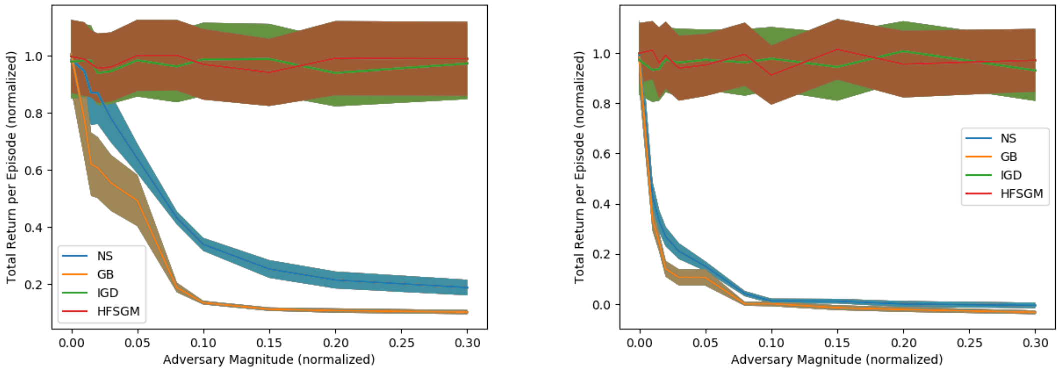
1 Adversarial Attack Results on MuJoCo



(a) Adversarial Attack on Hopper

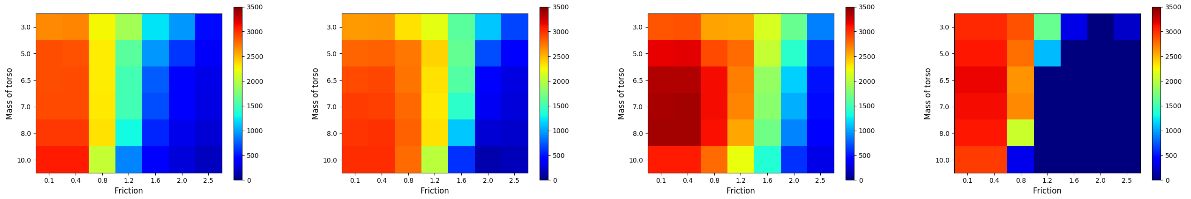
(b) Adversarial Attack on Half-Cheetah

Figure 1: Comparison of different attacks on DDPG for Hopper (1a) and Half Cheetah (1b). We can observe that Gradient Based (GB) performs better than Naive Sampling (NS). HFSGM[10] and Iterative Gadient Descent (IGD) attacks fail in MuJoCo environment.

We performed additional attack experiments and the results are shown in Fig. 1. The experiments showed that GB clearly outperforms NS in Hopper task. In Half-Cheetah environment, GB performs slightly better than NS. IGD and HFSGM are ineffective in these environments. Naive sampling works better than HFSGM and IGD because we explicitly seek adversarial states that lead to decrease in value function of agent (line 8 of Algo. A in supplementary). There is no such check for HFSGM and IGD which affects their performance. The GB performs better than NS as it performs a more directed search along the proposed gradient direction while retaining the adversarial check (line 11 of Algo. 1 in main paper).

2 Training with NS adversary and comparison with "Parameter Space Noise paper"[1]

2.1 Half-Cheetah



(a) "vanilla" DDPG

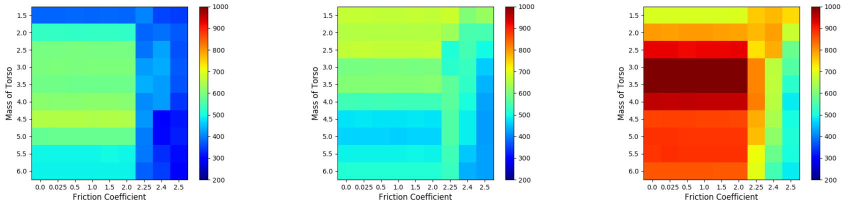
(b) DDPG trained with NS

(c) DDPG trained with GB

(d) DDPG trained with [1]

Figure 2: Comparison of adversarial training with different attacks on DDPG in Half-Cheetah MuJoCo environment. We can observe that DDPG training with Naive Sampling (NS) (Fig. 2b) slightly improves robustness as compared to "vanilla" DDPG (Fig. 2a) (observe columns corresponding to friction=$0.8, 1.2, 1.6$). DDPG training with Gradient Based attack (GB) (Fig. 2c) attack performs better than both NS (Fig. 2b) and "vanilla" DDPG (Fig. 2a). We also performed comparison of our method with [1] (Fig. 2d). We can observe that the agent becomes worse with higher friction values in Fig. 2d while it performs better in lower friction coefficient regime as compared to "vanilla" DDPG in (Fig. 2a). Finally, we observe that DDPG trained on GB (Fig. 2c) outperforms all the methods. This backs the claims made in the paper.

We can observe from Fig. 2 that training based on proposed attack (GB) outperforms all other methods.



(a) "vanilla" DDPG (b) DDPG trained with NS (c) DDPG trained with GB

Figure 3: Comparison of adversarial training with different attacks on DDPG in Hopper MuJoCo environment. We can observe that DDPG training with Naive Sampling (NS) (Fig. 3b) slightly improves robustness as compared to "vanilla" DDPG (Fig. 3a). DDPG training with Gradient Based attack (GB) (Fig. 3c) attack performs better than both NS (Fig. 3b) and "vanilla" DDPG (Fig. 3a). This backs the claims made in the paper.

2.2 Hopper

From Fig. 3, we can observe that the robust training based on GB attack beats robust training based on NS. Please note that we couldn't perform the Hopper experiments for [1] as the default hyperparameters mentioned in the paper ([1]) failed to produce any reasonable policy (received an average return of 345 per episode) on "nominal" parameters.

3 Conclusion

It is interesting to observe that the environments in which GB attack is more effective than NS (Hopper in Fig. 1a as compared to Half-Cheetah in Fig. 1b), we get better performance difference between GB robust training (Hopper in Fig. 3c and Fig. 3b) and NS robust training (Half-Cheetah in Fig. 2c and Fig. 2b). Thus, the robustness is due to adversarial attacks; better is attack, more robust the agent becomes when trained with such an adversary.

References

[1] Plappert Matthias, Houthooft Rein, Dhariwal Prafulla, Sidor Szymon, Chen Richard Y., Chen Xi, Asfour Tamim, Abbeel Pieter, and Andrychowicz Marcin. Parameter space noise for exploration. *International Conference on Learning Representations (ICLR)*, 2018.