Penalized Proximal Policy Optimization for Safe Reinforcement Learning

Linrui Zhang\textsuperscript{1*}, Li Shen\textsuperscript{2†}, Long Yang\textsuperscript{3}, Shixiang Chen\textsuperscript{2}
Xueqian Wang\textsuperscript{1}, Bo Yuan\textsuperscript{1}, Dacheng Tao\textsuperscript{2}
\textsuperscript{1}Tsinghua University
\textsuperscript{2}JD Explore Academy
\textsuperscript{3}Peking University
zlr20@mails.tsinghua.edu.cn, \{mathshenli, chenshxiang, dacheng.tao\}@gmail.com, yanglong001@pku.edu.cn, boyuan@ieee.org, wang.xq@sz.tsinghua.edu.cn

Abstract
Safe reinforcement learning aims to learn the optimal policy while satisfying safety constraints, which is essential in real-world applications. However, current algorithms still struggle for efficient policy updates with hard constraint satisfaction. In this paper, we propose Penalized Proximal Policy Optimization (P3O), which solves the cumbersome constrained policy iteration via a single minimization of an equivalent unconstrained problem. Specifically, P3O utilizes a simple-yet-effective penalty function to eliminate cost constraints and removes the trust-region constraint by the clipped surrogate objective. We theoretically prove the exactness of the proposed method with a finite penalty factor and provide a worst-case analysis for approximate error when evaluated on sample trajectories. Moreover, we extend P3O to more challenging multi-constraint and multi-agent scenarios which are less studied in previous work. Extensive experiments show that P3O outperforms state-of-the-art algorithms with respect to both reward improvement and constraint satisfaction on a set of constrained locomotive tasks.

1 Introduction
Reinforcement Learning (RL) has achieved significant successes in playing video games [Vinyals et al., 2019], robotic manipulation [Levine et al., 2016], mastering Go [Silver et al., 2017], etc. However, standard RL merely maximizes cumulative rewards, which may lead to undesirable behaviors in real-world applications especially for safety-critical tasks.

It is intractable to learn reasonable policies by directly penalizing unsafe interactions onto the reward function, since varied intensities of the punishment result in different Markov decision processes. There is still a lack of theories revealing the explicit relationship between policy improvement and safety satisfaction via reward shaping.

\textsuperscript{1}This work was done during Linrui Zhang's internship at JD Explore Academy, Beijing, China.
\textsuperscript{2}Corresponding Author.

Constrained Markov decision process (CMDP) [Altman, 1999] is a more practical and popular formulation that requires the agent to perform actions under given constraints. A common approach to solving such constrained sequential optimization is approximating the non-convex constraint function to a quadratic optimization problem via Taylor’s formulation [Achiam et al., 2017; Yang et al., 2020]. However, those algorithms still have the following drawbacks: (1) The convex approximation to a non-convex policy optimization results in non-negligible approximate errors and only learns near constraint-satisfying policies; (2) The closed-form solution to the primal problem involves the inversion of Hessian matrix, which is computationally expensive in large CMDPs with deep neural networks; (3) If there is more than one constraint, it is cumbersome to obtain the analytical solution without an extra inner-loop optimization, which limits their applications in multi-constraint and multi-agent scenarios; (4) If the primal problem is infeasible under certain initial policies, above algorithms require additional interactions for feasible recovery, which reduces their sample-efficiency.

To address above issues, we propose Penalized Proximal Policy Optimization (P3O) algorithm that contains two key techniques. Firstly, we employ exact penalty functions to derive an equivalent unconstrained optimization problem that is naturally compatible with multiple constraints and arbitrary initial policies. Secondly, we extend the clipped surrogate objective in Proximal Policy Optimization (PPO) [Schulman et al., 2017] to CMDPs, which eliminates the trust-region constraint on both reward term and cost term. As a consequence, our method removes the need for quadratic approximation and Hessian matrix inversion. By minimizing the unconstrained objective instead, the solution to the primal constrained problem can be obtained.

Conclusively, the proposed P3O algorithm has three main strengths: (1) accuracy: P3O algorithm is accurate and sample-efficient with first-order optimization over a theoretically equivalent unconstrained objective instead of solving an approximate quadratic optimization; (2) feasibility: P3O algorithm could admit arbitrary initialization with a consistent objective function and doesn’t need additional recovery methods for infeasible policies; (3) scalability: P3O is naturally scalable to multiple constraints or agents without increasing complexity due to the exact penalty reformulation.

In the end, we summarize our contributions as four-fold:
We propose the Penalized Proximal Policy Optimization (P3O) algorithm for safe RL, which employs first-order optimization over an equivalent unconstrained objective. We theoretically prove the exactness of the penalized method when the penalty factor is sufficiently large (doesn’t have to go towards positive infinity) and propose an adaptive penalty factor tuning algorithm. We extend clipped surrogate objectives and advantage normalization tricks to CMDPs, which are easy to implement and enable a fixed penalty factor for general good results across different tasks. We conduct extensive experiments to show that P3O outperforms several state-of-the-art algorithms with respect to both reward improvement and constraint satisfaction and demonstrate its efficacy in more difficult multi-constraint and multi-agent scenarios which are less studied in previous safe RL algorithms.

2 Related Work

Safety in reinforcement learning is a challenging topic formally raised by García and Fernández [2015]. Readers can refer to the survey [Liu et al., 2021] for recent advances in safe RL. In this section, we only summarize the most related studies to our algorithm.

Primal-Dual solution. Considering the success of Lagrangian relaxation in solving constrained optimization problems, Primal-Dual Optimization (PDO) [Chow et al., 2017] and its variants [Tessler et al., 2018; Stooke et al., 2020] leverage the Lagrangian Duality in constrained reinforcement learning. Although plenty of work has provided rigorous analysis on duality gap [Paternain et al., 2019], non-asymptotic convergence rate [Ding et al., 2020] and regret bound [Ding et al., 2021], Primal-dual methods are still hard to be implemented and applied in practical use due to their sensitivity to the initialization as well as the learning rate of Lagrangian multipliers.

Primal solution. Constrained Policy Optimization (CPO) [Achiam et al., 2017] directly searches the feasible policy in the trust region and guarantees a monotonic performance improvement while ensuring constraint satisfaction by solving a quadratic optimization problem with appropriate approximations. Projection-based CPO [Yang et al., 2020] updates the policy in two stages by firstly performing regular TRPO [Schulman et al., 2015] and secondly projecting the policy back into the constraint set. Those methods based on local policy search mainly suffer from approximate errors, the inversion of high-dimensional Hessian matrices, and the poor scalability to multiple constraints. To address drawbacks of the quadratic approximation, FOCOPS [Zhang et al., 2020] solves the optimum of constrained policy optimization within the non-parametric space and then derives the first-order gradients of the ℓ2 loss function within the parameter space. Nevertheless, FOCOPS has more auxiliary variables to learn than our first-order optimization objective, and the analytical solutions of FOCOPS are not straightforward with the increasing number of constraints.

The most similar work to our proposed algorithm is Interior-point Policy Optimization (IPO) [Liu et al., 2020] which uses log-barrier functions as penalty terms to restrict policies into the feasible set. However, the interior-point method requires a feasible policy upon initialization which is not necessarily fulfilled and needs a further recovery. Moreover, the log-barrier function possibly leads to numerical issues when the penalty term goes towards infinity, or the solution is not exactly the same as the primal problem. By contrast, we employ the exact penalty function to derive an equivalent unconstrained objective and restrict policy updates in the trust region by clipping the important sampling ratio on both reward and cost terms for the approximate error reduction, which is different from the prior work.

3 Preliminaries

A Markov decision process (MDP) [Sutton and Barto, 2018] is a tuple (S, A, R, P, µ), where S is the state space, A is the action space, R : S × A × S → R is the reward function, P : S × A × S → [0, 1] is the transition probability function to describe the dynamics of the environment, and µ : S → [0, 1] is the initial state distribution. A stationary policy π : S → P(A) maps the given states to probability distributions over action space and Π denotes the set of all stationary policies π. The optimal policy π∗ maximizes the expected discounted return J(π) = E[τ|s0 = s] ∑∞ t=0 γtR(st, at, st+1) where τ = (s0, s1, a1) ≥ 0 is a sample trajectory and τ ∼ π indicates the distribution over trajectories depending on s0 ∼ µ, a1 ∼ π(s), s1+1 ∼ P(·|s1, a1). The value function is defined as Vπ(τ) = E[τ|s0 = s] ∑∞ t=0 γtR(st, at, st+1)|s0 = s, a0 = a1 and the advantage function is defined as Aπ(π, a) = Qπ(π, a) − Vπ(π).

A constrained Markov decision process (CMDP) [Altman, 1999] extends MDP with a constraint set C consisting of cost functions C : S × A × S → R, i = 1, 2, ..., m. We define the expected discounted cost-return J(π) = E[τ|s0 = s] ∑∞ t=0 γtC(st, at, st+1) and Vπ, Qπ, Aπ similarly in MDPs. The set of feasible stationary policies for a CMDP is denoted as ΠC = {π ∈ Π |JC(π) ≤ d, ∀i}, where di is the upper limit of the corresponding safety or cost constraint. The goal of safe RL is to find an optimal policy over the hard safe constraint, i.e., π∗ = arg maxπ∈ΠC J(π).

4 Methodology

Iterative search for optimal policy is commonly used in literature [Schulman et al., 2015; Achiam et al., 2017; Zhang et al., 2020]. We formulate the unbiased constrained policy optimization problem over s ∼ dπ and a ∼ π as:

πk+1 = arg max π s∼dπ a∼π ∑ Aππ|s, a|.{J(π) + 1 γt (s, a) ≤ d, ∀i}.

where dπ(s) = (1 − γ) ∑∞ t=0 γtP(st+1|a, s) denotes the discounted future state distribution.
Proposition 1. The new policy \( \pi_{k+1} \) obtained from the current policy \( \pi_k \) via problem (1) yields a monotonic return improvement and hard constraint satisfaction.

Proof. See the supplemental material.

In this paper, we consider the parametric policy \( \pi(\theta) \), i.e., using the neural network. Let \( r(\theta) = \frac{\nabla \pi(\theta)}{\pi(\theta)} \) be the importance sampling ratio, then we rewrite the optimization problem (1) as follows:

\[
\theta_{k+1} = \arg\min_{\theta} \mathbb{E}_{s \sim d^k} \left[ -r(\theta) A^{\pi_k}_R(s, a) \right] \\
\text{s.t.} \quad \mathbb{E}_{s \sim d^k} \left[ r(\theta) A^{\pi_k}_C(s, a) \right] + (1 - \gamma)(J_{C_k}(\pi_k) - d_i) \leq 0, \forall i. \quad (2)
\]

Different from previous methods [Achiam et al., 2017; Yang et al., 2020] that approximate non-convex problem (2) to convex optimization via Taylor’s formulation on the trust region, we penalize constraints with ReLU operators to the objective function which yields an unconstrained problem:

\[
\theta_{k+1} = \arg\min_{\theta} \mathbb{E}_{s \sim d^k} \left[ -r(\theta) A^{\pi_k}_R(s, a) \right] + \kappa \sum_{i} \max\{0, \nonumber \\
\mathbb{E}_{s \sim d^k} \left[ r(\theta) A^{\pi_k}_C(s, a) \right] + (1 - \gamma)(J_{C_k}(\pi_k) - d_i) \} \quad (3)
\]

Intuitively, the penalized term takes effect when the agent violates the corresponding constraint; otherwise, the objective equals the standard policy optimization when all constraints are fulfilled. Below, we theoretically analyze the equivalence between problem (2) and problem (3), i.e., the ReLU operator constructs an exact penalty function with a finite penalty factor \( \kappa \). To our best knowledge, it is the first work to solve the constrained policy optimization in the perspective of the exact penalty method.

Theorem 1. Suppose \( \lambda \) is the corresponding Lagrange multiplier vector for the optimum of problem (2). Let the penalty factor \( \kappa \) be a sufficiently large constant (\( \kappa \geq ||\lambda||_{\infty} \)), problem (2) and problem (3) share the same optimal solution set.

Proof. See the supplemental material.

Notably, the finiteness of penalty factor \( \kappa \) in Theorem 1 is critical for policy updates because popular methods like square loss function and log-barrier function require the penalty term to go towards positive infinity for an infeasible solution. Otherwise, the optimal solution is not exactly the same as problem (2) [Liu et al., 2020]. However, this would bring about numerical issues in practice and result in unbounded approximate error which will be discussed later.

By now, we construct an equivalent unconstrained objective with ReLU operators and a finite penalty factor \( \kappa \). It is still intractable to solve problem (3) directly over the unknown future state distribution \( \sim d^F \) except for cumbersome off-policy evaluation [Jiang and Li, 2016] from current policy \( \pi_k \). We replace \( \sim d^F \) with \( \sim d^k \) and minimize the unconstrained objective over collected trajectories:

\[
\theta_{k+1} = \arg\min_{\theta} \mathbb{E}_{s \sim d^k} \left[ -r(\theta) A^{\pi_k}_R(s, a) \right] + \kappa \sum_{i} \max\{0, \nonumber \\
\mathbb{E}_{s \sim d^k} \left[ r(\theta) A^{\pi_k}_C(s, a) \right] + (1 - \gamma)(J_{C_k}(\pi_k) - d_i) \} \quad (4)
\]

This results in biases w.r.t problem (3), and we further provide the analysis of approximate error for that replacement.

Theorem 2. For any policy \( \pi \) and the given \( \pi_k \), define

\[
\delta = \mathbb{E}_{s \sim d^k} \left[ D_{KL}(\pi||\pi_k)[s] \right] .
\]

The worst-case approximate error is \( \mathcal{O}(\kappa m \sqrt{\delta}) \) if we replace problem (3) with problem (4).

Proof. See the supplemental material.

Theorem 2 indicates the necessity of a finite penalty factor \( \kappa \), or the approximate error would not be controlled. On the other hand, we have to restrict policy updates within the neighborhood of last \( \pi_k \) (i.e., \( \delta \to 0 \)) and improve the policy iteratively. Here, we incorporate the trust-region constraint via clipped surrogate objective [Schulman et al., 2017] in the approximate exactly penalized problem (4) to guarantee a proximal policy iteration. Then, the final optimization objective, abbreviated as P3O (penalized PPO), is derived as:
each agent is not shared in the CMDP execution. Note that the cost for hazards and pillars are calculated separately and hold different upper limits. (d) Simple Spread: Agents are rewarded for reaching corresponding destinations, but are constrained to the mutual collision. The observation of each agent is not shared in the CMDP execution.

\[
\mathcal{L}^{P3O}(\theta) = \mathcal{L}^{CLIP}_R(\theta) + \kappa \cdot \sum_i \max\{0, \mathcal{L}^{CLIP}_{C_i}(\theta)\},
\]

where

\[
\mathcal{L}^{CLIP}_R(\theta) = \mathbb{E}_{s \sim d_R, a \sim \pi_k} \left[ -\min\{r(\theta)A^\pi_k(s, a), 
\quad \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon)A^\pi_k(s, a)\} \right];
\]

\[
\mathcal{L}^{CLIP}_{C_i}(\theta) = \mathbb{E}_{s \sim d_{C_i}, a \sim \pi_k} \left[ \max\left\{r(\theta)A^\pi_k(s, a), \text{clip}(r(\theta),
\quad 1 - \epsilon, 1 + \epsilon)A^\pi_k(s, a)\right\} \right] + (1 - \gamma)(J_{C_i}(\pi_k) - d_i).
\]

Based on the clipped surrogate objective technique, we solve the difficult constrained problem (2) by alternately estimating and minimizing the unconstrained loss function (5).

The remaining problem is how to obtain the fine penalty factor \(\kappa\) due to the variety of the magnitude of \(\mathcal{L}^{CLIP}_R(\theta)\) and each \(\mathcal{L}^{CLIP}_{C_i}(\theta)\) that depends on different tasks and up-to-date policies. The value of \(\kappa\) is required to be larger than the unknown greatest Lagrange multiplier according to Theorem 1, but Theorem 2 implies too large \(\kappa\) decays the performance. Thus, there is a trade-off among the hyper-parameter tuning. As shown in Algorithm 2, we increase \(\kappa\) at every time step, and the early stopping condition is fulfilled when the distance between solutions of two adjacent steps is small enough or the current policy is out of the trust region. In practice, we utilize the normalization trick that maps the advantage estimation to an approximate standard normal distribution regardless of the tasks themselves. We find this technique enables a fixed \(\kappa\) for general good results across different tasks. Experimental results show that both of above algorithms work effectively and the learning processes are stable in a wide range of \(\kappa\).

Note that the first term in the P3O objective (5) is the same as standard PPO and the rest terms are the clipped constraint objectives that prevent the new policy from jumping out of the trust region during the penalty reduction. The loss function in P3O is differentiable almost everywhere and can be solved easily via the simple first-order optimizer [Kingma and Ba, 2014]. Conclusively, P3O inherits the benefits of PPO in CMDPs such as better accuracy and sample efficiency than the approximating quadratic optimization [Achiam et al., 2017; Yang et al., 2020]. Additionally, the P3O algorithm is naturally scalable to multi-constraint scenarios by adding penalized terms onto the objective (5). The pseudo-code of the P3O algorithm for solving general multi-constraint CMDPs is shown in Algorithm 1. In the end, we extend the P3O algorithm to the decentralized multi-agent setting in which the CMDP is partially observable and the observation can not be shared between each agent. Due to the space limitation, detailed algorithms are provided in the supplementary material.

5 Experiments

In this section, we empirically demonstrate the efficacy of the P3O algorithm from the following four perspectives:

- P3O outperforms the state-of-the-art algorithms, e.g. CPO [Achiam et al., 2017], PPO-Lagrangian [Ray et al., 2019] and FOCOPS [Zhang et al., 2020] in safe RL benchmarks with a single constraint.
- P3O is robust to more stochastic and complex environments [Ray et al., 2019] where previous methods fail to simultaneously improve the return effectively and satisfy the constraint strictly.
- P3O introduces only one additional hyper-parameter (i.e., the penalty factor \(\kappa\)), which is easy to tune and insensitive to different settings and constraint thresholds.
- P3O can be extended to multi-constraint and multi-agent scenarios that are barely tested in the previous work.

We design and conduct experiments in 2 single-constraint (Circle and Gather), 1 multi-constraint (Navigation) and 1 multi-agent (Simple Spread) safe RL environments respectively, as illustrated in Figure 1.

To be fair in comparison, the proposed P3O algorithm and FOCOPS [Zhang et al., 2020] are implemented with same rules and tricks on the code-base of Ray et al.[2019] for benchmarking safe RL algorithms. Also, we take standard PPO as the reference which ignores any constraint and serves as an upper bound baseline on the reward performance.

More information about experiment environments and detailed parameters are provided in the supplementary material.
Single-Constraint Scenario. The numerical results of different algorithms are listed in Table 1. The proposed P3O algorithm has +15%, +24%, +9% higher reward improvement over CPO, PPO-Lagrangian, FOCOPS on the AntCircle task, and +34%, +23%, +31% on the PointGather task respectively. Meanwhile, P3O converges to the upper limit of the safety constraint more tightly which enlarges the parametric search space, as well as satisfies the hard constraint.

As illustrated in Figure 2, P3O is the best for improving the policy while satisfying the given constraint. Conversely, CPO has more constraint violations and even fails to satisfy the constraint after convergence possibly due to Taylor’s approximation. This is especially prominent in complex environments like SafetyGym [Ray et al., 2019] shown in Figure 7. The non-negligible approximate error prevents the CPO agent from fully ensuring constraints on virtually over these environments whereas it enjoys a meaningless high return. As for PPO-Lagrangian, it is too conservative for the trade-off between reward improvement and constraint satisfaction. Besides, it fluctuates mostly on the learning curves which means it is less steady on the learning process. FOCOPS is better-performed than the two baselines above. However, the hyper-parameter \( \lambda \) takes a great effect on the final performance. At last, both PPO-Lagrangian and FOCOPS are more or less sensitive to the initial value as well as the learning rate of the Lagrange multiplier. Improper hyper-parameters would cause catastrophic performance on reward and constraint value.

Sensitivity Analysis. The penalty factor \( \kappa \) plays a critical role in P3O, which is supposed to vary across different tasks depending on the specific value of \( A_R \) and each \( A_c \). As shown in Algorithm 2, we can increase \( \kappa \) at every time steps. Otherwise we can map the advantage estimation to an approximate standard normal distribution by applying advantage normalizing tricks and use a fixed \( \kappa \) for general good results. The experimental results are stable in a wide range of \( \kappa \), as illustrated in Figure 3.

We further verify that P3O works effectively for different threshold levels by changing the cost limit \( d \). Figure 4 shows P3O can effectively learn constraint-satisfying policies for all the cases. Notably, P3O precisely converges to the cost upper limit. By contrast, the baseline algorithms are too conservative for policy updates and have more overshoot in the process of cost violation reduction.

Figure 2: Average episode return and cost in the single-constraint scenario. The x-axis is the number of interactions with the emulator. The y-axis is the average reward/cost-return. The solid line is the mean and the shaded area is the standard deviation. Each tested algorithm runs over five different seeds. The dashed line in the cost plot is the constraint threshold which is 50 for Circle and 0.5 for Gather.

Figure 3: Performance of P3O for different \( \kappa \) settings on AntCircle.

Figure 4: Performance of P3O for different cost limit \( d \) on AntCircle.

Table 1: Mean performance with normal 95% confidence interval on single-constraint tasks.
Multi-Constraint Scenario. It is convenient to extend P3O to multi-constraint scenarios by simply adding exact penalty functions with the minor modification on the loss function. Learning curves of average episode return and two independent costs for the navigation task are shown in Figure 5. The unconstrained algorithm is the original PPO. The hazard (pillar) constrained algorithm only takes the cost of hazard (pillar) into P3O loss function whereas ignoring the other one. It can be seen that those single-constrained algorithms maintain the specified constraint under its upper limit but fail to control the other one. When both constraints are incorporated into our algorithm, each one is well satisfied at the convergence. The episode return is lower than unconstrained and single-constrained settings, which is reasonable because the safety constraints are more strict.

It is remarkable that at the beginning of training, the agent violates the hazard constraint but barely hits pillars because there are more small hazards around the starting point and the pillars are nearly unreachable by random moving due to the initialization mechanism. However, with the increasing velocity, the agent is less likely to satisfy the pillar constraint. Satisfaction on the two types of constraints shows that P3O admits both feasible and infeasible initial policies and finds a constraint-satisfying solution in the end.

Multi-Agent Scenario. We also study Multi-Agent Penalized Policy Optimization (MAP3O) in the collaborative task and take unconstrained Multi-Agent PPO (MAPPO) [Chao et al., 2021] as the reference. To tackle the issue of partial observability, recurrent networks are used in both algorithms.

The given task is a decentralized partially observable Markov decision process (DEC-POMDP) in which each agent has an exclusive observation and a shared global reward/cost. We replace the MAPPO loss function with Eq. (5) and conduct a fully decentralized training and execution. The experimental results show that MAP3O can improve the total reward while ensuring the mutual collision constraint. Inspired by Chao et al. [2021], we enhance MAP3O by learning a centralized critic for both global reward and cost that takes in the concatenation of each agent’s local observation. The centralized version of MAP3O has better reward performance and satisfies the constraint more strictly. The learning curves of average episode return and cost are shown in Figure 6.

6 Conclusion

In this paper, we propose Penalized Proximal Policy Optimization (P3O), an effective and easy-to-implement algorithm for safe reinforcement learning. P3O has an unconstrained optimization objective due to the exact penalty reformulation and guarantees better performance on reward and constraint value empirically. Meanwhile, it is naturally compatible with multi-constraint and multi-agent scenarios. P3O adopts first-order minimization and avoids the inversion of high-dimensional Hessian Matrix, which is especially beneficial to large CMDPs with deep neural networks. Thus we consider the future work as end-to-end visual tasks which are less studied in safe reinforcement learning.
Acknowledgements
This work is supported by Science and Technology Innovation 2030 – “Brain Science and Brain-like Research” Major Project (No. 2021ZD0201402 and No. 2021ZD0201405).

References

[Achiam et al., 2017] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In International Conference on Machine Learning, pages 22–31. PMLR, 2017.

[Altman, 1999] Eitan Altman. Constrained Markov decision processes, volume 7. CRC Press, 1999.

[Chao et al., 2021] YU Chao, A VELU, E VINITSKY, et al. The surprising effectiveness of ppo in cooperative, multi-agent games. arXiv preprint arXiv:2103.01955, 2021.

[Chow et al., 2017] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. The Journal of Machine Learning Research, 18(1):6070–6120, 2017.

[Ding et al., 2020] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo R Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. In NeurIPS, 2020.

[Ding et al., 2021] Dongsheng Ding, Xiaohan Wei, Zhouran Yang, Zhaorao Wang, and Mihailo Jovanovic. Provably efficient safe exploration via primal-dual policy optimization. In International Conference on Artificial Intelligence and Statistics, pages 3304–3312. PMLR, 2021.

[Garcia and Fernández, 2015] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(1):1437–1480, 2015.

[Jiang and Li, 2016] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In International Conference on Machine Learning, pages 652–661. PMLR, 2016.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[Levine et al., 2016] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research, 17(1):1334–1373, 2016.

[Liu et al., 2020] Yongshuai Liu, Jiaxin Ding, and Xin Liu. Ipo: Interior-point policy optimization under constraints. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 4940–4947, 2020.

[Liu et al., 2021] Yongshuai Liu, Avishai Halev, and Xin Liu. Policy learning with constraints in model-free reinforcement learning: A survey. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 2021.

[Paternain et al., 2019] Santiago Paternain, Luiz FO Chamon, Miguel Calvo-Fullana, and Alejandro Ribeiro. Constrained reinforcement learning has zero duality gap. arXiv preprint arXiv:1910.13393, 2019.

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

[Schulman et al., 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In International conference on machine learning, pages 1889–1897. PMLR, 2015.

[Schulman et al., 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[Silver et al., 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. nature, 550(7676):354–359, 2017.

[Stooke et al., 2020] 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.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

[Tessler et al., 2018] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. arXiv preprint arXiv:1805.11074, 2018.

[Vinyals et al., 2019] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. arXiv preprint, 2018.

[Yang et al., 2020] Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. arXiv preprint arXiv:2010.03152, 2020.

[Zhang et al., 2020] Yiming Zhang, Quan Vuong, and Keith W Ross. First order constrained optimization in policy space. arXiv preprint arXiv:2002.06506, 2020.