Safer Reinforcement Learning by Going Off-policy: a Benchmark

Igor Kuznetsov [1]

Abstract

Avoiding penalizing safety constraints while learning solvable tasks is the main concern of Safe Reinforcement Learning (SafeRL). Most prior studies focus on solving SafeRL problems with the on-policy algorithms, which obtain stable results at the expense of sample efficiency. In this paper, we study SafeRL from the off-policy perspective. We argue that off-policy RL algorithms are better suited for SafeRL as minimizing the number of samples results in fewer safety penalties. We show that off-policy algorithms achieve better safety metrics for the same performance level than on-policy competitors and provide a benchmark of 6 modern off-policy algorithms tested on 30 environments from the state-of-the-art SafetyGymnasium environment set.

1. Introduction

Reinforcement learning (RL) aims to produce intelligent agents that are able to solve given tasks by a sequence of steps in time (Sutton & Barto, 2018). The field of safe reinforcement learning (SafeRL) introduces an additional layer of complexity by imposing safety constraints that the agent must adhere to. The presented problem is in jointly solving the task and operating safely in a given environment. As RL algorithms are actively deployed in the real world (Wu et al., 2023; Korshunova et al., 2022), safety concerns become increasingly important. For example, controlling a vehicle autonomously by an RL algorithm requires addressing several safety concerns for nearby pedestrians and human-driven vehicles (Muhammad et al., 2020). From another angle, the safety constraints can be usefully exploited by an RL agent itself. Consider the case of learning a real-world robot to interact with rigid objects in an environment. Without implying safety constraints to the dangerous state regions the robot may hurt the hardware by collisions which results in long expensive repairs. The challenge in SafeRL problems is that behaving dangerously is a necessary aspect of the learning process. Simultaneously, our goal is to minimize dangerous behaviors since, in the real world, they may lead to terminal states.

RL algorithms can be generally categorized based on how they utilize samples from the environment. On-policy methods process incoming information from the environment once without reuse. Employing environment parallelization has proven these algorithms to be stable but sample inefficient. In contrast to on-policy solutions, off-policy algorithms propose storing incoming environment samples and continuously reusing them to optimize the policy from past trajectories. Methods operating in an off-policy fashion have been proven to have better sample complexity (Lillicrap et al., 2015). Previous works on SafeRL problems have focused on either on-policy solutions (Achiam et al., 2017; Stooke et al., 2020; Marchesini et al., 2022) or a combination of on-policy and off-policy methods (Sootla et al., 2022; Yang et al., 2021). In recent years the research community advanced in studying safer constraints and algorithms. Nevertheless, SafeRL research currently lacks a unified benchmark for studying off-policy algorithms. This paper argues that off-policy approach is a more suitable choice for SafeRL problems and proposes a unified benchmark for developing and analyzing SafeRL in an off-policy fashion.

In this work, we conduct a benchmark of modern off-policy algorithms on a variety environments from the state-of-the-art SafeRL environments set (Ji et al., 2023). We unify the algorithms using identical evaluation procedures and learning routines, enabling easy future extension and comparison. To facilitate the future development of SafeRL off-policy solutions, we release the library OPRL (https://github.com/schatty/oprl). The library combines the discussed algorithms under the same evaluation routine and provides a frameworks for future off-policy SafeRL research. All learning curves, detailed reports of hyperparameters, and environment versions are available on the associated project website.

[*]Equal contribution [1]Independent Researcher, Tbilisi, Georgia. Correspondence to: Igor Kuznetsov <igorkuznetsov14@gmail.com>.

NextGenAISafety Workshop at the 41st International Conference on Machine Learning, Vienna, Austria, 2024.

2. Background

We consider a constrained reinforcement learning setup (Altman, 1999), in which an agent interacts with an environment $\mathcal{E}$ at discrete time steps aiming to maximize the reward signal while accumulating safety penalty costs. The environment is a Markov Decision Process (MDP) that can be defined as $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \rho, c, \gamma \rangle$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\mathcal{R}$ is a reward function, c is the task cost, ρ is a transition dynamics, and $\gamma \in [0, 1]$ is a discount factor. At time step t the agent receives state $s_t \in \mathcal{S}$ and performs action $a_t \in \mathcal{A}$ according to policy π, a distribution of a given s that leads the agent to the next state s_{t+1} according to the transition probability $\rho(s_{t+1}|s_t, a_t)$. After providing the action to $\mathcal{E}$, the agent receives a reward $r_t \sim \mathcal{R}(s_t, a_t)$. The discounted sum of rewards during the episode is defined as a *return* $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$.

The RL agent aims to find the optimal policy π_θ, with parameters θ, which maximizes the expected return from the initial distribution $J(\theta) = \mathbb{E}_{s_i \sim \rho_\pi, a_i \sim \pi_\theta}[R_0]$. The action-value function Q denotes the expected return when performing action a from the state s following the current policy π:

$$Q^\pi(s, a) = \mathbb{E}_{s_i \sim \rho_\pi, a_i \sim \pi} \left[R_t | s, a \right]. \tag{1}$$

In continuous control problems the actions are real-valued and the policy π_θ can be updated taking the gradient of the expected return $\nabla_\theta J(\theta)$ with deterministic policy gradient algorithm (Silver et al., 2014):

$$\nabla_\theta J(\theta) = \mathbb{E}_{s \sim \rho_\pi} \left[\nabla_a Q^\pi(s, a)|_{a=\pi(s)} \nabla_\theta \pi_\theta(s) \right]. \tag{2}$$

Actor-critic methods are the dominating approach for off-policy RL due to their stability and sample efficiency. Actor-critic models employ two parameterized functions. An actor represents policy π and the critic is an approximation of the Q-function. The critic is updated with temporal difference learning by iteratively minimizing the Bellman error (Watkins & Dayan, 1992):

$$J_Q = \mathbb{E} \left[(Q(s_t, a_t) - (r + \gamma Q(s_{t+1}, a_{t+1})))^2 \right]. \tag{3}$$

and the actor is learned to maximize the current Q function:

$$J_\pi = \mathbb{E} \left[Q(s, \pi(s)) \right]. \tag{4}$$

DDPG algorithm extends the DPG actor-critic method (Silver et al., 2014) for use with deep neural networks (Lillicrap et al., 2015). The proposed deep architecture enables the solution of complex continuous control tasks with a high-dimensional action space. In DDPG, the parameters of the Q-function are adjusted using an additional frozen target network $Q_{\theta'}$ which is updated by a proportion of τ to match the current Q-function $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$

$$J_Q = \mathbb{E} \left[(Q(s_t, a_t) - Q')^2 \right], \tag{5}$$

where

$$Q' = r(s_t, a_t) + \gamma Q_{\theta'}(s_{t+1}, a'), a' \sim \pi_{\theta'}(s_{t+1}). \tag{6}$$

TD3 is an improvement over DDPG algorithm that applies several modifications to increase the stability and performance of DDPG algorithm (Fujimoto et al., 2018). Firstly, it introduces the second critic and proposes applying *min* operation during the calculation of the target Q-value:

$$Q' = r(s_t, a_t) + \gamma \min \left[Q^1_{\theta'}(s_{t+1}, a'), Q^2_{\theta'}(s_{t+1}, a') \right]. \tag{7}$$

The proposed feature mitigates the Q-value overestimation, addressing the observation that Q-function approximation is prone to overestimating the true Q-value (Thrun & Schwartz, 2014). Additional changes include a reduced ratio of policy update with respect to critic update and the application of small noise from a normal distribution to the target Q-value, enhancing regularization.

SAC algorithm utilizes the maximum entropy framework by augmenting the RL objective with an entropy term (Haarnoja et al., 2018). The proposed change improves exploration in continuous action spaces providing better results than previous deterministic approaches.

TQC is the distributional variant of SAC that improves performance on complex continuous control environments (Kuznetsov et al., 2020). TQC uses a distributional representation of Q-value approximation as a set of quantiles and drops the largest quantiles to reduce Q-value overestimation.

REDQ is a variant of SAC that focuses on increasing sample efficiency by implementing the following features: (1) Increasing the ratio of network updates with respect to received environment samples (2) Employing an ensemble of Q-functions (3) Using the random subset of the Q-functions from the ensemble during the *min* operation for Q-value target function update (Chen et al., 2020). As a result, REDQ obtains higher sample efficiency at the cost of computational complexity.

DroQ is the computationally efficient version of REDQ that uses a smaller ensemble of dropout Q-functions (Hiraoka et al., 2021). The proposed Q-functions are equipped with a dropout layer and layer normalization. The suggested architectural improvements doubly increase computational complexity while maintaining comparative sample efficiency.

3. SafeRL with Off-policy Algorithms

In this section, we first introduce the tasks provided by the SafetyGymnasium benchmark (Ji et al., 2023). Then, we describe the evaluation procedure for benchmarking off-policy algorithms.

3.1. SafetyGymnasium Benchmark

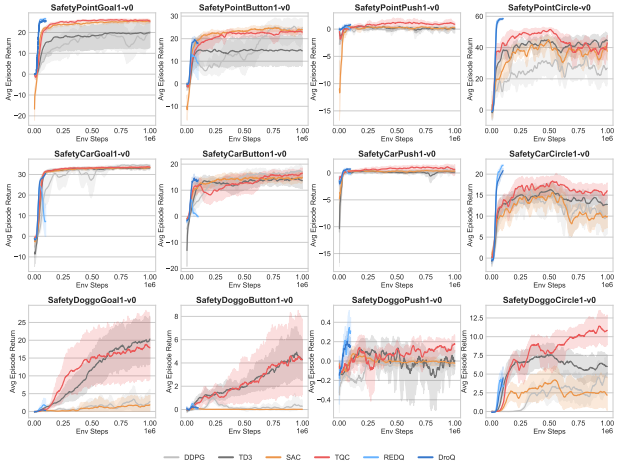

Figure 1: Performance, mean and std of episodic return.

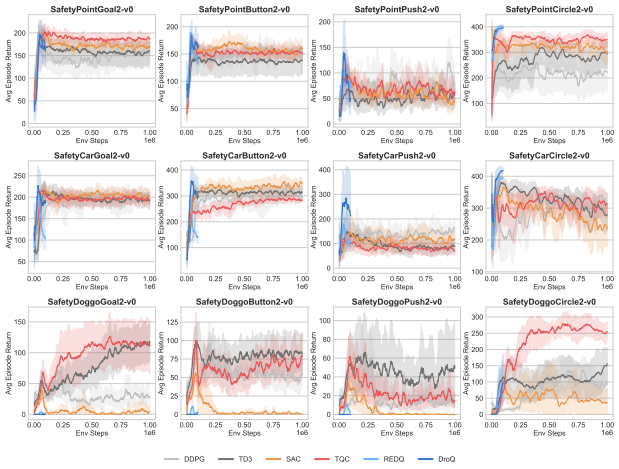

Figure 2: Safety penalty, mean and std of episodic cost.

SafetyGymnasium presents the modern benchmark that encompasses safety-critical tasks in both single and multi-agent scenarios, accepting vector and vision-only input. In this work, we focus on single-agent vector input scenarios for navigation and locomotion. We select 24 representative navigation environments and all available locomotion problems. For the navigation problem, we evaluate three agents: Point, Car, and Doggo on 4 tasks:

- Goal: the robot navigates to multiple goal positions; after reaching the goal, the next goal position is randomly set.
- Button: the objective is to activate a series of goal buttons distributed throughout the environment; after reaching the current button one of the left buttons activated to be the next target.
- Push: the objective is to move a box to a series of goal positions.
- Circle: the reward is maximized by moving along the green circle and penalized for crossing the boundaries that intersect with the circle area.

Each task is presented with two levels of difficulty, that reflect the safety level. The second level presents more hazards and unsafe regions for an agent making it difficult to maintain a low safety cost. The safety penalty is represented as a binary signal. The reward signal is dense for all environments.

For locomotion environments, the reward is given for gaining a velocity with a safety constraint for excess of the speed threshold.

3.2. Evaluation of Off-policy Algorithms

We evaluate 6 modern off-policy model-free RL algorithms on a set of 24 navigation and 6 velocity environments. For DDPG, TD3, SAC, and TQC we run 1M environment steps. For REDQ and DroQ we run 100k environment steps due to the increased computational complexity. However, REDQ and DroQ achieve comparable results on a small number of samples, often exceeding competitors due to the increased sample efficiency.

During the training, we perform policy evaluation every 2e3 steps. At each evaluation, we run the policy on 10 test random seeds with the subsequent averaging. Each training is run 10 times with different training random seeds. For TD3 and DDPG we use the source code from (Fujimoto et al., 2018). For deriving SAC, we extend the original TD3 code with an entropy objective. For REDQ and DroQ we refer to the source codes from (Chen et al., 2020) and (Hiraoka et al., 2021) respectively.

Following the evaluation procedure suggested in (Ray et al., 2019) we report the following metrics. The average episodic return J^R indicates the success of accomplishing the task. The episodic cost J^C shows the average cost at the end of the evaluation episode. This metric is used to show the safety of the final evaluated policy. The cost rate J^{cr} is the total cost accumulated during the training divided by the number of environment samples. This metric indicates the safety level during the optimization, which is crucial for the agents in the real world.

The learning and safety curves during the training for navigation tasks presented in Figures 1 and 2 respectively. Results for locomotion problems and numerical results over the last 10 evaluations are presented in Appendix. It is noteworthy

that for some tasks, the lowest cost is achieved when the agent fails to solve the task completely. If the agent does not move, it has fewer chances to encounter hazards, thereby minimizing its safety penalty. Since we are interested in both solving the task and monitoring the sensible cost rate, we highlight only those metrics for which the agent achieves an episodic reward higher than some performance threshold. We choose a threshold of 5, as agents empirically produce sensible trajectories exceeding this value. The results indicate the following observations:

- Policies that perform strongly typically incur high costs, as they learn to execute longer trajectories, resulting in higher returns but also encountering safety penalties.
- The DDPG algorithm often fails to solve a task with the complex Doggo agent.
- All off-policy algorithms perform poorly in the Push task, where the agent needs to learn to move the given rigid object to the desired position.
- Sample efficient REDQ and DroQ algorithm outperforms other algorithms for Circle task for the simple Point and Car agents while performing comparatively or poorly for the complex Doggo agent.
- For Point and Car agents, the majority of results achieve convergence in terms of episodic return, while for Doggo the return increase continues after 1M time steps.
- For navigation problems, TQC and REDQ achieve the best performance providing the highest episodic reward for 5 out of 24 tasks. For locomotion, TQC shows the highest return, outperforming other algorithms at 3 out of 6 tasks.
- In terms of episodic cost, no algorithm achieves outstanding safety, and the cost metric varies from task to task. Interestingly, the lowest episodic cost does not always match the lowest cost rate.

4. Experiments

In this section, we initially analyze the safety performance of both on-policy and off-policy algorithms, empirically showing that the off-policy approach is better suited for safety-critical tasks. Secondly, we conduct an ablation study for off-policy algorithms examining the impact of common RL hyperparameters on the final agent's performance.

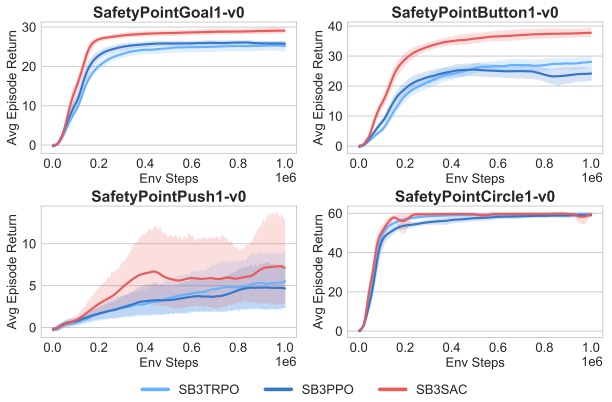

Figure 3: Learning curves of off-policy and on-policy approaches for SafetyPoint tasks.

4.1. Safety Performance of On-policy vs. Off-policy Algorithms

We argue that off-policy algorithms are generally safer than their on-policy competitors due to increased sample efficiency. To illustrate that, we compare the trusted off-policy baseline SAC (Haarnoja et al., 2018) with two common on-policy baselines: PPO (Schulman et al., 2017) and TRPO (Schulman et al., 2015). To ensure a fair comparison, we run the algorithms using the same codebase with an identical evaluation procedure implemented in StableBaselines3 (Raffin et al., 2021). We run 10 random seeds on 1M timesteps for four environments from SafetyGymnasium. Figure 3 shows the learning curves of the algorithms. SAC achieves higher sample efficiency, meaning that it requires fewer samples to achieve the same return than PPO and TRPO. Next, we compare the total cost that algorithms accumulate for three pivotal episodic return points. The points correspond to 10%, 50%, and 100% of minimal return across all algorithms from the final episode. Figure 4 visualizes the total costs gained by the agent at the moment of reaching the pivotal return. Off-policy algorithm incurs less cost penalties compared to the on-policy algorithms.

We acknowledge that employing off-policy algorithms does not eliminate concerns about the safety of RL solutions. However, it decreases the safety penalty due to the fewer numbers of environmental interactions. Therefore, it serves as a more suitable foundation for the development of SafeRL solutions.

4.2. Ablation Study

To better understand the design choices underlying off-policy algorithms for the studied tasks, we conducted an ablation study on common off-policy hyperparameters for the

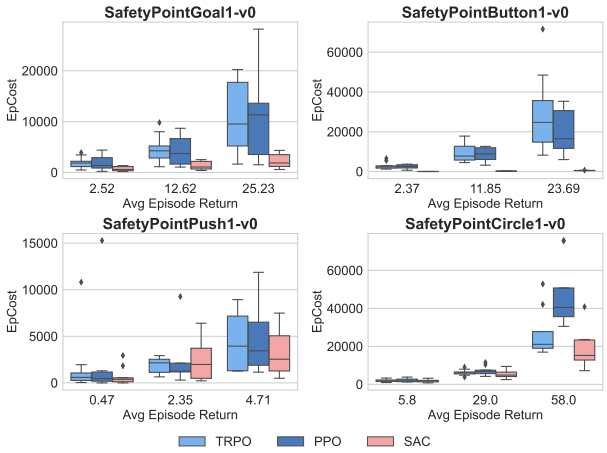

Figure 4: Visualization of cost efficiency of off-policy and on-policy approaches for SafetyPoint tasks.

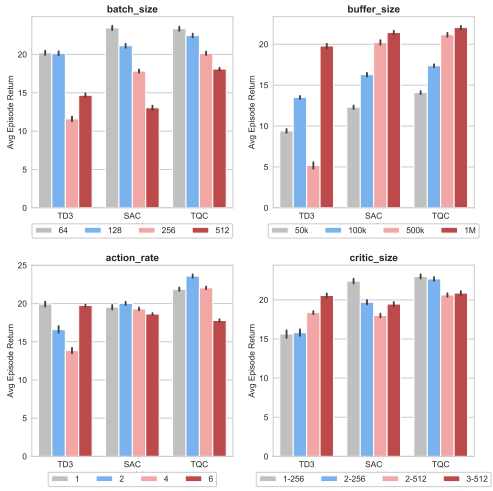

Figure 5: Off-policy algorithms ablation results for SafetyPointButton1-v0. The top left figure shows the effect of the batch size. The top right figure shows the effect of the replay buffer size. The bottom left figure shows the effect of the repeating action rate. The bottom right figure shows the effect of different critic sizes.

TD3, SAC, and TQC algorithms. We chose the task SafetyPointButton1 due to its intermediate complexity compared to other navigation tasks. Figure 5 presents the average episodic return with variations in different hyperparameters: batch size, replay buffer size, action rate, and critic architecture. Each experiment was conducted with 4 random seeds.

The size of the replay buffer can be viewed as a trade-off between on-policy and off-policy algorithms. A smaller replay buffer size results in a greater reliance on recent trajectories during policy optimization. The results suggest that the navigation task benefits from a larger replay buffer size across all algorithms. This implies that optimization with transitions from past policy experiences enhance the agent's performance.

Batch size has proven to be a subtle hyperparameter that frequently influences an agent's performance and computational efficiency (Nikulin et al., 2022). In our ablation study, algorithms do not exhibit significant benefits from large batch sizes across all tested algorithms; instead, the best performance is achieved at batch sizes of 64 and 128.

Repeating the same action for several time steps is a common reinforcement learning technique that has proven useful for tasks not requiring high-precision control (Sharma et al., 2016). However, in our experiments, we found that repeating actions did not prove to be beneficial. The best performance was achieved by using the action only once or twice.

Several works emphasize the importance of critic size in complex continuous control tasks (Kuznetsov et al., 2020; Hansen et al., 2022). We compare critics of different sizes: 1 hidden layer of size 256 (1-256), 2 hidden layers of size 256 (2-256), 2 hidden layers of size 512 (2-512), and 3 hidden layers of size 512 (3-512). For the TD3 algorithm, performance increases with larger critic sizes, while SAC and TQC exhibit their best performance with a 1-layered critic.

5. Discussion

In this work, we present a benchmark for off-policy algorithms applied to SafeRL problems. We address the lack of an in-depth study concerning off-policy SafeRL solutions by analyzing multiple metrics on six modern algorithms across a variety of environments. In this study, we focus on model-free RL algorithms for continuous control that can be applied to both navigation and locomotion tasks. We consider popular baselines such as DDPG, TD3, SAC, and TQC, as well as state-of-the-art sample-efficient approaches REDQ and DroQ. To understand the optimal architectural choice, we conduct an ablation study on common off-policy RL parameters.

The presented study highlights current limitations and future research directions. All tested algorithms struggle to solve the complex Push task. We hypothesize that RL algorithms need more trajectories to generalize to such a complicated problem. Another interesting direction is to change the problem perspective and employ the goal-oriented techniques (Andrychowicz et al., 2017) to improve the density of the reward signal.

Acknowledgements

The authors thank Alexander Nikulin for the helpful guidance about modern reinforcement learning approaches and algorithms used by the scientific community.

Impact Statement

This paper presents work whose goal is to facilitate the field safe intelligent agents. There are no direct potential societal consequences as the paper studies the abstract locomotion and navigation tasks in simulation only.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Altman, E. *onstrained Markov Decision Processes*. Routledge, 1999.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Chen, X., Wang, C., Zhou, Z., and Ross, K. W. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2020.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018.

Hansen, N., Wang, X., and Su, H. Temporal difference learning for model predictive control. In *International Conference on Machine Learning, PMLR*, 2022.

Hiraoka, T., Imagawa, T., Hashimoto, T., Onishi, T., and Tsuruoka, Y. Dropout q-functions for doubly efficient reinforcement learning. In *International Conference on Learning Representations*, 2021.

Ji, J., Zhang, B., Zhou, J., Pan, X., Huang, W., Sun, R., Geng, Y., Zhong, Y., Dai, J., and Yang, Y. Safety gymnasium: A unified safe reinforcement learning benchmark. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Korshunova, M., Huang, N., Capuzzi, S., Radchenko, D. S., Savych, O., Moroz, Y. S., Wells, C. I., Willson, T. M., Tropsha, A., and Isayev, O. Generative and reinforcement learning approaches for the automated de novo design of bioactive compounds. *Communications Chemistry*, 5(1): 129, 2022.

Kuznetsov, A., Shvechikov, P., Grishin, A., and Vetrov, D. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, pp. 5556–5566. PMLR, 2020.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Marchesini, E., Corsi, D., and Farinelli, A. Exploring safer behaviors for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7701–7709, 2022.

Muhammad, K., Ullah, A., Lloret, J., Del Ser, J., and de Albuquerque, V. H. C. Deep learning for safe autonomous driving: Current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems*, 22 (7):4316–4336, 2020.

Nikulin, A., Kurenkov, V., Tarasov, D., Akimov, D., and Kolesnikov, S. Q-ensemble for offline rl: Don't scale the ensemble, scale the batch size. In *3rd Offline RL Workshop: Offline RL as a"Launchpad"*, 2022.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Ray, A., Achiam, J., and Amodei, D. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sharma, S., Lakshminarayanan, A. S., and Ravindran, B. Learning to repeat: Fine grained action repetition for deep

reinforcement learning. In *International Conference on Learning Representations*, 2016.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr, 2014.

Sootla, A., Cowen-Rivers, A. I., Jafferjee, T., Wang, Z., Mguni, D. H., Wang, J., and Ammar, H. Sauté rl: Almost surely safe reinforcement learning using state augmentation. In *International Conference on Machine Learning*, pp. 20423–20443. PMLR, 2022.

Stooke, A., Achiam, J., and Abbeel, P. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pp. 9133–9143. PMLR, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Thrun, S. and Schwartz, A. Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 connectionist models summer school*, pp. 255–263. Psychology Press, 2014.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Wu, P., Escontrela, A., Hafner, D., Abbeel, P., and Goldberg, K. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, pp. 2226–2240. PMLR, 2023.

Yang, Q., Simão, T. D., Tindemans, S. H., and Spaan, M. T. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10639–10646, 2021.

A. Appendix

A.1. Learning and Safety Curves for Locomotion Tasks

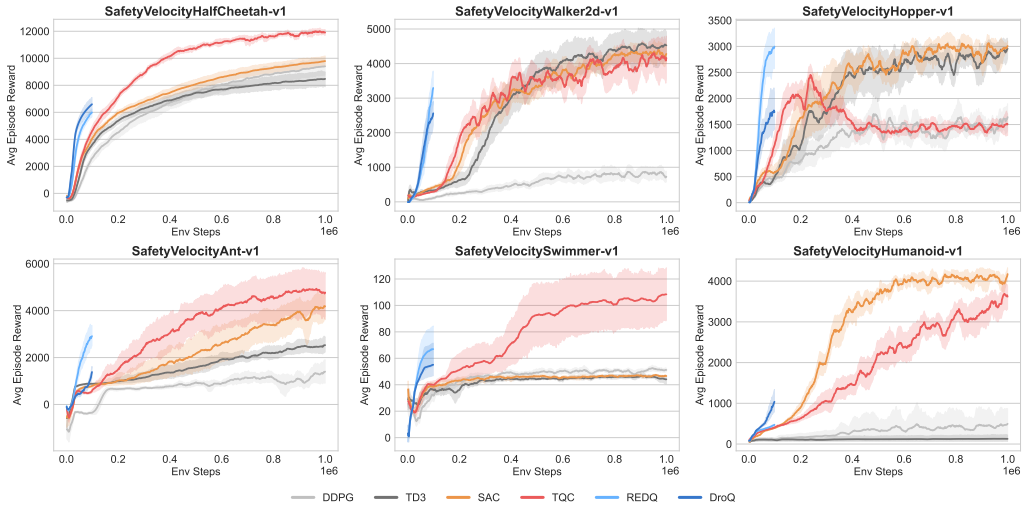

Figure 6: Performance, mean and std of episodic return.

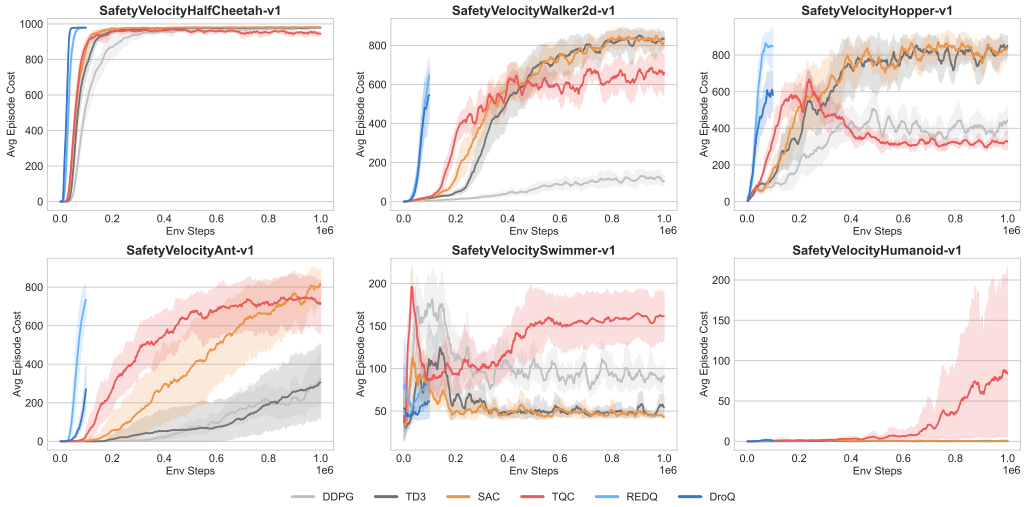

Figure 7: Safety penalty, mean and std of episodic cost.

A.2. Off-policy SafeRL Benchmark

	DDPG			TD3			SAC			TQC			REDQ			DrQ		
Env	J^R	J^C	J^{cr}	J^R	J^C	J^{cr}	J^R	J^C	J^{cr}	J^R	J^C	J^{cr}	J^R	J^C	J^{cr}	J^R	J^C	J^{cr}
PG1	20.61	63.36	0.05	19.87	57.19	**0.04**	25.41	50.36	**0.04**	25.28	54.13	**0.04**	**26.54**	50.68	0.05	25.61	**40.82**	0.05
PB1	**25.02**	137.17	0.13	14.33	**114.12**	**0.09**	24.12	158.46	0.10	22.90	149.31	0.10	8.89	114.67	0.11	17.99	139.78	0.12
PP1	0.16	70.88	0.03	0.25	61.86	**0.02**	0.22	59.07	**0.02**	0.90	78.98	0.03	-0.28	40.94	0.05	**0.96**	**32.24**	0.05
PC1	26.16	**131.03**	**0.28**	44.30	172.69	0.37	36.03	183.72	0.37	45.14	172.53	0.37	58.71	200.26	0.35	**58.76**	206.28	0.37
CG1	**34.56**	58.44	0.05	33.45	62.33	**0.04**	33.39	64.76	0.05	33.45	60.15	0.05	7.26	82.60	0.07	31.20	**57.68**	0.06
CB1	**16.88**	376.90	0.21	13.67	360.45	0.20	15.24	394.67	0.17	16.32	**331.10**	**0.14**	-0.12	109.61	0.14	14.23	340.76	0.24
CP1	0.63	56.57	0.04	0.11	39.58	0.01	0.33	36.98	0.03	0.88	40.12	0.03	-0.37	51.25	0.04	0.73	29.38	0.06
CC1	9.32	149.85	0.37	12.80	178.58	0.45	9.98	**141.22**	0.43	16.41	171.01	0.40	**22.28**	197.07	**0.34**	21.21	211.90	0.37
DG1	3.05	30.45	0.06	**20.25**	63.66	0.05	1.69	12.69	0.04	17.89	**57.85**	0.06	1.79	1.91	0.02	0.30	0.00	0.02
DB1	0.31	30.36	0.10	4.11	68.50	0.17	0.02	0.15	0.06	4.44	63.16	0.12	0.06	0.06	0.01	0.15	0.06	0.01
DP1	-0.05	8.19	0.03	0.03	38.16	0.01	-0.02	0.00	0.03	0.20	15.61	0.03	0.30	7.27	0.02	0.15	0.00	0.00
DC1	5.13	84.43	0.14	6.06	**78.65**	0.24	2.21	20.74	0.14	**10.79**	134.21	0.34	2.73	59.95	0.08	4.62	107.64	0.06
PG2	18.62	**153.53**	**0.13**	20.17	160.92	0.14	22.37	167.93	0.16	23.56	186.60	0.17	**25.27**	166.78	0.16	20.94	155.18	0.16
PB2	21.43	**136.57**	0.14	17.97	139.35	**0.12**	**21.49**	152.49	0.13	21.03	152.33	**0.12**	3.79	129.44	0.13	17.31	169.00	0.15
PP2	0.45	59.67	0.04	0.38	52.42	0.02	0.35	45.89	0.03	0.74	73.73	0.04	0.32	59.47	0.08	1.26	80.20	0.07
PC2	32.45	**216.40**	**0.47**	36.43	293.19	0.62	40.65	305.39	0.70	45.14	352.91	0.75	**58.56**	398.60	0.69	58.20	397.24	0.69
CG2	29.00	209.09	**0.16**	26.90	188.99	**0.16**	28.63	199.67	**0.16**	**29.29**	192.23	0.18	1.43	95.50	0.14	23.54	202.08	0.18
CB2	17.18	315.92	0.21	15.97	311.59	0.22	15.72	349.19	0.21	**18.34**	**280.80**	**0.20**	-0.32	138.65	0.16	13.89	289.14	0.25
CP2	0.39	171.00	0.05	-0.11	92.05	0.04	0.12	128.03	0.03	0.25	71.93	0.05	-0.09	119.60	0.10	0.91	217.08	0.12
CC2	10.98	295.43	0.67	12.30	281.39	0.81	7.06	**221.79**	0.78	13.70	317.38	0.79	**22.28**	398.86	**0.65**	21.15	416.80	0.69
DG2	2.15	29.96	0.05	21.18	119.92	**0.09**	0.06	5.98	0.04	**22.71**	**109.84**	0.12	0.50	0.35	0.01	0.25	0.0	0.01
DB2	1.44	48.77	0.07	**6.79**	**83.97**	**0.10**	0.02	0.23	0.04	3.03	72.99	0.11	0.05	0.01	0.02	0.28	2.02	0.02
DP2	-0.04	7.48	0.04	0.02	46.10	0.02	-0.01	0.0	0.05	0.02	17.15	0.05	0.05	0.24	0.02	0.08	0.00	0.00
DC2	3.34	87.60	0.27	7.29	**164.22**	**0.35**	1.52	258.54	0.19	**8.96**	258.54	0.62	3.79	121.07	0.10	4.14	90.92	0.11
Chth	9407.5	979.2	0.87	8496.3	979.4	0.91	9789.8	979.4	0.92	**11942.8**	**948.0**	0.90	6100.7	978.4	**0.72**	6701.8	979.7	0.82
Wlk	785.4	**115.8**	**0.21**	**4574.2**	841.8	0.61	4330.0	829.6	0.64	4193.3	666.0	0.76	3486.8	700.2	0.38	2842.6	593.12	0.34
Hpr	1560.1	420.2	**0.78**	**3023.4**	868.7	0.90	3011.8	841.1	0.90	1483.7	**323.6**	0.90	3009.2	858.0	0.88	1860.8	650.42	0.86
Ant	1473.7	337.4	0.12	2603.4	**317.9**	0.09	4224.5	820.4	0.43	**4818.6**	724.3	0.71	3077.2	766.0	0.32	1678.5	349.00	**0.08**
Swm	51.8	91.4	0.12	44.0	53.5	**0.06**	47.0	**42.1**	**0.06**	**108.4**	160.4	0.13	67.8	82.4	0.11	55.9	59.98	0.08
Hmn	530.5	1.1	**0.00**	125.8	0.08	**0.00**	**4298.4**	0.10	**0.00**	3521.4	71.7	0.03	486.7	0.77	**0.00**	1122.7	0.80	**0.00**

Table 1: Performance and safety comparison of DDPG, TD3, SAC, REDQ, and DroQ. The navigation environment names are coded with the [A][T][L] format, where [A] is the agent type (**P**oint, **C**ar, **D**oggo), [T] is the task (**G**oal, **B**utton, **P**ush, **C**ircle), and [L] is the level of difficulty. For episodic return J^R the maximum value across all algorithms is highlighted. For the episodic cost J^C and the cost rate J^{cr} the minimum value across algorithms are highlighted.