General response. Reviewers give great feedback on improving the structure of this paper under space constraints, and we plan to reorganize our paper: (1) Move non-critical theorems and optimization techniques to appendix and leave space for discussions and proof sketches. (2) Include a small running example (as in Appendix A) of SA-MDP. (3) Rephrase any claims that seem too strong, add additional reference and discuss more connections to previous works. (4) Use more plots (like Fig. 11 and 12) (5) Fix typos, format and refine notations.

Table A: Training time

Method/Model	vanilla	SA (PGD/SGLD)	SA (Conv)
DDPG (Reacher)	5.21h	7.10h	6.75h
DDPG (Ant)	6.08h	8.16h	7.70h
PPO (Hopper)	0.57h	1.17h	1.38h
PPO (Walker)	0.61h	1.56h	1.80h
PPO (Humanoid)	4.63h	11.0h	20.3h
DQN (RoadRunner)	15.2h	38.6h	46.5h
DQN (Freeway)	14.9h	44.7h	57.7h

R1. Paper too long We will reorganize our paper (see general response). Std. across training runs In Fig. 11 and 12 (appendix), the rewards are collected from 30 and 11 training runs for PPO and DDPG, respectively. In **Table B**, we train DQN and SA-DQN >5 times. The red lines in bars represent median rewards. We improve reward under attacks consistently across runs. Adversary bounds We use the ℓ_∞ norm based adversary bounds as in many works on attacking Deep RL [20,24,29,42,69]. We vary ϵ bounds in Fig. 9. Critic/Random attacks improve performance The small "improvement" in random attack is just by chance (Fig. 9 is more clear; yellow lines fluctuate). Critic attack sometimes improves PPO performance (green lines of Fig. 9). It is not a bug. In PPO, the critic is a value function $V(s)$ rather than $Q(s, a)$, thus critic attack is applied differently (appendix L676-681): the "attack" searchers a state with the worst value in $B(s)$, and the agent takes the action for the worst case. It is a more conservative action which sometimes prevents the agent from failing and improves performance. Weak adversaries implemented Our proposed robust Sarsa (RS) and MAD adversaries are not weak. From Table 1 and 7, our two new attacks are considerably stronger than the commonly used critic attack. 2nd-order optimization expensive We avoid 2nd order optimization (L180-181). SGLD (L188-196) is a first order method and only requires gradients. The convex relaxation method (L197-207) first produces a relaxed counterpart of the underlying neural network, then uses gradient descent to optimize it. Assumption 2 strange We need this assumption otherwise the adversary can arbitrarily change state and make the problem trivial. Practically it is a norm constraint as in [20,24,29,42,69]. Explain Thm 5 and 6 Following Thm 4 we cannot find an Markovian optimal policy for SA-MDP. Instead, Thm 5 upper bounds the performance loss by regularizing total variation (TV) distance. Thm 6 gives TV distance for DDPG. Thm 3 proof See appendix L616-620. Vanilla DQN performs comparably Vanilla DQN performs comparably only under clean evaluation; it performs poorly under attacks. For Pong, the reward is the lowest possible reward (-21). Table 2 structure and more results Full results for each attack are in appendix Table 7 to save space. Runtime assessment See **Table A**. Ablation study for perturbation budget In Fig. 9, we analyze the agent performance over different perturbation budgets ϵ. Limitations See reply to R2.

R2. We will reorganize our paper as suggested, detailed in our general response. Limitations It is possible to construct an MDP that every nearby state requires a vastly different action, so a typical robustness prior does not hold. In the classification setting, a similar situation is to learn a parity function $f(x) = x_1 \oplus x_2 \cdots \oplus x_n$ ($\oplus$ is XOR) where robustness is impossible. For most realistic problems it's reasonable to assume that a robustness/smoothness prior is valid and helpful. Sum instead of max max represents the strongest adversary; sum or expectation over $B(s)$ is similar to adding random noise with certain distribution. This is a weaker adversary (like random attack in Table 1 and 7).

R3. Related attacks We will enhance the related work section as suggested. Existing attacks rely on the critic learned with the policy. Our MAD and RS attacks do not depend on this critic as using it can be suboptimal (L241-246). Why RS attack better than MAD MAD is myopic and maximizes one step difference without reducing cumulative rewards. RS attack learns a robust *action-value function*, where by definition gives a worst action to reduce cumulative rewards. Safeness specifications We conduct additional experiments on Ant and Humanoid and define the *safe rate* as the percentage that agent does not fall over 50 episodes. Vanilla DDPG (PPO) achieves 56% (2%) safe rate without attacks and 0% (0%) under attack, while SA-DDPG (SA-PPO) achieves 100% (68%) safe rate without attacks and 100% (34%) under attack for Ant (Humanoid, respectively). Partial observability In PO-MDPs, the observation is statistically related to groundtruth state. In SA-MDPs, the observation is an adversarially perturbed state: the adversary is assumed to know the weakness of the policy and can supply the worst-case state, which cannot be directly characterized as conditional observation probabilities in PO-MDP. SAC and TD3 We conduct experiments and find SAC policies are also not robust. SA-SAC significantly improves robustness (**Table C**). We leave model based methods as future work.

R4. Related work We will discuss the connection to smoothing in supervised learning and zero-sum game. We already cited Zhang et al. as [75] and will cite Miyato et al. For RL, not all techniques from supervised learning can be applied directly (line 32-36), so our theory is still valuable. Tighten constant Thank you for pointing this out. One $1/(1-\gamma)$ factor in our bound is to cancel out a $(1-\gamma)$ in the definition of $d^{\pi}_{s_0}$ in (20) in the appendix. Another $1/(1-\gamma)$ factor is from the sum of a geometric sequence. We cannot see an obvious way to tighten it but will keep thinking about it.

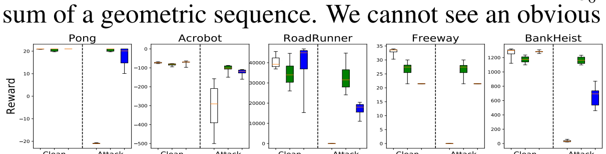

Table B: Box plot to show DQN performance with and without attacks across training runs. We train each setting at least 5 times (DQN training is expensive).

Env.	ϵ	Method	Natural Reward	Best Attack Reward
Hopper	.075	SAC	3494 ± 3	808 ± 42
		SA-SAC	3553 ± 7	1478 ± 220
Walker	.05	SAC	4371 ± 39	1725 ± 1551
		SA-SAC	4126 ± 80	3854 ± 109
Ant	.2	SAC	5236 ± 628	-212 ± 348
		SA-SAC	4728 ± 603	1940 ± 1612

Table C: The median model performance of 11 training runs for SAC