Benchmarking Safety Performance in Reinforcement Learning with Statistical Metrics

Feng, Lan
lafeng@student.ethz.ch
He, Junzhe
junzhe@student.ethz.ch
Li, Lei
leilil@student.ethz.ch

Abstract

In reinforcement learning (RL) tasks, the agents have to explore their environments to find an optimal policy that maximizes the long-term discounted return based on a designated value function. However, in some real-world situations where the safety of the agent is particularly crucial, (e.g. expensive robot arms; implementation of RL in autonomous driving), safe exploration becomes paramount during the training and testing processes in the real world. In this work, we make three noteworthy contributions to advance the study of safe exploration. First, we introduce two sets of metrics to evaluate safe RL algorithms' performance in the training and testing phases, respectively, which complementarily enhance each other and reveal some details that cannot be caught by either phase individually. Second, in the testing phase, the vast majority of the published results of safe RL benchmarks are usually expressed in terms of point estimates of aggregate performances such as mean and median scores across different tasks, which ignores the statistical uncertainty raised by the limited number of training runs. Given the current status quo, we are the first to include statistical uncertainty in the safe RL algorithm's testing-phase evaluation to avoid discrepancies between the point estimates and the true distributions of data. Last but not least, to enhance the diversity of the evaluated algorithms, we extend our benchmark from model-free to model-based safe RL algorithms, and make some improvements on the algorithms for comparison.

1 Introduction

In safety-constrained problems, a "bad", namely an unsafe policy might cause severe damage to the agent. Therefore, RL algorithms have rarely been implemented under these situations and researchers are paying more attention to Safe RL algorithms. Safe RL can be defined as a process of learning to find optimal policies lowering the probability of "bad" policies that might cause unacceptable damage while maximizing the expectation of the accumulated return during the training and testing processes.[1]. To our understanding, Safe RL has a promising prospect in some safety-critical industries that have been developing prosperously. And this has aroused our interest to explore the field of Safe RL.

In the training process, the performance of a safe RL algorithm can be revealed by the change of costs and rewards across learning steps during training trials. In the paper by the team of Safety Gym[2], several baselines constrained RL algorithms on Safety Gym environments (PPO-Lagrangian, TRPO-Lagrangian, etc.) have been evaluated using some well-designed environment sets and metrics. However, the evaluation methodology presented in this paper is not sufficient to thoroughly compare the performance of all state-of-the-art constrained RL methods. The metrics and test environments

Foundations of Reinforcement Learning HS2021, Department of Computer Science, ETH Zürich.

could be less helpful in evaluating performance in terms of some important factors, for example, the steadiness in training runs.

In the testing process, the algorithm is predominantly evaluated by comparing its final costs and rewards across different tasks. Most of the safe RL benchmarks use point estimates, namely the mean and median of costs and rewards for runs in different tasks, which ignores the statistical uncertainty in results.

Thus, we design two sets of metrics for both the training and testing phases to avoid the aforementioned cons of the existing benchmarks. To improve the diversity and test our benchmark comprehensively, we adopt some improvements on the algorithms we are evaluating. For model-free algorithms, we implement PID-controlled Lagrangian methods [3] for comparison in terms of training stability. For the model-based RL, we improve its performance by re-weighting safe data and unsafe data.

2 Related Work

Benchmarking RL and RL Safety: Various benchmark environments have been proposed to measure progress on different RL problems. Bellemare et al. [4] proposed the Arcade Learning Environment (ALE). Brockman et al. [5] proposed OpenAI Gym, an interface to a wide variety of standard tasks. Tassa et al. [6] proposed the Deepmind Control Suite, a set of high-dimensional physics simulation-based tasks. Ray et al. [2] proposed OpenAI SafetyGym, tools for accelerating safe exploration research.

RL Evaluation: Efron [7] proposed bootstrap method in distribution estimation. Colas [8] et al. believe that evaluating more runs per task is necessary to reduce uncertainty and obtain reliable estimates. Agarwal et al. [9] used statistical tools to account for uncertainty and scrutinize performance evaluations of existing algorithms.

Safe RL Algorithms: There are different kinds of safe RL algorithms. Recently, several constrained model-free RL algorithms have attracted much attention. Ray et al. [2] used Lagrangian approach for constrained optimization. Stooke et al. [10] proposed PID controlled Lagrangian method, which achieves favorable learning dynamics through damping and predictive measures. As for model-based safe RL, Liu et al. [11] used model predictive control (MPC) as the basic control framework and proposed a robust cross-entropy (RCE) method that takes into account model uncertainty and constraints to optimize the control sequence, and achieves better constraint satisfaction than baseline safe RL methods.

3 Methods/Algorithms

3.1 Training Evaluation Metrics

To evaluate an algorithm's safety performance, we adopt the metrics proposed by the OpenAI Safety Gym team and measure the following throughout training:

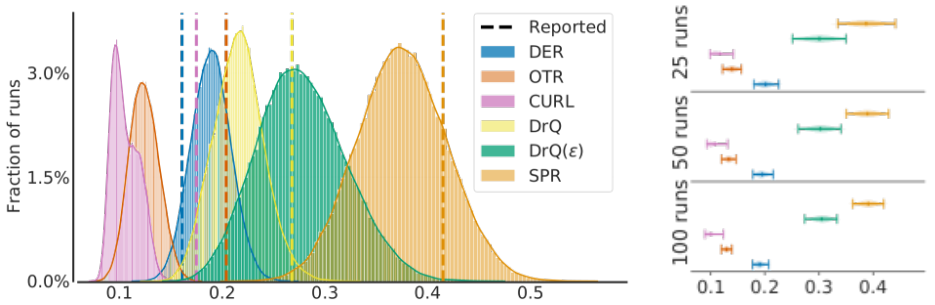
- The averaged return in each episode $J_r(\theta)$.
- The averaged sum of costs in each episode, $J_c(\theta)$.
- The sum of all costs divided by the total number of environment interaction steps, ρ_c. Namely the cost rate.

According to Safety Gym[2], it complies with our intuition that the cost rate does not indicate the stability of the cost curve. For instance, a steady training run could have a similar cost rate compared to a run with high-amplitude oscillations in cost signal as long as the costs in both cases oscillate around a specific average. To better evaluate the performance of the constrained RL methods especially in terms of stability, we introduce a new metric, the standard deviation of the episodic costs, to account for the steadiness of the cost curve.

- The standard deviation of episodic costs, σ_c This quantity describes the stability of the training process and we aim to keep it small and steady.

3.2 Testing Evaluation Metrics

Point estimates are widely used to evaluate the performance of RL algorithms. However, the uncertainty in results is not negligible. There's a substantial discrepancy between point estimates and the sampling distribution.



(a) Distribution of normalized scores

(b) 95% Confidence Interval

Figure 1: **[9] Left. Distribution of median normalized scores** computed using 100,000 different sets of N runs subsampled uniformly with replacement from 100 runs. **Right. 95% CIs** for the IQM scores for varying N.

As shown in Figure 1a, for a given algorithm, the sampling distribution indicates that the median scores estimated using different sets of runs vary across different experiments. However, published point estimates of median scores (shown as dashed lines) fail to show the variability in median scores and fall quite far away from the expected median. Thus, we use two novel metrics, Stratified Bootstrap Confidence Interval [12] (Figure 4) and the Performance Profile [13] (Figure 3) to account for the uncertainty and variability in the test results.

3.2.1 Stratified Bootstrap Confidence Interval

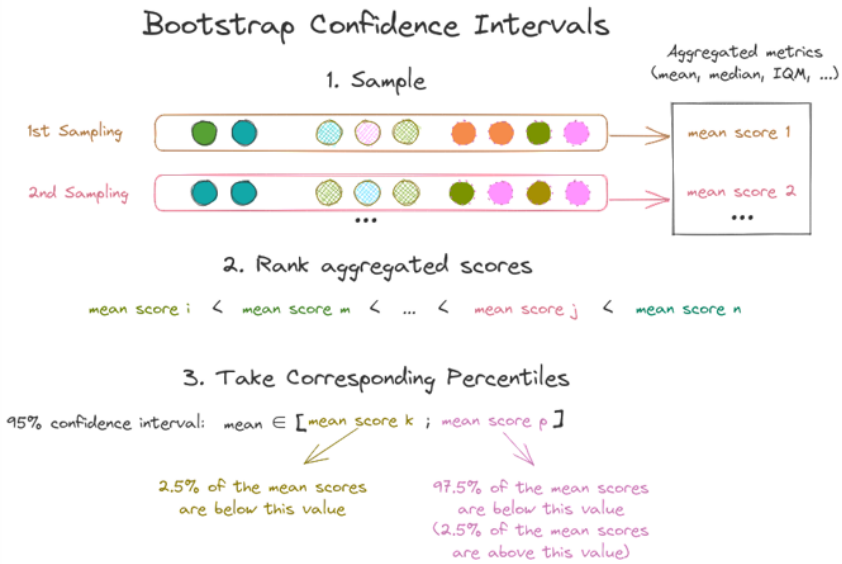


Figure 2: Stratified Bootstrap

Unlike a point estimate which directly approximates the algorithm's aggregate performance, namely the true score, a confidence interval (CI) indicates the range where the true score possibly lies. Given the status quo where we deal with a small sample size, a bootstrap confidence interval seems to be more reasonable than using sample standard deviations to find the CI. The Bootstrapping approach

approximates sampling from the true distribution by random resampling with replacement. Thus we can approximate the confidence interval by resampling a data-set with a small sample size for a given task. Furthermore, instead of standard bootstrapping by re-sampling N runs with replacement independently for a single task, we can aggregate across tasks and create a stratified bootstrap sampling set with M(tasks) times N(runs) data points (as shown in Figure 2). Then we calculate the interquartile mean(IQM) of the sampling set. After repeating this process over 1000 times, we get a fair approximation of the sampling distribution. By sorting out the resulting statistics and cutting off the first and last 2.5 %, we acquire the 95 % Confidence Interval, as shown in Figure 1b. According to Agarwal, B et al. , percentile CIs provide good interval estimates for as few as N = 10 runs for IQM scores.

3.2.2 Performance Profile

For a safe RL algorithm, the performance may vary widely across different tasks and may be outlier-prone. Under this context, an algorithm's performance cannot be fully captured by either point estimates or interval estimates. In comparison to mean and standard deviation, performance profiles provide a much clearer indication of the degree of performance variability across tasks by visibly representing the score distribution for each element in the sampling set (Figure 3), which shows the fraction of runs whose normalized scores are above a certain threshold (τ) and is given by:

$$\hat{F}_X(\tau) = \hat{F}(\tau; x_{1:M,1:N}) = \frac{1}{M}\sum_{m=1}^{M}\hat{F}_m(\tau) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N}[\mathbb{1}x_{m,n} > \tau] \tag{1}$$

where $x_{m,n}$ is the score of run n in task m, M the number of tasks and N the number of runs for each task.

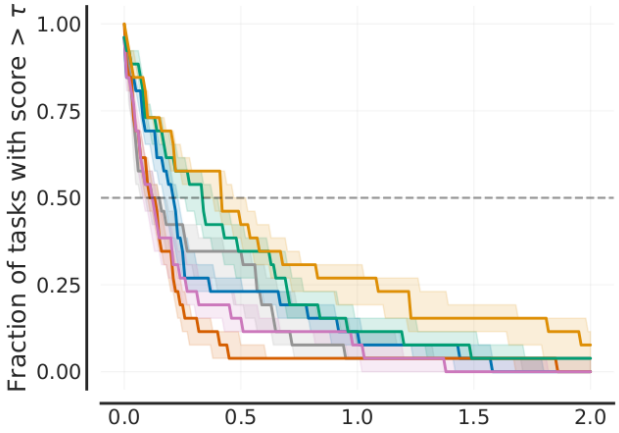


Figure 3: Performance Profile

3.3 Baseline Model-free Methods

The base method (PPO) is a classical RL algorithm without any constraints. In constrained optimization, the errors of gradient and Hessian matrix estimation may lead to poor performance on constraint satisfaction. Lagrangian methods are a classical approach to solving constrained optimization problems, usually multiplying the constraint term (cost function) $g(x)$ by a Lagrangian multiplier λ added to the original objective function. Generally, the objective function of Lagrangian methods is:

$$L(x, \lambda) = f(x) + \lambda g(x) \tag{2}$$

In this way, the Lagrangian approach incorporates the constraints into the objective function with a Lagrange multiplier, and it can be optimized with policy parameters using gradient descent, which avoids constrained optimization. One problem with the Lagrangian method is that during training, constraints are often violated. Violation of the constraint leads to fluctuations in cost function value, which leads to fluctuations in Lagrange multiplier λ. This is typical caused by integral control[3]. Following this perspective, we also adapt the PID Lagrangian method to update the Lagrange multiplier λ in a dynamic way based on the PID.

In summary, for model-free methods, we reproduce and benchmark PPO, PPO-Lagrangian, TRPO-Lagrangian, and PPO-PID-Lagrangian.

3.4 Model-based Method

To give our metrics a more comprehensive evaluation of safe RL, we introduce an improved Model-based safe RL Algorithm. Here, we use two multilayer perceptrons (MLP) to learn the dynamics of the input data and the cost model and obtain a higher accuracy for the dynamics model and cost model. Then, following the approach used in [11], we use model predictive control (MPC) as the basic control framework for our model-based approach, and the goal of MPC is to maximize the cumulative reward over a sequence of actions.

Besides, based on our Model-free training results, we found that after the RL algorithm learned some policies, the unsafe data that violates the safety constraint represent only a small fraction during training. However, the unsafe data is valuable for improving the performance of the RL policy. The algorithms usually get bottlenecked, which means that the effectiveness of the algorithms is difficult to improve at a later stage. Therefore, we want to use a sampling strategy to improve the performance of the RL algorithm by reweighing the distribution of unsafe and safe data. The resampling rule is to sample the unsafe data repeatedly. Instead of selecting each example from the training set with equal probability, we resample more difficult samples. Then, we retrain our two MLPs on our resampled data.

4 Experiments/Results/Discussion

4.1 Training Evaluation

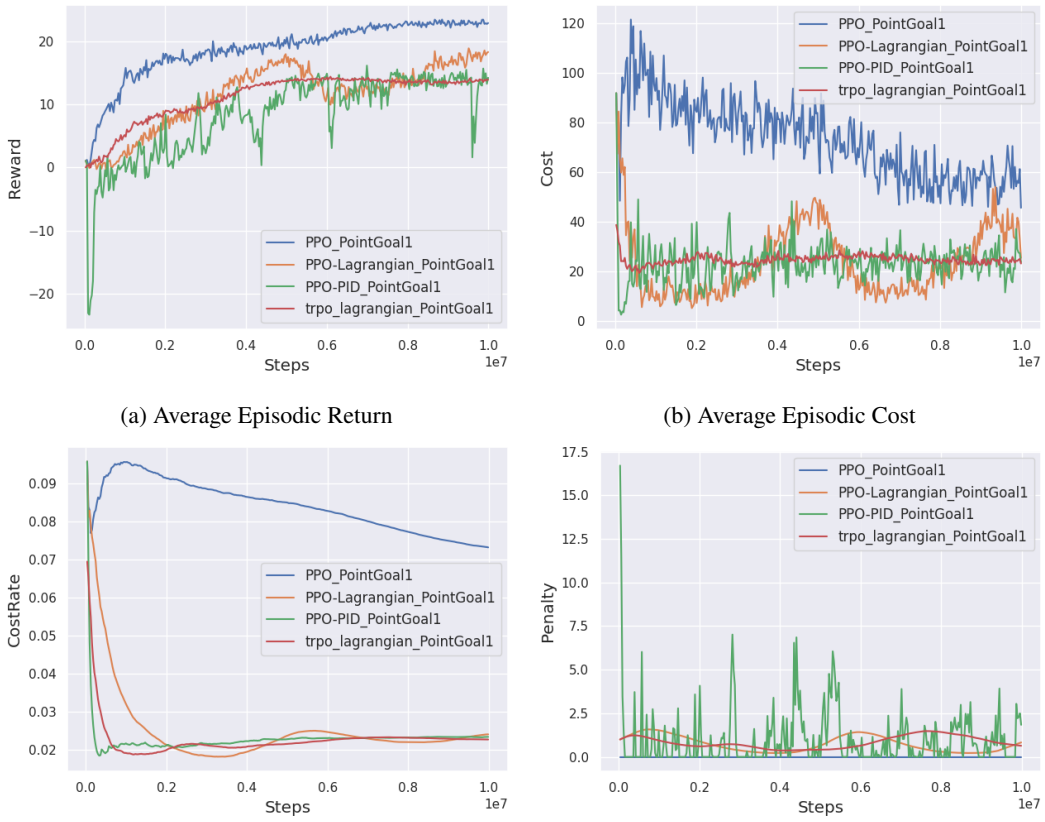
We use normalized metrics to compare the performance across different environments:

$$\begin{aligned} \bar{J}_r(\theta) &= \frac{J_r(\theta)}{J_r^k}, \qquad \bar{\rho}_c(\theta) = \frac{\rho_c(\theta)}{\rho_c^k}, \qquad \bar{\sigma}_c(\theta) \quad = \frac{\sigma_c(\theta)}{\sigma_c^k} \\ \bar{M}_c(\theta) &= \frac{\max(0, J_c(\theta) - d)}{\max(\epsilon, J_c^k - d)} \qquad \epsilon = 10^{-6} \end{aligned} \tag{3}$$

where d is a hyperparameter, for example, the safe threshold, and $J_r^k, J_c^k, \rho_c^k, \sigma_c^k$ are selected characteristic metrics, which were obtained from experimental data of the unconstrained PPO implementation, for environment set k. In Figure 4, we show the learning curves from training the aforementioned RL algorithms on a pre-defined Safety Gym environments. These learning curves show the un-normalized metrics $J_r(\theta)$, $J_c(\theta)$, $\rho_c(\theta)$, and the penalty. In Table 1, we report averaged normalized metrics from the end of training across different test environments.

Table 1: Averaged normalized metrics from the conclusion of training.

	Return $J_r(\theta)$	Violation $M_c(\theta)$	Cost Rate $\bar{\rho}_c(\theta)$	STDEV $\bar{\sigma}_c(\theta)$
PPO	1.0	1.0	1.0	1.0
PPO-Lagrangian	0.732	0.441	0.443	0.891
TRPO-Lagrangian	0.587	0.108	0.258	0.367
PPO-PID-Lagrangian	0.598	0.142	0.312	0.632



(a) Average Episodic Return

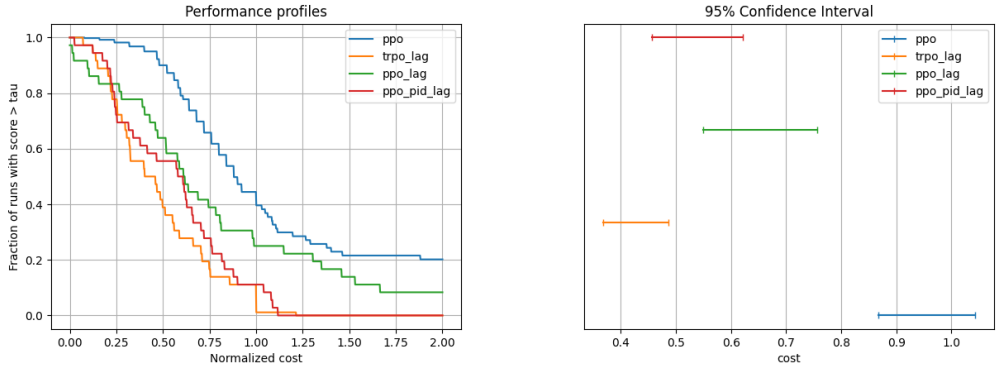
(b) Average Episodic Cost

(c) Cost Rate

(d) Penalty

Figure 4: Results from unconstrained and constrained RL algorithms on a pre-defined environment. We set cost-limit=25 for Lagrangian methods, and test three RL algorithms on PointGoal1 in safety gym.

4.2 Testing Evaluation



(a)

(b)

Figure 5: **Left. Performance Profiles** for 6 tasks and 6 runs for each task calculated with $\tau = 2$. **Right. Stratified Bootstrap 95% Confidence Interval** with 1000 re-sampling trials.

From Figure 4b, we can see that the cost of PID-Lagrangian fluctuates around the cost limit, while there's a more significant periodic fluctuation for the Lagrangian method, which indicates that the

PPO-PID-Lagrangian method better meets the safety requirement (basically near the cost limit). A similar trend can be observed from Figure 4c, in which the cost rate of PPO-PID-Lagrangian drops significantly quicker than other methods. This is because the PID controller tends to drag the cost curve towards a given threshold. To evaluate this characteristic, as mentioned in section 3.1, we introduced a new metric σ_c, the standard deviation of episodic cost. As shown in Table 1, because of better penalty control, the PID-Lagrangian method has a smaller standard deviation compared to the Lagrangian method, which, put another way, means that it is stabler and more preferable. At the same time, the PID-Lagrangian method has a lower Violation, which indicates that this method provides a much safer policy than the Lagrangian method as we expected. Overall, the TRPO-Lagrangian outperforms all other algorithms with respect to safety concerns. However, it seems to perform the worst in terms of return. But this is acceptable as we put more weight on the safety constraints.

In the testing phase evaluation, through performance profile and confidence interval, we have a more comprehensive understanding of the performance of these methods. For example, we can see from the tail of the performance profile that PPO-PID-Lagrangian and TRPO-Lagrangian perform the best in the worst cases since they reach the x-axis faster, while TRPO-Lagrangian performs the best on average cost indicated by the confidence interval. This complies with what we get during the training phase, where the Violation for TRPO-Lagrangian and PPO-PID-Lagrangian are both at a low level and the cost rate for TRPO-Lagrangian is the lowest among all.

4.3 Comprehensive Evaluation of Model-based and Model-free Method

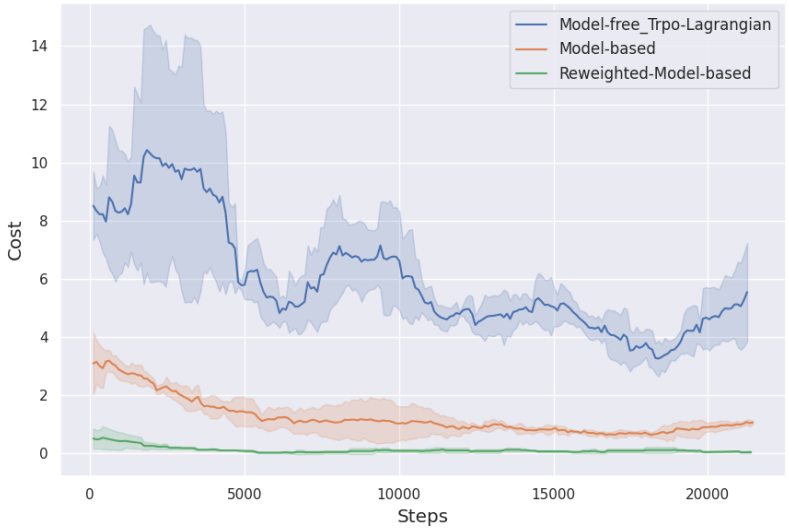


Figure 6: Cost return of model-based and model-free method

We compare the approach of our improved Model-based, the baseline Model-based approach and the best-performing Model-free method, TRPO-Lagrangian. Compared with the baseline Model-based approach, our improved model-based approach uses two MLPs to fit the cost function and the dynamic model, which is able to learn them accurately and faster. As shown in Figure 6, our method avoids unsafe behavior during exploration and achieves the lowest constraint violation. However, the model-based model is more difficult to train and takes a longer time in the same training step compared to the Model-free approach.

5 Conclusion

In this work, we make the following contributions:

- We design two sets of metrics to evaluate safe RL algorithms' performance in both the training and testing phases.

- We are the first to evaluate RL algotihms' safety performance with statistical tools to avoid discrepancies between the point estimates and the true distributions of data.
- We are the first to compare three different genres of safe RL algorithms, including Lagrangian approach, PID Lagrangian approach, and model-based method.
- Our experiment reveals that TRPO Lagrangian and PPO Lagrangian have better worst-case performances than other baseline methods, which is not reported in [2].
- We apply reweighing methods on model-based RL, making it able to learn the cost function accurately and faster, greatly avoiding unsafe behavior during exploration, and achieving the lowest constraint violation rate compared to all other methods.

The current safety gym only contains five different kinds of hazard objects, including dangerous areas and fragile objects, etc, which we believe are not enough for a comprehensive safety evaluation. As a result, we plan to add more diverse environments in safety gym in the future work. For example, adversarial objects that have a certain level of intelligence.

6 Contributions

Lan focuses on code reproduction and comparison of Model-free methods, Junzhe focuses on new metrics realization and performance analysis, and Lei focuses on Model-based testing. We hereby acknowledge all contributions from each team member, the help from our assigned TA Armengol Urpi Nuria, all other TAs, and the professor of introduction to Reinforcement Learning, Niao He, without which we cannot reach this point and finish this project successfully. Every member of this team thinks diligently and communicates efficiently and we are looking forward to the next collaboration opportunity.

References

[1] Javier García, Fern, and o Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015. URL `http://jmlr.org/papers/v16/garcia15a.html`.

[2] Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7, 2019.

[3] Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by PID lagrangian methods. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9133–9143. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/stooke20a.html`.

[4] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[7] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[8] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

[9] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pages 9133–9143. PMLR, 2020.

[11] Baiming Chen Sicheng Zhong Martial Hebert Ding Zhao Zuxin Liu, Hongyi Zhou. Safe model-based reinforcement learning with robust cross-entropy method, 2020.

[12] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7 (1):1 – 26, 1979. doi: 10.1214/aos/1176344552. URL `https://doi.org/10.1214/aos/1176344552`.

[13] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *CoRR*, cs.MS/0102001, 2001. URL `https://arxiv.org/abs/cs/0102001`.