# Mean-Semivariance Policy Optimization via Risk-Averse Reinforcement Learning (Extended Abstract)[*]

**Xiaoteng Ma**[1] , **Shuai Ma**[2] , **Li Xia**[2] and **Qianchuan Zhao**[1]
[1]Department of Automation, Tsinghua University
[2]School of Business, Sun Yat-sen University
ma-xt17@mails.tsinghua.edu.cn, mash35@mail.sysu.edu.cn,
xiali5@sysu.edu.cn, zhaoqc@tsinghua.edu.cn

## Abstract

Keeping risk under control is often more crucial than maximizing expected rewards in real-world decision-making situations, such as finance, robotics, autonomous driving, etc. The most natural choice of risk measures is variance, while it penalizes the upside volatility as much as the downside part. Instead, the (downside) semivariance, which captures negative deviation of a random variable under its mean, is more suitable for risk-averse proposes. This paper aims at optimizing the mean-semivariance (MSV) criterion in reinforcement learning w.r.t. steady reward distribution. Since semivariance is time-inconsistent and does not satisfy the standard Bellman equation, the traditional dynamic programming methods are inapplicable to MSV problems directly. To tackle this challenge, we resort to Perturbation Analysis (PA) theory and establish the performance difference formula for MSV. We reveal that the MSV problem can be solved by iteratively solving a sequence of RL problems with a policy-dependent reward function. Further, we propose two on-policy algorithms based on the policy gradient theory and the trust region method. Finally, we conduct diverse experiments from simple bandit problems to continuous control tasks in MuJoCo, which demonstrate the effectiveness of our proposed methods.

## 1 Introduction

Reinforcement learning (RL) has shown great promise in solving complex decision problems, such as Go [Silver *et al.*, 2017], video games [Berner *et al.*, 2019; Vinyals *et al.*, 2019] and dexterous robotic control [Nagabandi *et al.*, 2020]. Learning by trial and error, RL enables an agent to maximize its accumulated expected rewards through interaction with a simulator. However, RL deployment in real-world scenarios is still challenging and unreliable [Garcıa and Fernández, 2015; Dulac-Arnold *et al.*, 2019]. One of the reasons is that real decision-makers need to consider multi-objective functions. The desired policy should perform well for broader metrics, not just for expectation. That raises the demand of *risk-sensitive learning*, which aims at balancing the return and risk in the face of uncertainty.

The risk-sensitive decision-making has been widely studied beyond the scope of RL, which can be traced back to the *mean-variance* (MV) optimization theory established by Markowitz [Markowitz, 1952]. Variance, which captures the fluctuation and concentration of random variables, is a natural choice of the risk measure. As Markowitz only considers the single-period problem, many studies focus on extending the results to multi-period scenarios, from stochastic control [Li and Ng, 2000] to Markov decision process [Sobel, 1982; Filar *et al.*, 1989]. However, the variance of a multi-period problem depends on the average value of the whole process. It breaks the essential property of dynamic programming— time consistency, and makes it hard to design model-free learning algorithms under the standard RL framework. Developing an efficient algorithm to optimize MV is still an ongoing topic in the RL community [Xie *et al.*, 2018; Bisi *et al.*, 2020; Xia, 2020; Zhang *et al.*, 2021; Ma *et al.*, 2022b; Ma *et al.*, 2022a].

While MV analysis is the most widely applied risk-return analysis in practice, variance metric is questionable as a risk measure. As a measure of volatility, variance penalizes upside deviations from the mean as much as downside deviations. It could be problematic as the upside deviation comes from the higher return which is desirable. In general, the outcome distributions in the real world are often asymmetrical, such as the ones in the stock market [Estrada, 2007; Bollerslev *et al.*, 2020], suggesting that we should control the "good" and "bad" volatility separately. Hence, Markowitz 1959 presents the *mean-semivariance* (MSV) as an alternative measure, which only penalizes the "bad" volatility, performing as a downside risk indicator. Even if the distribution is symmetrical, optimizing MSV is at least effective as optimizing MV. To better illustrate the difference between variance and semivariance, we construct a simple MDP example shown in Figure 1. The two policies result in two reward distributions symmetrically, for which variances are indistinguishable. However, the policy going right is preferred since it results in a lower semivariance.

Though MSV is a more plausible measure of risk, optimizing MSV is even more complicated than MV. It inherits

[*]Here is the original journal paper [Ma *et al.*, 2022c].

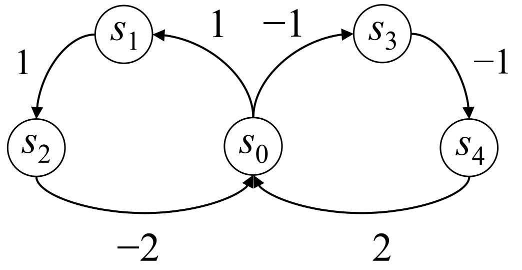


Figure 1: A toy example illustrates the effect of MSV. We refer the policy going left as $l$ and the other as $r$. Two policies have the same average return $\eta^l = \eta^r = 0$ and the same variance $\zeta^l = \zeta^r = 2$. However, since the semivariance $\zeta^l_- = 4/3 > \zeta^r_- = 2/3$, the policy going right has a smaller (downside) semivariance. It shows that MSV enables to avoid extreme costs compared with MV.

time inconsistency from variance and introduces a truncation function of mean, making the analysis non-trivial. Due to the complexity of this objective, existing works consider a subset of problems restricted with a fixed mean [Wei, 2019] or heuristic algorithms for MSV [Yan *et al.*, 2007; Zhang *et al.*, 2012; Liu and Zhang, 2015; Chen *et al.*, 2019]. To the best of our knowledge, there are currently no relevant studies on MSV in the RL literature.

In this paper, we aim to fill the gap of the previous study on the single-period MSV problem and extend the static methods to online RL algorithms. To achieve that, we resort to Perturbation Analysis (PA) theory [Cao, 2007] (also called the sensitivity-based optimization theory or the relative optimization theory) for Markov systems, which lays the basis of many efficient RL methods, such as TRPO [Schulman *et al.*, 2015], CPO [Achiam *et al.*, 2017] and MBPO [Janner *et al.*, 2019]. The contributions of our work are threefold. *Firstly*, instead of constructing a Bellman operator, we establish the MSV performance difference formula of two policies (see Section 3 for details). The result indicates that the performance difference can be decomposed into two parts: the improvement corresponding to a reward function depending on the current policy and the average performance change from the current to the updated one. *Second*, we iteratively optimize MSV by considering the shift in mean locally and constructing a surrogate reward function. We develop two algorithms based on the policy gradient theory and the trust region method, respectively. We show that optimizing the surrogate reward function in the trust region has a similar performance lower bound with the standard TRPO, which guarantees monotonic improvement if the trust region is tight. *Finally*, we conduct diverse experiments to examine the effectiveness of our proposed methods, including a bandit problem, a tabular portfolio management problem, and robotic control tasks based on MuJoCo. The results demonstrate that the proposed algorithms successfully improve the performance under the criterion of MSV, which is better than standard RL from a risk-averse perspective.

## 2 Preliminaries

In this paper, we focus on the infinite-horizon discrete-time MDP as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, P, \pi_0 \rangle$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $r : \mathcal{S} \times \mathcal{A} \mapsto [-R_{\max}, R_{\max}]$ denotes a bounded reward function and $P : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ is the transition matrix and $\pi_0 \in \Delta(\mathcal{S})$ denotes the initial state distribution. We assume that all the involved MDPs are ergodic. Let $\mu : \mathcal{S} \mapsto \Delta(\mathcal{A})$ denote a Markovian randomized policy and $\Pi$ denote the randomized policy space.

We are interested in the long-run average reward

$$\eta^\mu := \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{\pi_0,\mu} \left[ \sum_{t=0}^{T-1} r_t \right], \tag{1}$$

where $\mathbb{E}_{\pi_0,\mu}$ stands for the expectation with $s_0 \sim \pi_0, a_t \sim \mu(\cdot \mid s_t), s_{t+1} \sim P(\cdot \mid s_t, a_t)$. Note that $\eta^\mu$ is independent of $\pi_0$ when $T \to \infty$. The variance and semivariance w.r.t. $\mu$ are defined by

$$\zeta^\mu := \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{\pi_0,\mu} \left[ \sum_{t=0}^{T-1} (r_t - \eta^\mu)^2 \right], \tag{2}$$

$$\zeta^\mu_- := \lim_{T\to\infty} \frac{1}{T} \mathbb{E}_{\pi_0,\mu} \left[ \sum_{t=0}^{T-1} (r_t - \eta^\mu)^2_- \right], \tag{3}$$

where $(\cdot)_- := \min\{0, \cdot\}$. In this paper, we focus on the mean-semivariance criterion,

$$\xi^\mu_- := \eta^\mu - \beta \zeta^\mu_-,$$

where $\beta \geq 0$ is the parameter for the trade-off between mean and semivariance. Analogously, when mean-variance criterion is mentioned, we mean $\xi^\mu := \eta^\mu - \beta \zeta^\mu$.

We further respectively define the state-value function, action-value function, and advantage function for average reward as

$$\begin{aligned} V^\mu_\eta(s) &:= \mathbb{E}_\mu \left[ \sum_{t=0}^{\infty} (r_t - \eta^\mu) \mid s_0 = s \right], \\ Q^\mu_\eta(s,a) &:= \mathbb{E}_\mu \left[ \sum_{t=0}^{\infty} (r_t - \eta^\mu) \mid s_0 = s, a_0 = a \right], \\ A^\mu_\eta(s,a) &:= Q^\mu_\eta(s,a) - V^\mu_\eta(s). \end{aligned}$$

Similarly, the value functions for semivariance are defined as

$$\begin{aligned} V^\mu_{\zeta_-}(s) &:= \mathbb{E}_\mu \left[ \sum_{t=0}^{\infty} \left( (r_t - \eta^\mu)^2_- - \zeta^\mu_- \right) \mid s_0 = s \right], \\ Q^\mu_{\zeta_-}(s,a) &:= \mathbb{E}_\mu \left[ \sum_{t=0}^{\infty} \left( (r_t - \eta^\mu)^2_- - \zeta^\mu_- \right) \mid s_0 = s, a_0 = a \right], \\ A^\mu_{\zeta_-}(s,a) &:= Q^\mu_{\zeta_-}(s,a) - V^\mu_{\zeta_-}(s). \end{aligned}$$

For notation simplicity, we will omit the superscript "$\mu$" when the context is clear, e.g., the average rewards $\eta^\mu, \eta^{\mu'}$ are written as $\eta, \eta'$ instead. When $r$ is mentioned, we omit $(s,a)$ and use $r$ in short.

## 3 Perturbation Analysis

In this section, we derive the *MSV performance difference formula* (MSVPDF), where the core concept—performance difference formula—comes from the PA for Markov systems, also called the sensitivity-based optimization theory. With the aid of MSVPDF, we obtain the necessary optimality condition for the MSV problem. It also lays the basis for developing optimization algorithms (see Section 4), such as the policy gradient method and the trust region method.

### 3.1 Performance Difference Formula

MSVPDF is formally stated below.

**Theorem 1.** *For any two policies $\mu, \mu' \in \Pi$, we have*

$$\begin{aligned}\xi'_- - \xi_- =&\mathbb{E}_{s\sim\pi', a\sim\mu'}[A^{\mu}_{\eta}(s,a) - \beta A^{\mu}_{\zeta_-}(s,a)] \\ &- \beta\mathbb{E}_{s\sim\pi', a\sim\mu'}[(r-\eta')^2_- - (r-\eta)^2_-]\end{aligned} \tag{4}$$

The MSVPDF in Equation 4 claims that the MSV improvement can be divided into two parts. The first term in Equation 4 is a standard MDP with $f$ as the reward function, and the second term is caused by the perturbation of the mean. It clearly quantifies the difficulty of solving the MSV problem, i.e., *the policy-dependent reward function breaks down the time-consistent nature of MDPs*. Meanwhile, it also shows us the standard MDP algorithm such as policy iteration (PI) is unavailable. A PI-like algorithm may be efficient in improving the first term, but the sign of the remaining term (dependent on $\eta'$) is unpredictable. It suggests that we need novel tools to guarantee policy improvement.

### 3.2 Performance Derivative Formula

While Equation 4 describes the performance difference between any two policies, we still need the local structure of the MSV problem to guide the direction of optimization. Following the line of the last part, we present the *MSV performance derivative formula* in this subsection, which describes the performance derivative at $\mu$ towards another policy $\mu'$.

**Theorem 2.** *Given any two policies $\mu, \mu' \in \Pi$, we consider a mixed policy $\mu^{\nu}$,*

$$\mu^{\nu}(a \mid s) = (1-\nu)\mu(a \mid s) + \nu\mu'(a \mid s),$$

*where the action follows $\mu$ with probability $1-\nu$, and follows $\mu'$ with probability $\nu$ for $\nu \in [0,1]$. We have*

$$\frac{\mathrm{d}\xi_-}{\mathrm{d}\nu} = \mathbb{E}_{s\sim\pi, a\sim\mu'}[(1+2\beta\eta_-)A^{\mu}_{\eta}(s,a) - \beta A^{\mu}_{\zeta_-}(s,a)].$$

The above equality indicates that the performance derivative is related to a pseudo another reward function:

$$g(s,a) = (1+2\beta\eta_-)r - \beta(r-\eta)^2_-, \tag{5}$$

and the derivative formula can be written as

$$\frac{\mathrm{d}\xi_-}{\mathrm{d}\nu} = \mathbb{E}_{s\sim\pi, a\sim\mu'}[A^{\mu}_{g}(s,a)], \tag{6}$$

where $A^{\mu}_{g}(s,a)$ is the advantage function w.r.t. $g$.

## 4 Optimization and Algorithms

In this section, we propose two approaches to optimize MSV with the parameterized policy. We firstly extend the policy gradient method to MSV with the pseudo reward function (cf. Equation 5) in Section 3. Following the same idea, we propose a trust region method to solve the MSV problem and prove the lower bound for its performance improvement. The two approaches together establish an iterative framework to solve the MSV problem.

### 4.1 MSV Policy Gradient Method

Policy gradient theorem is an essential foundation of modern deep RL algorithms, such as Actor-Critic methods. Here we consider the policy $\mu$ parameterized by $\theta \in \Theta$, which can be implemented with any differentiable function. We first give the MSV Policy Gradient (MSVPG) theory formally as follows.

**Theorem 3.** *For a policy $\mu$ parameterized by $\theta$, we have*

$$\nabla_{\theta}\xi_- = \mathbb{E}_{s\sim\pi, a\sim\mu}[\nabla_{\theta}\log\mu(a \mid s)A^{\mu}_{g}(s,a)]. \tag{7}$$

The policy gradient for MSV can be easily proved by PA, which follows the same lines as the derivative formula.

### 4.2 MSV Trust Region Method

While PG has a concise form, it often suffers from the difficulty of selecting step-sizes and the sensitivity to initial points in practice, especially when it works with neural networks. To address these drawbacks, trust region method [Schulman *et al.*, 2015] is proposed to solve a surrogate problem in a local trust region and perform an approximate policy iteration.

**Monotonic Improvement Guarantee**

We extend the idea of trust region in the standard MDP into MSV and propose the MSV Trust Region Policy Optimization (MSVTRPO) method. In MSVTRPO, we iteratively solve the problem below

$$\begin{aligned}&\max_{\mu_{\theta}} \mathcal{L}^{\mu}_{g}(\mu_{\theta}) \\ &\text{s.t. } \mathbb{E}_{s\sim\pi}D_{\mathrm{TV}}(\mu_{\theta}(\cdot \mid s) \parallel \mu(\cdot \mid s)) \le \epsilon_{\mu},\end{aligned} \tag{8}$$

where

$$\mathcal{L}^{\mu}_{g}(\mu_{\theta}) := \mathbb{E}_{s\sim\pi, a\sim\mu_{\theta}}\left[A^{\mu}_{g}(s,a)\right].$$

**Remark 1.** *The trust region method updates the policy via the direction of maximum derivative (cf. the performance derivative formula in Equation 6), constrained in the proximity policy space with the $TV$-divergence. In contrast, the standard policy iteration scheme updates the policy in the same direction without constraint, which breaks the monotonic improvement for MSV.*

Next, we will show that MSVTRPO enjoys an analogous performance improvement bound. When the trust region is tight enough, i.e., $\epsilon_{\mu} \to 0$, the lower bound is dominated by the first-order term.

**Theorem 4.** *Let $\mu'$ be the solution to the problem defined by Equation 8. We have*

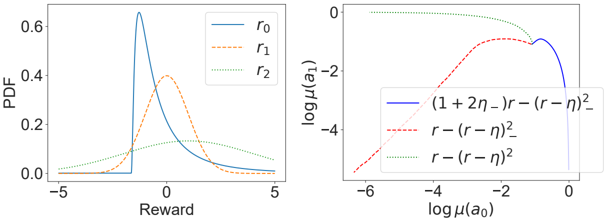
$$\xi' - \xi \ge \mathcal{L}^{\mu}_{g}(\mu') - 2(\kappa'-1)\epsilon_{g}\epsilon_{\mu} - 12\beta(\kappa')^2 R^2_{\max}\epsilon^2_{\mu},$$

*where $\epsilon_g = \max_s |\mathbb{E}_{a\sim\mu'}[A^{\mu}_{g}(s,a)]|$ and $\kappa'$ is Kemeny's constant under $\mu'$.*

## 5 Experiments

In the previous sections, we analyze the properties of MSV problem and find that it can be solved by iteratively optimizing a surrogate reward function $g$ (cf. Equation 5). We also propose two methods to solve the MSV problem in the parameterized policy space.

To validate the effectiveness of our proposed methods in solving the MSV problem, we conduct a series of experiments to answer the corresponding questions:



(a) Reward distributions.

(b) Policies paths.

Figure 2: The bandit problem. (a) Reward distributions in the bandit problem. (b) Policies paths in the bandit problem. The paths are shown in the logarithmic parameter space.

- Is the MSV really optimized by the surrogate reward function $g$? Specifically, what is the difference between optimizing $g$ instead of $f$?
- What is the difference between the MV [Xia, 2020] and MSV criteria?
- Does the proposed algorithms work well with the current deep RL algorithms?

## 5.1 Bandit Problem

We start with a simple bandit problem. In this problem, there are three actions with only a single state. Different actions result in different rewards following the distributions shown in Figure 2(a). We compare three different agents, which optimize different reward functions. The first one optimizes $g = (1 + 2\eta_-)r - (r - \eta)^2_-$, which is the derived reward function with $\beta = 1$ in this work. The second one optimizes $f = r-(r-\eta)^2_-$, which is the Monte-Carlo return of MSV. We further consider a third agent which optimizes $r-(r-\eta)^2$ [Xia, 2020], an MV objective to illustrate the difference between MSV and MV problems.

The result tells us optimizing the reward $f = r - \beta(r-\eta)^2_-$ cannot optimize the MSV objective even in such a simple problem. This reflects the most essential difference between the optimization of policy-dependent reward and other problems. As discussed in Section 3, to optimize a problem with a policy-dependent reward function, we must consider the perturbation of the mean, at least in MSV problems.

## 5.2 Portfolio Management

In this part, we compare the performances of MSV- and MV-optimal policies in a portfolio management problem, where we need to manage two independent assets and cash.

We change the risk preference parameter $\beta$ and compare the MSVTRPI and MVPI. We depict the result in Figure 3, showing that with a fixed $\beta$, optimizing MSV always results in a larger return than that of MV. Besides, MV is more sensitive than MSV in terms of $\beta$, meaning that a small change of $\beta$ will lead to a quick drop in both the return and risk. To better compare MSV and MV, we also show the “normalized” results of MSV, where we double $\beta$ to provide the same penalty strength as MV. The result shows the normalized MSV also outperforms MV in terms of the average reward, illustrating that MSV is more plausible than MV.

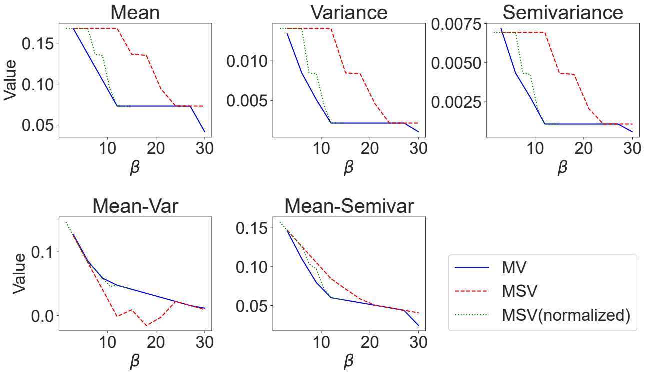


Figure 3: Comparison of MSVTRPI and MVPI in the portfolio management problem. The normalized MSV means $\beta$ is doubled in comparison.

## 5.3 Robotic Control

To demonstrate the effectiveness of our proposed method in more general problem setups, we implement a “deep” variant algorithm named mean-semivariance policy optimization (MSVPO), which is based on the recently developed method APO [Ma *et al.*, 2021] for average-reward RL problems.

We evaluate MSVPO with different $\beta$’s in the noisy Walker2d with different noise levels. When the agent falls, we penalize it with an extra cost -10 and reset the system. In the noiseless environment (noise level = 0), we interestingly find that risk-averse policy ($\beta = 0.1$) achieves competitive average reward with lower semivariance. It indicates that in complex scenes, optimizing a risk-averse metric may generate more robust policies with better performances compared with a risk-neutral one.

To better understand the performance difference with different risk preference policies, we visualize the reward distributions of typical agents in Figure 4, where each agent of noise level 0.1 is evaluated for 1000 steps. We can see that risk-averse policies successfully avoid unsafe states. Meanwhile, the agent uses smaller steps forward with the risk parameter $\beta$ increasing. Instead, the risk-neutral agent tends to take the risk of falling for larger gains.

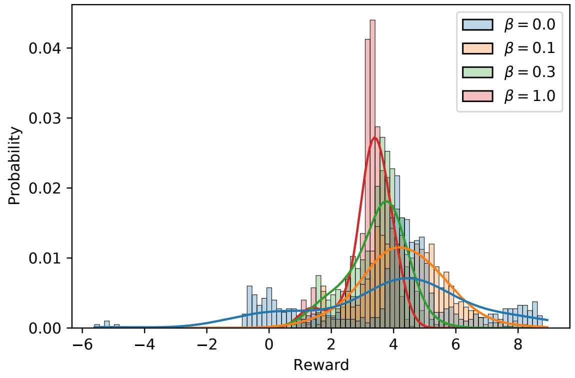


Figure 4: Reward distribution of Walker2d with noise.

## Acknowledgments

This work is funded by the National Natural Science Foundation of China (No. U1813216, 62192751, 61425027, 62073346, 11931018, U1811462), the National Key Research and Development Project of China under Grant 2017YFC0704100 and Grant 2016YFB0901900, in part by the 111 International Collaboration Program of China under Grant BP2018006, BNRist Program (BNR2019TD01009), the National Innovation Center of High Speed Train R&D project (CX/KJ-2020-0006), the Guangdong Province Key Laboratory of Computational Science at the Sun Yat-Sen University (2020B1212060032) and the Guangdong Basic and Applied Basic Research Foundation (2021A1515011984).

## References

[Achiam *et al.*, 2017] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, volume 70, pages 22–31, 2017.

[Berner *et al.*, 2019] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *ArXiv preprint*, abs/1912.06680, 2019.

[Bisi *et al.*, 2020] Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. In *International Joint Conference on Artificial Intelligence*, pages 4583–4589, 2020.

[Bollerslev *et al.*, 2020] Tim Bollerslev, Sophia Zhengzi Li, and Bingzhi Zhao. Good volatility, bad volatility, and the cross section of stock returns. *Journal of Financial and Quantitative Analysis*, 55(3):751–781, 2020.

[Cao, 2007] Xi-Ren Cao. *Stochastic Learning and Optimization: A Sensitivity-Based Approach*. Springer, 2007.

[Chen *et al.*, 2019] Wei Chen, Dandan Li, Shan Lu, and Weiyi Liu. Multi-period mean–semivariance portfolio optimization based on uncertain measure. *Soft Computing*, 23(15):6231–6247, 2019.

[Dulac-Arnold *et al.*, 2019] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *ArXiv preprint*, abs/1904.12901, 2019.

[Estrada, 2007] Javier Estrada. Mean-semivariance behavior: Downside risk and capital asset pricing. *International Review of Economics & Finance*, 16(2):169–185, 2007.

[Filar *et al.*, 1989] Jerzy A Filar, Lodewijk CM Kallenberg, and Huey-Miin Lee. Variance-penalized Markov decision processes. *Mathematics of Operations Research*, 14(1):147–161, 1989.

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

[Janner *et al.*, 2019] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, pages 12498–12509, 2019.

[Li and Ng, 2000] Duan Li and Wan-Lung Ng. Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, 10(3):387–406, 2000.

[Liu and Zhang, 2015] Yong-Jun Liu and Wei-Guo Zhang. A multi-period fuzzy portfolio optimization model with minimum transaction lots. *European Journal of Operational Research*, 242(3):933–941, 2015.

[Ma *et al.*, 2021] Xiaoteng Ma, Xiaohang Tang, Li Xia, Jun Yang, and Qianchuan Zhao. Average-reward reinforcement learning with trust region methods. In *International Joint Conference on Artificial Intelligence*, pages 2797–2803, 2021.

[Ma *et al.*, 2022a] Shuai Ma, Xiaoteng Ma, and Li Xia. An optimistic value iteration for mean–variance optimization in discounted markov decision processes. *Results in Control and Optimization*, 8:100165, 2022.

[Ma *et al.*, 2022b] Shuai Ma, Xiaoteng Ma, and Li Xia. A unified algorithm framework for mean-variance optimization in discounted Markov decision processes. *ArXiv preprint*, abs/2201.05737, 2022.

[Ma *et al.*, 2022c] Xiaoteng Ma, Shuai Ma, Li Xia, and Qianchuan Zhao. Mean-semivariance policy optimization via risk-averse reinforcement learning. *Journal of Artificial Intelligence Research*, 75:569–595, 2022.

[Markowitz, 1952] Harry M Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.

[Markowitz, 1959] Harry M Markowitz. *Portfolio Selection: Efficient Diversification of Investments*. John Wiley & Sons, New York, 1959.

[Nagabandi *et al.*, 2020] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112, 2020.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, volume 37, pages 1889–1897, 2015.

[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.

[Sobel, 1982] Matthew J Sobel. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4):794–802, 1982.

[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. *Nature*, 575(7782):350–354, 2019.

[Wei, 2019] Qingda Wei. Mean–semivariance optimality for continuous-time Markov decision processes. *Systems & Control Letters*, 125:67–74, 2019.

[Xia, 2020] Li Xia. Risk-sensitive Markov decision processes with combined metrics of mean and variance. *Production and Operations Management*, 29(12):2808–2827, 2020.

[Xie *et al.*, 2018] Tengyang Xie, Bo Liu, Yangyang Xu, Mohammad Ghavamzadeh, Yinlam Chow, Daoming Lyu, and Daesub Yoon. A block coordinate ascent algorithm for mean-variance optimization. In *Advances in Neural Information Processing Systems*, volume 31, pages 1073–1083, 2018.

[Yan *et al.*, 2007] Wei Yan, Rong Miao, and Shurong Li. Multi-period semi-variance portfolio selection: Model and numerical solution. *Applied Mathematics and Computation*, 194(1):128–134, 2007.

[Zhang *et al.*, 2012] Wei-Guo Zhang, Yong-Jun Liu, and Wei-Jun Xu. A possibilistic mean-semivariance-entropy model for multi-period portfolio selection with transaction costs. *European Journal of Operational Research*, 222(2):341–349, 2012.

[Zhang *et al.*, 2021] Shangtong Zhang, Bo Liu, and Shimon Whiteson. Mean-variance policy iteration for risk-averse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10905–10913, 2021.