COMPREHENSIVE ROBUSTNESS EVALUATION OF MULTI-AGENT REINFORCEMENT LEARNING ALGORITHMS USING GAUSSIAN NOISE ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, multi-agent reinforcement learning (MARL) has shown its ability in solving sequential decision-making problems in complicated multi-agent environments. However, uncertainties from observations and executions undermine its performance when MARL methods are deployed in real-world applications. While crucial for deployment, a systematic robustness evaluation for MARL algorithms is not present. In this work, we utilize Gaussian noise attacks (GNAs) to examine the robustness of a benchmark MARL algorithm: multi-agent deep deterministic policy gradient (MADDPG). To the best of our knowledge, our work is the first to comprehensively investigate the robustness of MADDPG to GNAs to observation and execution information. Our experiments show that GNA has totally different patterns in observation-wise attacks and execution-wise attacks. Furthermore, there are counter-intuitive insights from the experimental results which could guide researchers in future MARL algorithms design.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has attracted increasing attention in recent years for its capability of solving real-world sequential decision-making problems which involve the interactions of multiple agents in a shared environment, for example, in game playing, traffic management, autonomous driving, and robotics (Mnih et al., 2015; Sallab et al., 2017; He et al., 2022). However, uncertainties from observations and executions may degrade the performance of MARL algorithms and may yield unpleasant results in real-world scenarios (Tessler et al., 2019; Zhang et al., 2021). Different origins of uncertainties, including measurement errors, model errors, operation errors, etc., need to be considered before algorithm deployments. Thus, to ensure that the MARL algorithms are reliable, adaptable, trustworthy, and suitable in a wide range of real-world applications, it is essential to evaluate robustness of MARL algorithms before their deployment. However, there is a lack of comprehensive robustness evaluations for MARL algorithms.

Researches commonly quantify the robustness of machine learning (ML) methods by testing the performance of an ML algorithm after addicting Gaussian noise (a statistical noise with a Gaussian distribution) into the input (Pezzementi et al., 2018; Rauber et al., 2017; Turchetta et al., 2020). This approach, known as **Gaussian noise attack (GNA)**, provides a universal baseline robustness evaluation for ML methods. Therefore, in this work, we systematically and comprehensively evaluate the robustness of a benchmark MARL algorithm (multi-agent deep deterministic policy gradient, i.e., MADDPG (Lowe et al., 2017)) using GNAs. To the best of our knowledge, our work is the first to comprehensively investigate the robustness of the benchmark MARL algorithm to Gaussian noise attacks to observation and execution information. We find that GNA has totally different patterns in the results of observation-wise attacks and execution-wise attacks. MADDPG's performance also highly depends on the multi-agent environment settings: in complicated environments, MADDPG can even achieve better performance under GNA than without attacks. Other counter-intuitive experimental results also help researchers to better design robust MARL algorithms (Yu et al., 2021).

2 METHODOLOGY AND EXPERIMENTS

MADDPG is a benchmark MARL algorithm that works very well in practice for both cooperative and competitive environments (Lowe et al., 2017). We train MADDPG policies for agents in **MPE**

(`https://github.com/openai/multiagent-particle-envs`), a benchmark multi-agent environment. There are 8 multi-agent scenarios in MPE: Mutual communication (MC), Cooperative communication (CC), Cooperative navigation (CN), Physical deception (PD), Encrypted communication (EC), Keep-away (KA), Predator-prey (PP) and Complicated game (CG). Each scenario is a multi-agent game that requires multiple agents to collaborate or/and compete. We provide detailed descriptions of these scenarios in the appendix. The hyper-parameters we use to train our policies are provided in the appendix, table 1. After each step of execution, we collect the reward received by each agent. By the end of the testing, which lasts for 10000 steps, we use agents' mean reward as a metric of the performance of MADDPG. In particular, we set this metric under no noise as our baseline. In the procedure of **Robustness Evaluation**, we inject a series of *i.i.d.* Gaussian noise into either the observation information (input of the policy) or the execution (input of the environment) when testing the well-trained MADDPG policies in each scenario. We use $\mathcal{N}(\mu, \sigma)$ to denote a Gaussian noise with mean μ and standard deviation (std) σ. To comprehensively understand the effect of Gaussian noise attack to MADDPG algorithm, we respectively set $\mu = -3, -2, -1, 0.001, 0.05, 0.1, 0.25, 0.5, 1, 2, 3$ and $\sigma = 3, 2, 1, 0.5, 0.25, 0.1$.

Robustness to observation-wise GNA: Depending on scenarios, robustness of MADDPG to GNA are distinct. See Fig. 1 and Fig. 3 for illustration. In scenarios MC, CC, CN, KA, and CG, under observation-GNA, there is a major decline on the agents' mean reward compared to the baselines (the mean reward of MADDPG under no noise). Interestingly, such decline even occurs under $\mathcal{N}(0, 0.1)$, which is supposed to closely resemble the baseline. Also unexpectedly, instead of decline, scenarios PP, EC and PD witness improvement of the agents' mean reward under GNA with certain parameters. Due to page limits, more experimental results are in Appendix.

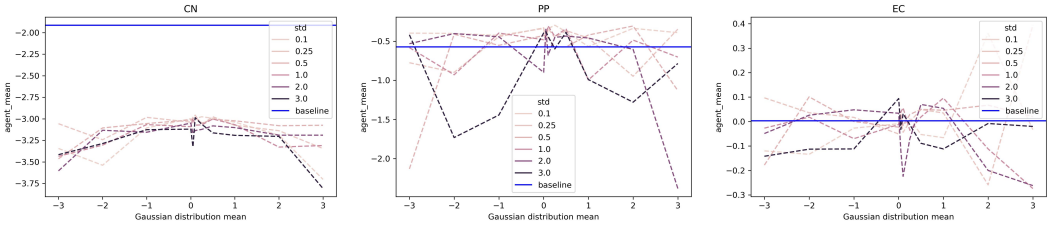

Figure 1: Agents' mean rewards under observation-wise GNA with different μ and σ.

Robustness to execution-wise GNA: In scenarios CN, CC, KA, the agents' mean reward is reluctant to the change of GNAs' mean values and MADDPG is robust to GNAs with small σ, i.e. the mean reward is below but close to that of the baseline. In scenarios PD, PP and CG, GNAs' μ has still limited influences on the agents' mean reward, but σ plays a key role: although small σ degrades the performance of agents as expected, large σ even improves the agents' reward in PD compared to the baseline; moreover, we observe the inverse effect of σ in PP and CG, that is, the agents' reward increases under GNAs with small σ but decreases with large σ. See Fig. 2 and 4 for illustration. More experimental results are in Appendix.

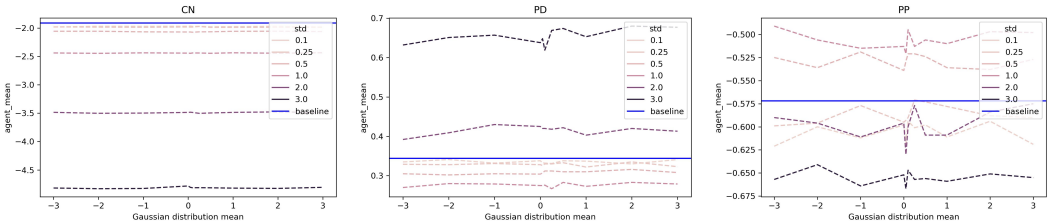

Figure 2: Agents' mean rewards under execution-wise GNA with different μ and σ.

Conclusion and Future Work: Considering that there is limited work regarding robustness evaluation for multi-agent reinforcement learning (MARL) methods, we conduct a systematic robustness analysis for a benchmark MARL algorithm with respective to observation and execution uncertainties, under Gaussian noise attacks. The counter-intuitive phenomena we find in the experiments could potentially help researchers better design robust MARL algorithms. However, as real-world data noise sometimes follows a non-Gaussian distribution, it would be valuable to see other types of noise/attacks in robustness evaluation, which is considered as our future work.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2023 Tiny Papers Track.

REFERENCES

Sihong He, Yue Wang, Shuo Han, Shaofeng Zou, and Fei Miao. A robust and constrained multi-agent reinforcement learning framework for electric vehicle amod systems. *arXiv preprint arXiv:2209.08230*, 2022.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.

Zachary Pezzementi, Trenton Tabor, Samuel Yim, Jonathan K Chang, Bill Drozd, David Guttendorf, Michael Wagner, and Philip Koopman. Putting image manipulations in context: robustness testing for safe perception. In *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 1–8. IEEE, 2018.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017.

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pp. 6215–6224. PMLR, 2019.

Matteo Turchetta, Andreas Krause, and Sebastian Trimpe. Robust model-free reinforcement learning with multi-objective bayesian optimization. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10702–10708. IEEE, 2020.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*, 2021.

A APPENDIX

A.1 MULTI-AGENT ENVIRONMENTS

In this subsection, we introduce the multi-agent environments (Lowe et al., 2017) (Mordatch & Abbeel, 2017) we use in our experiments. By mixed game, we mean a game involving both cooperation and competition.

Mutual Communication (MC) This is a cooperative game in which there are 2 agents, 3 landmarks of different colors. Each agent wants to get to their target landmark, which is known only by the other agent. Reward is collective, so agents have to learn to communicate the goal of the other agent, and navigate to their landmark. This is the same as the Cooperative Communication scenario where both agents are simultaneous speakers and listeners.

Cooperative Communication (CC) This is a cooperative game same as Mutual Communication, except that one agent is the 'speaker' that does not move but observes goal of other agent, and the other agent is the listener who cannot speak, but must navigate to correct landmark.

Cooperative Navigation (CN) This is a cooperative game in which there are 3 agents and 3 landmarks. Agents are rewarded based on how far any agent is from each landmark. Agents are penalized if they collide with other agents. So, agents have to learn to cover all the landmarks while avoiding collisions.

Physical Deception (PD) This is a competitive game in which there are 1 adversary (red), 2 good agents (green), and 2 landmarks. All agents observe position of landmarks and other agents. One landmark is the 'target landmark' (colored green). Good agents rewarded based on how close one of them is to the target landmark, but negatively rewarded if the adversary is close to target landmark. Adversary is rewarded based on how close it is to the target, but it does not know which landmark is the target landmark. So good agents have to learn to 'split up' and cover all landmarks to deceive the adversary.

Encrypted Communication (EC) This is a mixed game in where there are two good agents (alice and bob), one adversary (eve). Alice must sent a private message to bob over a public channel. Alice and bob are rewarded based on how well bob reconstructs the message, but negatively rewarded if eve can reconstruct the message. Alice and bob have a private key (randomly generated at beginning of each episode), which they must learn to use to encrypt the message.

Keep-away (KA) This is a competitive game in where there are 1 agent, 1 adversary and 1 landmark. Agent is rewarded based on distance to landmark. Adversary is rewarded if it is close to the landmark, and if the agent is far from the landmark. So the adversary learns to push agent away from the landmark.

Predator-prey (PP) This is a Predator-prey environment as well as a competitive game. 1 good agents (green) is faster and wants to avoid being hit by other 3 adversaries (red). Adversaries are slower and want to hit the good agent. Obstacles (large black circles) block the way.

Complicated Game (CG) This is a mixed game similar to Predator-prey, except (1) there is food (small blue balls) that the good agents are rewarded for being near, (2) we now have 'forests' that hide agents inside from being seen from outside; (3) there is a 'leader adversary" that can see the agents at all times, and can communicate with the other adversaries to help coordinate the chase.

Table 1: Hyper-parameters

Parameter	**MADDPG**
optimizer	Adam
learning rate	0.01
discount factor	0.95
replay buffer size	10^6
number of hidden layers	2
activation function	Relu
number of hidden unites per layer	64
number of samples per minibatch	1024
target network update coefficient τ	0.01
iteration steps	20
episodes in training	10k
time steps in one episode	25

B MORE EXPERIMENT RESULTS

More discussions on observation-wise GNA: We study how the parameters μ and σ affect the results of GNAs. In scenarios MC, CC, CN, KA, PP and CG, the parameters of observation-wise GNA influence the reward of agents in the following way: when μ is close to zero, the amount of degrade in agents' reward is not much related to μ, but if the mean of noise is strongly biased (*i.e.* with large absolute value) to either positive or negative, the agents' reward further degrades, and it degrades symmetrically. For these scenarios, σ also plays a role in the extent of the degradation, but

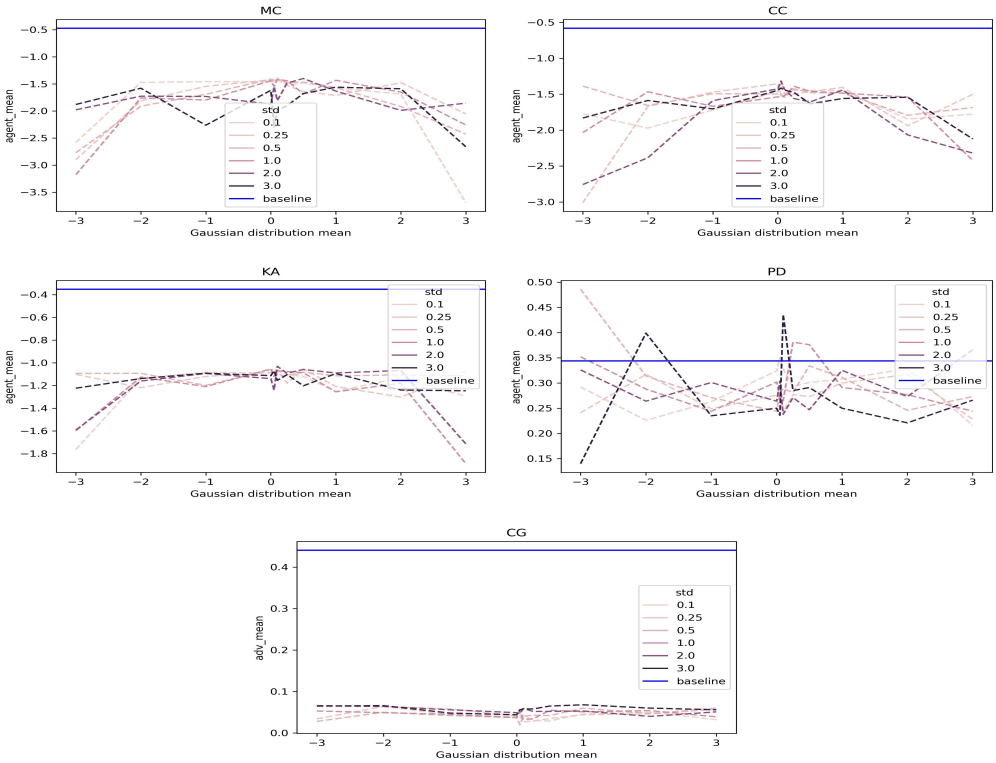

Figure 3: Agents' mean rewards under observation-wise GNA with different μ and σ.

there is no clear pattern (*e.g.*, monotonicity). In scenario EC, the effect of μ is no longer symmetric: the performance of agents has more variation under GNA with $\mu > 0$, and is less sensitive to negative-mean GNA. In scenario PD, GNA affects the performance of agents in an unexpected way: the agents perform even better under a strongly biased noise. See Fig. 3 and previous Fig. 1 for illustration.

More discussions on execution-wise GNA: As for the execution-wise GNA, we also study the effects of different μ and σ. In scenario PP, when σ of GNA is larger than 1, GNA with a larger σ leads to a lower reward. However, when $\sigma < 1$, GNA with a smaller σ leads to a lower reward. In the other 7 scenarios, GNA with a larger standard deviation has more power to undermine MADDPG, i.e. lower rewards.

On the other hand, μ affects agents' mean rewards on a case by case basis. MADDPG is not extremely sensitive to the means of GNAs in scenarios CC, CN, PD, PP and KA. Nevertheless, in scenario MC, the larger $|\mu|$ is, the less reward the MADDPG algorithm can achieve. Remarkably in scenario EC with $\sigma \leq 0.25$, as μ increases to more positive, agents' reward increase. In scenario CG, the value of σ affects the performance of MADDPG dominantly, and the mean of GNA makes a difference only if σ is small. See Fig. 2 and 4.

On the adversary's mean rewards under GNA: For the mixed/competitive scenarios where adversary agents exist, we also check how the adversaries' rewards change under GNAs. See Fig. 5 for the adversary agents' mean rewards under observation-wise GNA and Fig. 6 under execution-wise GNA. First, the observation-wise GNA causes significant decrease in adversary agents' rewards in scenarios PD, EC, PP and CG, but in scenario KA, either decrease or increase could happen depending on the parameters of the GNA. Under execution-wise GNA, in scenario PD and PP, the adversary's rewards is mostly determined by σ. Particularly, if σ is really small, the rewards are close to the baseline, if not better. In scenario EC, the adversary's rewards is decreased with larger $|\mu|$ and larger σ. In scenario KA, a large σ drives adversary agents' reward sensitive to the change of μ, contrasted to what is observed in scenario CG, but in both cases, the performance of MADDPG is worse than the baselines.

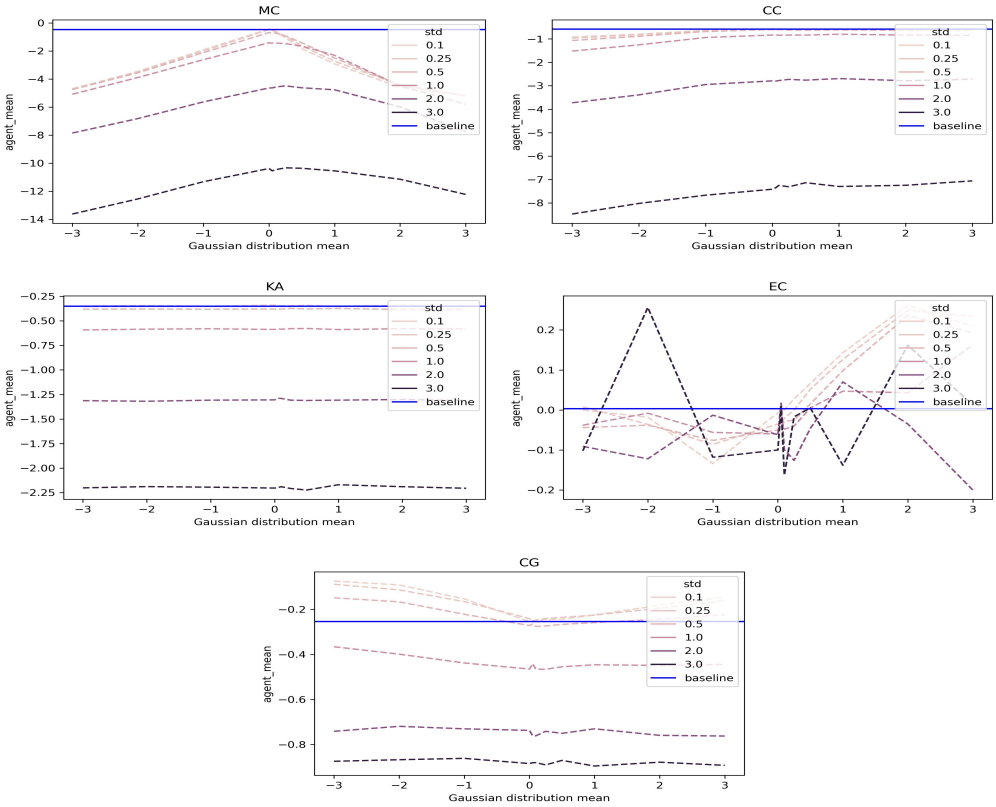

Figure 4: Agents' mean rewards under execution-wise GNA with different μ and σ.

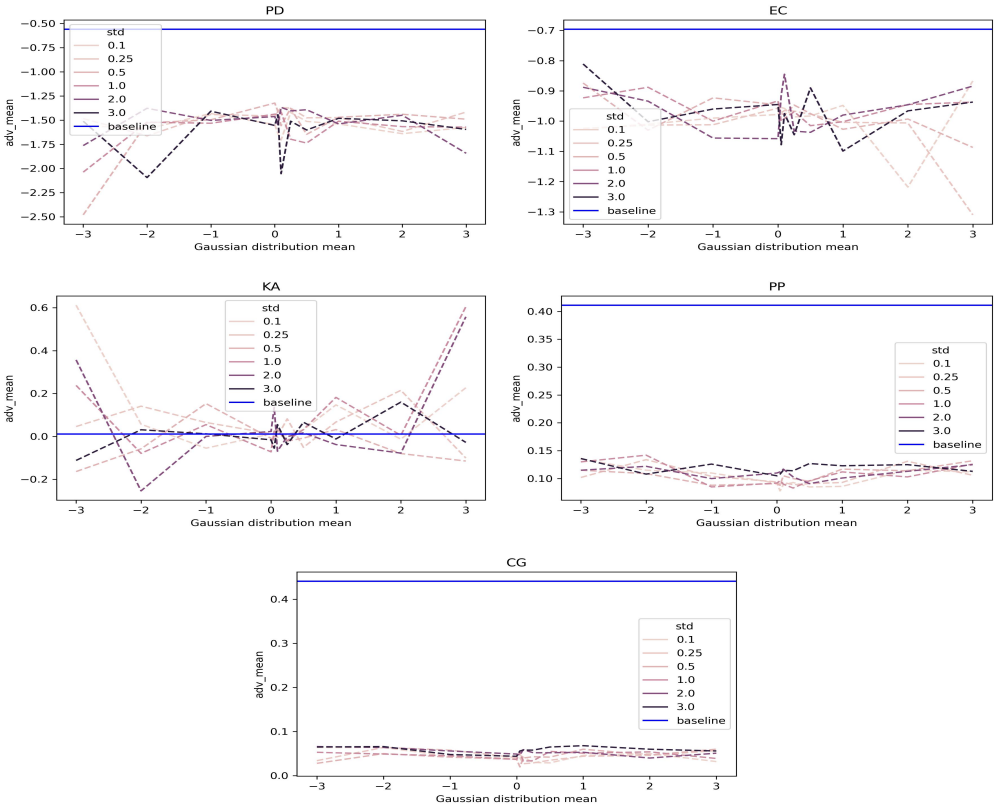

Figure 5: Adversary agents' mean rewards under observation-wise GNA with different μ and σ.

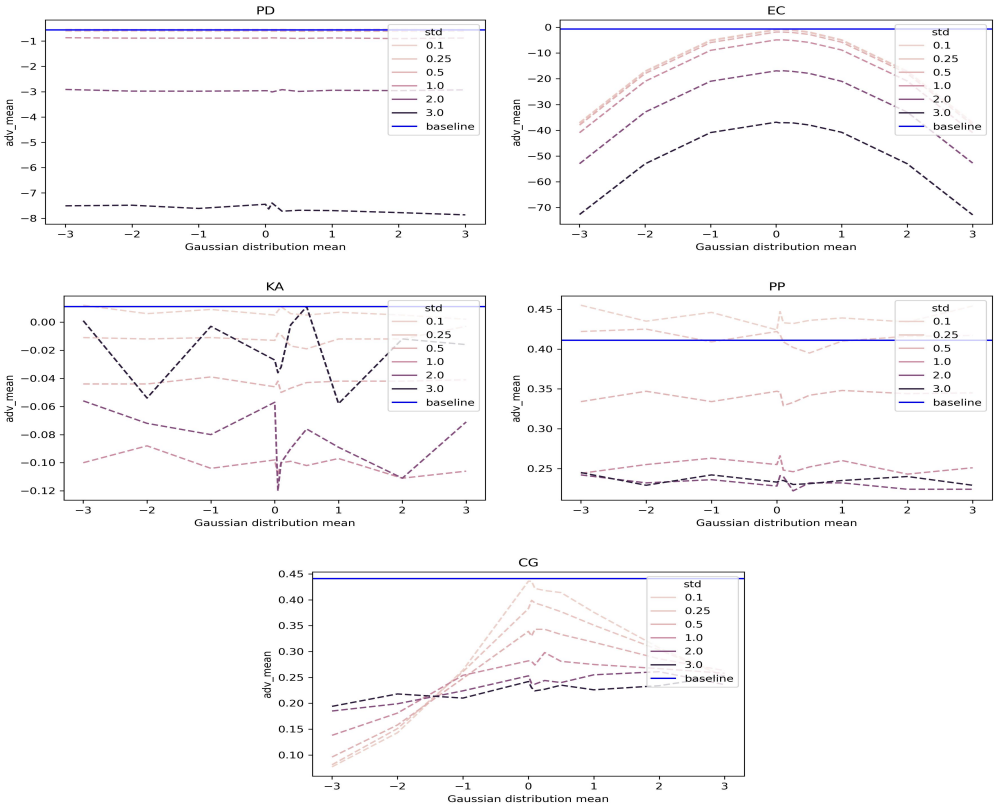

Figure 6: Adversary agents' mean rewards under execution-wise GNA with different μ and σ.