equivariant graph neural networks." In: **International conference on machine learning**. PMLR. 2021, pp. 9323–9332.
20. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. "Continuous control with deep reinforcement learning." In: **arXiv preprint arXiv:1509.02971** (2015).

8 Appendix - Hyperparameters

This section will focus on describing the different hyperparameters that were used to reproduce the original paper's experiments. The following list summarises the main hyperparameters:

- Learning Rate: The learning rate is defined for both actor and critic networks.
- Hidden Layers: Both actor and critic networks have been implemented through standard Multi Layer Perceptron (MLP) networks with linear layers. We have two hidden layer parameters for each network, allowing us to control their architecture.
- Update steps: This parameter dictates the number of steps between policy updates.
- Gamma: The discount value gamma, used in the Q-Learning loss function.
- World Size: This parameter determines the size of the simulated environment.
- Number of episodes: The number of simulated runs during training.
- Evader speed: The speed at which the evader moves.
- Pursuer speed start and end: During training, the pursuer's speeds are varied linearly, originally going from 0.4 to 1.2 in *Decay* number of episodes. Lower speeds result in a large increase in training time as there is a higher chance the pursuers never catch the evader.
- Decay: This parameter dictates the number of episodes required in training to test all of the different pursuer speeds.
- Distance start and end: This value dictates the distance that the agents could see, due to the reduced world size, this value was also reduced proportionally.
- Number of steps per episode: This parameter determines the limit number of steps in an episode that the pursuers have to catch the evader. After this threshold is reached, the evader is said to have won.
- Replay Buffer Length: The replay buffer used by DDPG to minimize sample correlation and learn from past experiences[20].

Table 1 shows the changes performed in the different hyperparameters.

Hyperparameter	Original Value	Changed Value
Learning Rate (actor)	0.0001	-
Learning Rate (critic)	0.001	-
Hidden Layers (actor)	2 x 128	-
Hidden Layers (critic)	3 x 128	-
Update steps	5	-
Gamma	0.99	-
World Size	6.0	2.0
Number of episodes	125,000	42,000
Evader speed	1.0	-
Pursuer speed start	1.2	-
Pursuer speed end	0.4	0.5
Decay	15,000	5,000
Distance start and end	4.5	1.5
Number of steps per episode	500	167
Replay Buffer Length	500,000	-

Table 1. Hyperparameter values used. The third column denotes the values that were changed from the original study.

9 Appendix - Code-Base

The code for the original paper was structured into two different repositories: https://github.com/ngrupen/multiagent_fairness and https://github.com/ngrupen/multiagent-particle-envs. Our code was placed into the following repository: https://anonymous.4open.science/r/multiagent_fairness_reproducibility-6934