POLICY GRADIENT WITH EXPECTED QUADRATIC UTILITY MAXIMIZATION: A NEW MEAN-VARIANCE APPROACH IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world decision-making problems, risk management is critical. Among various *risk management* approaches, the *mean-variance* criterion is one of the most widely used in practice. In this paper, we suggest *expected quadratic utility maximization* (EQUM) as a new framework for *policy gradient* style *reinforcement learning* (RL) algorithms with mean-variance control. The quadratic utility function is a common objective of risk management in finance and economics. The proposed EQUM framework has several interpretations, such as *reward-constrained variance minimization* and *regularization*, as well as agent utility maximization. In addition, the computation of the EQUM framework is easier than that of existing mean-variance RL methods, which require *double sampling*. In experiments, we demonstrate the effectiveness of the proposed framework in benchmark setting of RL and financial data.

1 INTRODUCTION

Reinforcement learning (RL) and planning in *Markov decision processes* (MDPs) is one type of dynamic decision-making problem (Puterman, 1994; Bertsekas & Tsitsiklis, 1996; sut, 1998). While the typical objective is to maximize the expected cumulative reward, *risk-aware decision-making* has attracted attention in real-world applications, such as finance, robotics, and playing games (Geibel & Wysotzki, 2005; García & Fernández, 2015). The notion of risk in RL is related to the fact that even an optimal policy may perform poorly in some cases owing to the stochastic nature of the problem. To capture the risk, various criteria have been proposed, such as Value at Risk (Luenberger, 1998; Chow & Ghavamzadeh, 2014; Chow et al., 2017) and variance (Markowitz, 1952; Markowitz et al., 2000; Tamar et al., 2012; L.A. & Ghavamzadeh, 2013). Among them, we focus on the mean-variance tradeoff in RL problems.

Typical *mean-variance RL* (MVRL) methods attempt to maximize the expected cumulative reward while maintaining the variance threshold (Tamar et al., 2012; L.A. & Ghavamzadeh, 2013; Prashanth & Ghavamzadeh, 2016; Xie et al., 2018; Bisi et al., 2020; Zhang et al., 2020). However, most existing MVRL methods suffer from high computational costs owing to the *double sampling issue* when approximating the gradient of the variance term (Tamar et al., 2012; L.A. & Ghavamzadeh, 2013; Prashanth & Ghavamzadeh, 2016). To avoid the double sampling issue, Xie et al. (2018) proposed a method based on the Legendre-Fenchel duality (Boyd & Vandenberghe, 2004). Although the method does not suffer from the double sampling issue, we cannot apply a standard policy gradient method and must use a coordinate descent algorithm. In addition, the method cannot control risk at a certain desirable level.

From an economics perspective, the difference in objectives arises from the forms of utility functions. When the objective of an agent is cumulative expected reward maximization, the utility function is *risk-neutral*; when an agent attempts to control the risk based on an expected reward, the utility function is *risk-averse* (Mas-Colell et al., 1995). In economics, there have been several risk-averse utility functions proposed. The *quadratic utility function* is one such function and is frequently used in financial economics (Hull, 2015). Under the quadratic utility function, we can prove that the mean-variance optimization maximizes the utility of the agent. In this study, as one

of the MVRL approaches, we consider the *expected quadratic utility maximization* (EQUM) based on the policy gradient (Williams, 1988; 1992; Sutton et al., 1999; Baxter & Bartlett, 2001).

EQUM has the following advantages: (i) low computational cost; (ii) numerous interpretations; and (iii) direct connections to real-world applications, such as finance. In this study, as interpretations of EQUM, we propose the minimization of the variance under the constraint of the expected cumulative reward, reward-targeting optimization, and regularization. Thus, this study contributes to the context of risk-averse RL and MVRL. In the following sections, we first formulate the problem setting in Section 2 and review the existing methods in Section 3. Then, we propose the EQUM framework and main algorithm based on the simplest policy gradient (Brockman et al., 2016) in Section 4. Finally, we investigate the effectiveness of EQUM in Section 5.

2 Problem Setting

We consider the standard RL framework, where a learning agent interacts with an unfamiliar, dynamic, and stochastic environment modeled by a Markov decision process (MDP) in discrete time. We define the MDP as the tuple $(\mathcal{S}, \mathcal{A}, r, P, P_0)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, $P : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the transition kernel, and $P_0 : \mathcal{S} \to [0, 1]$ is an initial state distribution. The initial state $\mathcal{S}_1$ is sampled from P_0. Let $\pi_\theta : \mathcal{A} \times \mathcal{S} \to [0, 1]$ be a parameterized stochastic policy mapping states to actions, where θ is the tunable parameter. At time step t, an agent chooses an action A_t according to a policy $\pi_\theta(\cdot \mid S_t)$. We assume that the policy π_θ is a differentiable function with respect to θ; that is, $\frac{\partial \pi_\theta(a,s)}{\partial \theta}$ exists.

There are several performance measures for a policy π_θ. One popular measure is the expected total reward from time step t to u defined as $\mathbb{E}_{\pi_\theta}[R_{t:u}]$, where $R_{t:u} = \sum_{i=0}^{u} \gamma^i r(S_{t+i}, A_{t+i})$, $\gamma \in [0, 1]$ is a discount factor and $\mathbb{E}_{\pi_\theta}$ denotes the expectation operator over a policy π_θ, and S_1 is generated from P_0. When the discount factor $\gamma = 1$, to ensure the total reward well defined, it is usually assumed that all policies are proper (Bertsekas & Tsitsiklis, 1996); that is, for any policy π_θ, the agent goes to a recurrent state S^* with probability 1. After the agent passes the recurrent state S^* at a time τ, the rewards are always 0. Such a finite horizon case is called *episodic* MDPs (Puterman, 1994; Tamar et al., 2012). For brevity, we denote $R_{t:u}$ as R when we do not distinguish finite and infinite horizon cases. Under these criteria, the agent may attempt to obtain a higher total reward while taking higher risks. In real-world applications, such as portfolio management in finance (Markowitz, 1952; Markowitz et al., 2000), such risky decision-making is not always desired, and we therefore consider the tradeoff between the expected total reward and the variance. Thus, while the goal of the risk-neutral MDP problem is to find the parameter θ that maximizes the cumulative reward, we consider the mean-variance tradeoff between the total expected reward and the variance of the total reward R in the risk-averse MVRL problem.

3 Existing MVRL Methods

In this section, we introduce existing studies of MVRL.

3.1 Constrained Trajectory-Variance Problem

Tamar et al. (2012), L.A. & Ghavamzadeh (2013), and Xie et al. (2018) formulated MVRL by a constrained optimization problem defined as $\max_{\theta \in \Theta} \mathbb{E}_{\pi_\theta}[R] \text{ s.t. } \mathrm{Var}_{\pi_\theta}(R) \leq \eta$. In their formulation, the goal is to maximize the expected cumulative reward with controlling the trajectory-variance at a certain level. To solve this problem, Tamar et al. (2012), L.A. & Ghavamzadeh (2013), and Xie et al. (2018) consider a penalized method defined as $\max_{\theta \in \Theta} \mathbb{E}_{\pi_\theta}[R] - \delta g\big((\mathrm{Var}_{\pi_\theta}(R) - \eta) \big)$, where $\delta > 0$ is a constant and $g : \mathbb{R} \to \mathbb{R}$ is a penalty function, such as $g(x) = x$ or $g(x) = x^2$.

3.1.1 Double Sampling Issue

Tamar et al. (2012), L.A. & Ghavamzadeh (2013), and Xie et al. (2018) report the double sampling issue in MVRL, which requires sampling from two different trajectories for estimating the policy gradient. For instance, in an episodic MDP with the discount factor $\gamma = 1$ and the stopping time $\tau = \min\{t \mid S_t = S^*\}$, the gradients of $\mathbb{E}_{\pi_\theta}[R]$, $\mathbb{E}_{\pi_\theta}\big[R^2\big]$, and $\big(\mathbb{E}_{\pi_\theta}[R]\big)^2$ are given

as $\nabla_\theta \mathbb{E}_{\pi_\theta}[R] = \mathbb{E}_{\pi_\theta}\left[R \sum_{t=1}^{\tau} \nabla_\theta \log \pi_\theta(S_t, A_t)\right]$, $\nabla_\theta \mathbb{E}_{\pi_\theta}\left[R^2\right] = \mathbb{E}\left[R^2 \sum_{t=1}^{\tau} \nabla_\theta \log \pi_\theta(S_t, A_t)\right]$, and $\nabla_\theta \left(\mathbb{E}_{\pi_\theta}[R]\right)^2 = 2\mathbb{E}_{\pi_\theta}[R]\nabla_\theta \mathbb{E}_{\pi_\theta}[R]$ (Tamar et al., 2012). Besides, the gradient of the variance is given as $\mathbb{E}_{\pi_\theta}\left[R^2 \sum_{t=1}^{\tau} \nabla_\theta \log \pi_\theta(S_t, A_t)\right] - 2\mathbb{E}_{\pi_\theta}[R]\nabla_\theta \mathbb{E}_{\pi_\theta}[R]$. Because optimizing the policy π_θ using the gradients directly is computationally intractable, we replace them with their unbiased estimators. Suppose that there is a simulator generating a trajectory k with $\{(S_t^k, A_t^k, r(S_t^k, A_t^k))\}_{t=1}^{\tau^k}$, where τ^k is the stopping time of the trajectory. Then, we can naively construct unbiased estimators of $\mathbb{E}_{\pi_\theta}[R]$ and $\mathbb{E}_{\pi_\theta}\left[R^2\right]$ as $\widehat{\nabla}_\theta \mathbb{E}_{\pi_\theta}[R] = \widehat{R}^k \sum_{t=1}^{\tau^k} \nabla_\theta \log \pi_\theta(S_t^k, A_t^k)$ and $\widehat{\nabla}_\theta \mathbb{E}_{\pi_\theta}\left[R^2\right] = \left(\widehat{R}^k\right)^2 \sum_{t=1}^{\tau^k} \nabla_\theta \log \pi_\theta(S_t^k, A_t^k)$, where $\widehat{R}^k$ is a sample approximation of $\mathbb{E}_{\pi_\theta}[R]$ at the episode k. However, obtaining an unbiased estimator of $\nabla_\theta \left(\mathbb{E}_{\pi_\theta}[R]\right)^2 = 2\mathbb{E}_{\pi_\theta}[R]\nabla_\theta \mathbb{E}_{\pi_\theta}[R]$ is difficult because it requires sampling from two different trajectories for approximating $\mathbb{E}_{\pi_\theta}[R]$ and $\nabla_\theta \mathbb{E}_{\pi_\theta}[R]$, respectively. This double sampling issue makes the optimization problem difficult.

3.1.2 Existing Solutions to Double Sampling Issue

For the double sampling issue, Tamar et al. (2012), L.A. & Ghavamzadeh (2013), and Xie et al. (2018) presented the following solutions.

Multi-time-scale stochastic optimization: The proposed methods of Tamar et al. (2012) and L.A. & Ghavamzadeh (2013) are based on stochastic approximation to find an equilibrium point of an ordinary differential equation and saddle point of the objective, respectively.

Coordinate descent optimization: Xie et al. (2018) proposed using the Legendre-Fenchel dual transformation with coordinate descent algorithm. First, based on Lagrangian relaxation, Xie et al. (2018) set an objective function as $\max_{\theta \in \Theta} \mathbb{E}_{\pi_\theta}[R] - \delta\left(\mathrm{Var}_{\pi_\theta}(R) - \eta\right)$. Then, Xie et al. (2018) transformed the objective function as $\max_{\theta \in \Theta, y \in \mathbb{R}} 2y\left(\mathbb{E}_{\pi_\theta}[R] + \frac{1}{2\delta}\right) - y^2 - \mathbb{E}_{\pi_\theta}\left[R^2\right]$ and estimated a parameter by solving the optimization problem via a coordinate descent algorithm.

Weakness of existing approaches: The multi-time-scale approaches proposed by Tamar et al. (2012) and L.A. & Ghavamzadeh (2013) are known to be sensitive to the choice of the step-size schedules, which are not easy to be controlled (Xie et al., 2018). The method proposed by Xie et al. (2018) does not reflect the constraint condition η as shown above; that is, in the objective function of Xie et al. (2018), there does not exist the constraint condition η.

3.2 Constrained Per-Step Variance Problem

Bisi et al. (2020) and Zhang et al. (2020) proposed solving a constrained per-step variance problem for MVRL. Bisi et al. (2020) showed that the per-step variance $\mathrm{Var}_{\pi_\theta}(R) \leq \frac{\mathrm{Var}_{\pi_\theta}(r(S_t, A_t))}{(1-\gamma)^2}$, which implies that the minimization of the per-step variance $\mathrm{Var}(r(S_t, A_t))$ also minimizes trajectory-variance $\mathrm{Var}_{\pi_\theta}(R)$. Therefore, they train a policy π_θ by maximizing $\mathbb{E}_{\pi_\theta}[r(S_t, A_t)] - \kappa \mathrm{Var}_{\pi_\theta}(r(S_t, A_t))$, where $\kappa > 0$ is a parameter of the penalty function. The methods of Bisi et al. (2020) and Zhang et al. (2020) are based on the trust region policy optimization (Schulman et al., 2015) and coordinate descent with Legendre-Fenchel duality (Xie et al., 2018), respectively.

3.3 Constrained Cumulative Expected Reward Problem

While existing MVRL studies mainly focus on a constrained trajectory-variance problem, a constrained cumulative expected reward optimization is also frequently used in practical situations, such as finance (Markowitz, 1952; Markowitz et al., 2000). In the constrained cumulative expected reward problem, we solve the following problem:

$$\min_{\theta \in \Theta} \mathrm{Var}_{\pi_\theta}(R) \quad \text{s.t. } \mathbb{E}_{\pi_\theta}[R] = \xi. \tag{1}$$

Our proposed EQUM framework more focus on this problem. As shown in the following section, from a computational perspective, there is a critical difference between the constrained cumulative expected reward and trajectory-variance problems.

4 EQUM FRAMEWORK

In this paper, as a novel approach for MVRL, we propose a Expected Quadratic Utility maximization RL (EQUM). In economic model, by using two parameters $\alpha > 0$ and $\beta > 0$, the quadratic utility function for the total return R is defined as $U(R; \alpha, \beta) = \alpha R - \frac{1}{2}\beta R^2$ for $\alpha > 0, \beta \geq 0$. The quadratic utility function captures the preference of a risk-averse agent over the total return R. Let us consider the expected quadratic utility function defined as

$$\begin{aligned}\mathbb{E}_{\pi_\theta}\left[U(R;\alpha,\beta)\right] &= \alpha\mathbb{E}_{\pi_\theta}\left[R\right] - \frac{1}{2}\beta\mathbb{E}_{\pi_\theta}\left[R^2\right] \\ &= \alpha\mathbb{E}_{\pi_\theta}\left[R\right] - \frac{1}{2}\beta\big(\mathbb{E}_{\pi_\theta}[R]\big)^2 - \frac{1}{2}\beta\mathbb{E}_{\pi_\theta}\left[\big(\mathbb{E}_{\pi_\theta}[R] - R\big)^2\right]. \end{aligned} \tag{2}$$

Thus, we can represent the expected utility function using the mean and variance. In the EQUM framework, we train a policy by maximizing the expected quadratic utility function.

4.1 INTERPRETATIONS

Here, we introduce three interpretations of EQUM. We can interpret the EQUM as an approach for (i) an expected utility maximization, (ii) a targeting optimization problem to achieve an expected cumulative reward ζ, (iii) a constrained trajectory-reward problem with a quadratic penalized function, and (iv) an expected cumulative reward maximization with regularization.

First, we discuss the connection with respect to training an agent to achieve a predefined return. Let ζ be a target return that the algorithm aims to achieve. Then, we consider the following mean squared error (MSE) minimization problem, which minimizes the expected deviation of the return from a target ζ:

$$\arg\min_{\theta\in\Theta} J(\theta;\zeta) = \arg\min \mathbb{E}_{\pi_\theta}\left[\big(\zeta - R\big)^2\right] \tag{3}$$

We can decompose the MSE into the bias and variance as follows:

$$\begin{aligned}&\mathbb{E}_{\pi_\theta}\left[\big(\zeta - R\big)^2\right] \\ &= \underbrace{\big(\zeta - \mathbb{E}_{\pi_\theta}[R]\big)^2}_{\text{Bias}} + \underbrace{2\mathbb{E}_{\pi_\theta}\left[\big(\zeta - \mathbb{E}_{\pi_\theta}[R]\big)\big(\mathbb{E}_{\pi_\theta}[R] - R\big)\right]}_{0} + \underbrace{\mathbb{E}_{\pi_\theta}\left[\big(\mathbb{E}_{\pi_\theta}[R] - R\big)^2\right]}_{\text{Variance}} \\ &= \zeta^2 - 2\zeta\mathbb{E}_{\pi_\theta}[R] + \big(\mathbb{E}_{\pi_\theta}[R]\big)^2 + \mathbb{E}_{\pi_\theta}\left[\big(\mathbb{E}_{\pi_\theta}[R] - R\big)^2\right].\end{aligned}$$

Thus, the minimization problem (3) optimizes the policy π_θ to achieve the goal under the tradeoff between the bias $\big(\zeta - \mathbb{E}_{\pi_\theta}[R]\big)^2$ and variance $\mathbb{E}_{\pi_\theta}\left[\big(\mathbb{E}_{\pi_\theta}[R] - R\big)^2\right]$. Moreover, we find that the EQUM is equivalent to the reward-targeting optimization when $\zeta = \frac{\alpha}{\beta}$; that is,

$$\begin{aligned}\arg\min_{\theta\in\Theta} J\left(\theta;\frac{\alpha}{\beta}\right) &= \arg\min_{\theta\in\Theta}\left(\frac{\alpha}{\beta}\right)^2 - 2\frac{\alpha}{\beta}\mathbb{E}_{\pi_\theta}[R] + \big(\mathbb{E}_{\pi_\theta}[R]\big)^2 + \mathbb{E}_{\pi_\theta}\left[\big(\mathbb{E}_{\pi_\theta}[R] - R\big)^2\right] \\ &= \arg\max_{\theta\in\Theta}\mathbb{E}_{\pi_\theta}\left[U(R;\alpha,\beta)\right].\end{aligned}$$

Second, we can interpret that the bias-variance tradeoff heuristically provides a solution to a constraint optimization problem (1) with the constraint $\xi = \zeta = \frac{\alpha}{\beta}$ by minimizing the following penalized problem:

$$\min_{\theta\in\Theta}\mathrm{Var}_{\pi_\theta}(R) + \big(\mathbb{E}_{\pi_\theta}[R] - \xi\big)^2. \tag{4}$$

Third, we can regard the quadratic utility function as an expected cumulative reward maximization with a regularization term defined as $\mathbb{E}\big[R^2\big]$; that is, minimization of the risk $\mathcal{R}(\pi_\theta)$:

$$\mathcal{R}(\theta) = \underbrace{-\mathbb{E}_{\pi_\theta}[R]}_{\text{Risk of expected cumulative reward maximization}} + \underbrace{\psi\mathbb{E}_{\pi_\theta}\big[R^2\big]}_{\text{Regularization term}} \tag{5}$$

where $\psi > 0$ is a regulation parameter and $\psi = \frac{\beta}{2\alpha} = \frac{1}{2\zeta}$. As $\zeta \to \infty$, $\mathcal{R}(\pi_\theta) \to -\mathbb{E}_{\pi_\theta}[R]$.

4.2 MERITS OF THE EQUM FRAMEWORK

In this section, we present two advantages of the EQUM framework. The first advantage is in computation. The EQUM framework is an MVRL method. However, compared with existing MVRL methods, which involve the double sampling issue, the computation of the EQUM framework is much simpler because we transform the MVRL problem (4) into the optimization problem without the term $(\mathbb{E}_{\pi_\theta}[R])^2$ (see (2) and (5)). The second advantage is that it provides a variety of interpretations. Because the EQUM framework can interpret economic theory, it is applicable for modeling economic dynamics. In addition, as one of the MVRL methods, it is suitable for various real-world applications, such as finance (Deng et al., 2016) and playing games. By regarding the proposed EQUM as a regularization framework, we can combine it with existing RL methods.

4.3 IMPLEMENTATIONS OF THE EQUM FRAMEWORK

Here, we discuss the implementations of the EQUM framework.

Simplest policy gradient with EQUM With the EQUM framework, we introduce a main algorithm with simplest policy gradient (SPG) algorithm (Brockman et al., 2016), which is also called the REINFORCE algorithm (Williams, 1992). For an episode k with length n, the method replaces the expectations $\mathbb{E}_{\pi_\theta}\left[R\right]$ and $\mathbb{E}_{\pi_\theta}\left[R^2\right]$ with the sample approximations $\sum_{t=1}^{n}\gamma^{t-1}r(S_t,A_t)$ and $\left(\sum_{t=1}^{n}\gamma^{t-1}r(S_t,A_t)\right)^2$, respectively (Brockman et al., 2016). Then, the unbiased gradients are

$$\widehat{\nabla}_\theta\mathbb{E}_{\pi_\theta}[R]=\widehat{R}^k\sum_{t=1}^{n}\nabla_\theta\log\pi_\theta(S_t^k,A_t^k)\text{ and }\widehat{\nabla}_\theta\mathbb{E}_{\pi_\theta}\left[R^2\right]=\left(\widehat{R}^k\right)^2\sum_{t=1}^{n}\nabla_\theta\log\pi_\theta(S_t^k,A_t^k).$$

Therefore, for a sample approximation $\widehat{R}^k$ of $\mathbb{E}_{\pi_\theta}\left[R^2\right]$ at the episode k, we optimize the policy with ascending the following gradient: $\left(\alpha\widehat{R}^k-\frac{1}{2}\beta\left(\widehat{R}^k\right)^2\right)\sum_{t=1}^{n}\nabla_\theta\log\pi_\theta(S_t^k,A_t^k)$.

Actor-critic with EQUM: For another combination with the EQUM framework, we apply an actor-critic (AC) based algorithms, which is also refereed to as the advantage actor-critic (A2C) algorithm (Williams & Peng, 1991; Mnih et al., 2016). Extending the AC algorithm, for an episode k with the length n, we train the policy by a gradient defined as

$$\nabla_\theta\log\pi_\theta(S_t^k,A_t^k)\left\{\left(\alpha\widehat{R}_{t:n}^k-\frac{1}{2}\beta\left(\widehat{R}_{t:n}^k\right)^2\right)-\left(\alpha M_{\hat{\omega}_k^{(1)}}^{(1)}(S_t^k)-\frac{1}{2}\beta M_{\hat{\omega}_k^{(2)}}^{(2)}(S_t^k)\right)\right\},$$

where where $\widehat{R}_{t:n}^k=\sum_{i=0}^{n}\gamma^i r(S_{t+i}^k,A_{t+i}^k)$, and $M_{\hat{\omega}_k^{(1)}}^{(1)}(S_t^k)$ and $M_{\hat{\omega}_k^{(2)}}^{(2)}(S_t^k)$ are value functions approximating $\mathbb{E}[R_{t:\infty}]$ and $\mathbb{E}[R_{t:\infty}^2]$ with parameters $\hat{\omega}_k^{(1)}$ and $\hat{\omega}_k^{(2)}$, respectively. Note that n can vary from state to state. Additionally, we present alternative objective functions in Appendix A. At a k-th critic step, we estimate $M_{\hat{\omega}_k^{(1)}}^{(1)}(S_t^k)$ and $M_{\hat{\omega}_k^{(2)}}^{(2)}(S_t^k)$ as

$$\hat{\omega}_k^{(1)}=\underset{\omega\in\Theta_\omega}{\arg\min}\left(\widehat{R}_{t:n-1}^k-M_\omega^{(1)}(S_t^k)\right)^2\text{ and }\hat{\omega}_k^{(2)}=\underset{\omega\in\Theta_\omega}{\arg\min}\left(\left(\widehat{R}_{t:n-1}^k\right)^2-M_\omega^{(2)}(S_t^k)\right)^2.$$

Note that our proposed framework EQUM is agnostic to the learning method; that is, we can extend the main algorithms to use other policy gradient methods such as TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) algorithms.

Determining α and β: Next, we discuss how to determine the parameter α and β, which is equivalent to determine ζ, ξ, and ψ). As explained above, the parameter ψ has several meanings based on the interpretations of the EQUM framework, such as the parameter of the quadratic utility function, targeting optimization, and constrained optimization. In addition, from the regularization perspective, we can adjust the parameter to maximize the expected total reward in the test data. Thus, we propose the following four perspectives to choose the parameter ψ. First, in economic applications, such as finance, we choose $\frac{\beta}{2\alpha}=\psi$ based on the characteristics of the economic players and economic empirical studies. Second, we can set $\zeta=\frac{1}{2\psi}$ as the target reward to achieve. Third, we

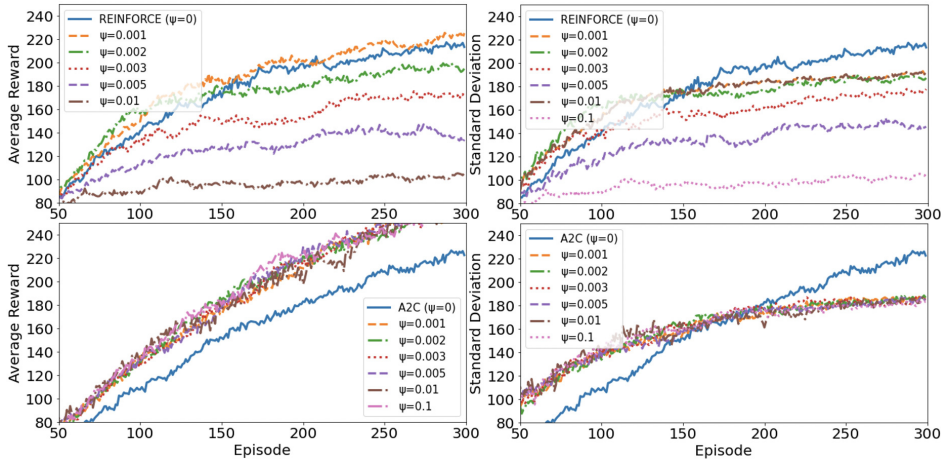

Figure 1: Sensitivity analysis regarding ψ. The upper two graphs are the results using the SPG-based method and the lower two graphs are the results using the AC-based algorithms. The average reward of the SPG result with $\psi = 0.1$ is lower than 80, and we do not show the result here.

can regard the parameter ψ as the constrained problem (1). Fourth, through cross-validation, we optimize the regularization parameter ψ to maximize the expected total reward in the test data.

5 Experiments

In this section, we report the experimental performances of the proposed EQUM framework with well-known benchmarks. Although it is difficult to construct RL methods working well on all problems, we can investigate how the behaviors of existing RL methods are changed by adding the additional $\mathbb{E}[R^2]$ term. Because our motivation of the experiments is in comparing the EQUM framework with the existing RL methods, in experiments of Section 5.1 and 5.2, we use a simple two-layer perceptron for modeling the policy, which is also used in Pytorch example (Paszke et al., 2019). To implement SPG and AC algorithms, we mainly follow the Pytorch example[1]. For brevity, we denote the EQUM framework parameter by ψ. For the SPG-based algorithms, we define the cumulative reward as finite sum with $\gamma = 1$ following (Tamar et al., 2012). For the AC-based algorithms, we define the cumulative reward as infinite sum with $\gamma = 0.99$ following (L.A. & Ghavamzadeh, 2013).

5.1 Sensitivity Analysis on ψ

First, we investigate the sensitivity of the EQUM framework to the parameter ψ. We apply the SPG and AC methods (Section 4.3) with the EQUM framework to the Cartpole problem, which is a classic control problem. We use $\psi = 0.001, 0.002, 0.003, 0.005, 0.01, 0.1$ and compare the results with the standard SPG and AC methods. For instance, from the targeting optimization perspective, EQUM with $\psi = 0.001$ is equal to minimize the MSE between the cumulative reward and $\zeta = 500 = 1/(2 * 0.001)$. We train the model by 300 episodes. We calculate the average reward and standard deviation at each period by conducting $1,000$ trials. The results are shown in Figure 1. In the SPG results, we can confirm the mean-variance tradeoff. In contrast, in the SPG with $\psi = 0.001$ and all AC results, the EQUM framework improves the expected cumulative reward. These results are contrary to our expectations because MVRL methods typically decrease the expected cumulative return to decrease the variance. We discuss this topic in more detail in Section 5.4.

5.2 Experiments of Atari Games

When playing games, we often consider risk-control while maintaining a certain level of reward. Under this motivation, we benchmark the proposed EQUM framework on four Atari game tasks from the OpenAI gym (Brockman et al., 2016). Among the games, we choose BeamRider, Seaquest,

[1] https://github.com/pytorch/examples/tree/master/reinforcement_learning

Table 1: The results on Atari games. The highest ARs and lowest SDs of SPG and EQUM-based methods are highlighted in **bold**, respectively.

Method	SPG		EQUM with SPG				AC		EQUM with AC			
			$\psi = 0.003$		$\psi = 0.005$				$\psi = 0.003$		$\psi = 0.005$	
Game	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD
BeamRider	99.6	315.0	136.8	**288.6**	**265.2**	436.7	325.6	**128.2**	387.6	219.0	**432.8**	242.1
Seaquest	26.0	59.7	26.0	59.7	**34.0**	**61.1**	310.0	154.7	346.0	**145.5**	**354.0**	178.9
Qbert	40.0	**64.8**	**55.0**	71.5	40.0	**64.8**	277.5	197.4	222.5	**80.3**	**290.0**	120.9
SpaceInvaders	169.5	146.0	**171.0**	147.2	109.5	**141.4**	149.0	121.2	**194.0**	115.4	149.0	**100.0**

Qbert, and SpaceInvaders in which SPG and AC methods work well. We use simplified environments in which the observations are the RAM of the Atari machine, consisting of only 128 bytes. We compare our methods with the standard SPG, AC algorithms, and MVLR methods of Tamar et al. (2012) and Xie et al. (2018). We calculate the average reward (AR) at the last episode over 5 trials and the standard deviation (SD). In Table 1, we show the results of SPG and AC with the standard setting and EQUM framework, where ψ is chosen from 0.003 and 0.005. We show the other results in Appendix B. When using $\psi = 0.005$, it is equal to minimize the MSE between the expected cumulative reward and the target 100. In almost all cases, the EQUM framework shows the better AR than the standard methods. As well as the previous sensitivity experiment, this result is contrary to our expectation.

5.3 Experiments of Portfolio Optimization

In financial portfolio experiments, we formulate the problem by an episodic MDP. We use well-known benchmarks called Fama and French (FF) datasets[2] Fama & French (1992) to ensure the reproducibility of the experiment. The FF datasets have been recognized as standard datasets and heavily adopted in finance research because of their extensive asset classes and very long historical data series. We use the FF25 and FF48 datasets. The FF25 dataset includes 25 portfolios formed based on size and book-to-market ratio; the FF48 dataset contains monthly returns of 48 portfolios representing different industrial sectors. We use both datasets covering monthly data from July 1980 to June 2020. The state is past 12 months returns of each asset and the action is defined as portfolio weight; that is, the number of actions is equal to that of assets. The reward is obtained as the portfolio return. Here, the portfolio return at time $1 \leq t \leq T$ is defined as $y_t = \sum_{j=1}^{m} y_{j,t} w_{j,t-1}$, where $y_{j,t}$ is the return of j asset at time t, $w_{j,t-1}$ is the weight of j asset in the portfolio at time $t-1$, and m is the number of assets. The length of the episode is 1 years (12 months). For the stochastic policy, we adopt a three-layer feed-forward neural network with the ReLU activation function where the number of units in each respective layer is equal to the number of assets, 100, 50, and the number of actions. We use the softmax function for the output layer.

Portfolio models: In our experiment, we use the following portfolio models. An equally-weighted portfolio (EW) weights the financial assets equally (DeMiguel et al., 2007). A mean-variance portfolio (MV) computes the optimal variance under a mean constraint (Markowitz, 1952). For computing the mean vector and covariance matrix, we use the latest 10 years (120 months) data. An Kelly growth optimal portfolio with ensemble learning (EGO) is proposed by Shen et al. (2019). We set the number of resamples as $m_1 = 50$, the size of each resample $m_2 = 5\tau$, the number of periods of return data $\tau = 60$, the number of resampled subsets $m_3 = 50$, and the size of each subset $m_4 = n^{0.7}$, where m is number of assets; that is, $m = 25$ in FF25, $m = 48$ in FF48 and $m = 100$ in FF100. A portfolio blending via Thompson sampling (BLD) is proposed by Shen & Wang (2016). We use the latest 10 years (120 months) data to compute for the sample covariance matrix and blending parameters. A policy gradients with variance related risk criteria (Tamer) is proposed by Tamar et al. (2012). We set the target variance terms as 500. A block coordinate ascent algorithm proposed by Xie et al. (2018), which is called Mean-Variance Optimization (MVP). Then, let us denote the SPG with the EQUM framework as EQUM. The parameter ψ is chosen from $1/3$, $2/3$, and 1. For optimizing Tamer, MVP and EQUM, we set the Adam optimizer with learning rate as 0.01 and weight decay parameter as 0.1. To train the parameters of network, we set the number of episode as 10. Each portfolio is updated by sliding one-month-ahead.

[2] https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Table 2: The performance of each portfolio models during out-of-sample period (from July 2000 to June 2020) for FF25 dataset (upper table) , FF48 (middle table) and FF100 (lower table). For each dataset, the best performance is highlighted in **bold**.

Dataset	FF25								
Method	EW	MV	EGO	BLD	Tamer	MVP	EQUM		
							$\psi = 1/3$	$\psi = 2/3$	$\psi = 1$
TR↑	191.35	25.62	209.76	134.21	286.34	215.95	**300.18**	194.12	243.93
RISK↓	18.53	25.36	19.17	**12.00**	20.21	19.28	13.42	12.19	14.95
R/R↑	0.52	0.05	0.55	0.56	0.71	0.56	**1.12**	0.80	0.82
MaxDD↓	0.54	0.75	0.57	0.37	0.58	0.57	0.31	**0.30**	0.36

Dataset	FF48								
Method	EW	MV	EGO	BLD	Tamer	MVP	EQUM		
							$\psi = 1/3$	$\psi = 2/3$	$\psi = 1$
TR↑	194.73	34.81	249.14	124.45	188.10	**333.66**	226.39	246.61	236.43
RISK↓	16.58	30.38	19.56	**10.76**	14.96	21.87	19.81	15.38	11.75
R/R↑	0.59	0.06	0.64	0.58	0.63	0.76	0.57	0.80	**1.01**
MaxDD↓	0.53	0.84	0.53	0.38	0.54	0.52	0.45	0.45	**0.29**

Dataset	FF100								
Method	EW	MV	EGO	BLD	Tamer	MVP	EQUM		
							$\psi = 1/3$	$\psi = 2/3$	$\psi = 1$
RT↑	193.91	34.26	207.37	127.92	238.27	228.53	262.54	227.04	**327.16**
RISK↓	18.77	26.36	19.72	**11.89**	14.35	31.36	13.33	13.64	15.51
R/R↑	0.52	0.06	0.53	0.54	0.83	0.36	0.98	0.83	**1.05**
MaxDD↓	0.55	0.73	0.58	0.36	0.49	0.65	**0.24**	0.31	0.39

Performance measures: The following measures widely used in finance to evaluate portfolio strategies (Brandt, 2010) are chosen. The total return (TR), annualized risk as the standard deviation of return (RISK) and risk-adjusted return (R/R) are defined as follows: $\mathrm{TR} = \sum_{t=1}^{T} y_t$, $\mathrm{RISK} = \sqrt{\frac{12}{T-1}\sum_{t=1}^{T}(y_t - \mathrm{TR}/T)^2}$, and $\mathrm{R/R} = \frac{12}{T} \times \mathrm{AR/RISK}$. Among them, R/R is the most important measure for a portfolio strategy. We also evaluate the maximum draw-down (MaxDD), which is another widely used risk measure Magdon-Ismail & Atiya (2004) for the portfolio strategy. In particular, MaxDD is the largest drop from a peak defined as $\mathrm{MaxDD} = \min_{t\in[1,T]}\left(0, \frac{W_t}{\max_{\tau\in[1,t]} W_\tau} - 1\right)$, where W_k is the cumulative return of the portfolio until time k; that is, $W_t = \prod_{t'=1}^{t}(1 + y_{t'})$.

Experimental results: Table 2 reports the performances of the portfolios. In almost all cases, the EQUM portfolio achieves the highest R/R and the lowest MaxDD. Therefore, we can confirm that the EQUM portfolio has a high R/R, and avoids a large drawdown.

5.4 Discussion

In many experimental results, contrary to our expectations, we observed that the EQUM framework also improved the expected cumulative reward, not only the variance (risk). We consider that this occurred because there is often a limit associated with the cumulative rewards that can be achieved by the standard expected cumulative reward maximizing algorithms. For instance, when the reward at each period is 1 and the discount factor is 0.99, the infinite sum is 100. In such a case, instead of naive reward maximization, MSE minimization against the target reward 100 may result in more stable performance empirically. Therefore, unless we could increase the cumulative reward infinitely, the proposed EQUM framework can stabilize the performance. From this aspect, we can confirm that the EQUM provides a regularization effect.

6 Conclusion

In this paper, we proposed an EQUM framework as a variant of MVRL. Compared with existing MVRL methods, the EQUM framework is computationally friendly. The proposed EQUM framework also includes various interpretations, such as targeting optimization and regularization, and is suitable for many real-world applications, such as finance and playing games. We investigated the effect of the EQUM framework compared with the standard RL and existing MVRL methods through the experiments.

REFERENCES

Reinforcement Learning: An Introduction. 1998.

Jonathan Baxter and Peter L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 2001.

Dimitri Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming*. 1996.

Lorenzo Bisi, Luca Sabbioni, Edoardo Vittori, Matteo Papini, and Marcello Restelli. Risk-averse trust region optimization for reward-volatility reduction. In *IJCAI*, 2020.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Michael W Brandt. Portfolio choice problems. In *Handbook of financial econometrics: Tools and techniques*. 2010.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for cvar optimization in mdps. In *NeurIPS*. 2014.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 2017.

Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The review of Financial studies*, 2007.

Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 2016.

Eugene F Fama and Kenneth R French. The cross-section of expected stock returns. *the Journal of Finance*, 1992.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 2015.

Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 2005.

J. Hull. *Options, Futures, and Other Derivatives*. 2015.

Prashanth L.A. and Mohammad Ghavamzadeh. Actor-critic algorithms for risk-sensitive mdps. In *NeurIPS*. 2013.

D.G. Luenberger. *Investment Science*. 1998.

Malik Magdon-Ismail and Amir F Atiya. Maximum drawdown. *Risk Magazine*, 2004.

Harry Markowitz. Portfolio selection. *The journal of finance*, 1952.

H.M. Markowitz, G.P. Todd, and W.F. Sharpe. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. 2000.

Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. 1995.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pp. 8024–8035. Curran Associates, Inc., 2019.

L. A. Prashanth and Mohammad Ghavamzadeh. Variance-constrained actor-critic algorithms for discounted and average reward mdps. *Machine Learning*, 2016.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Weiwei Shen and Jun Wang. Portfolio blending via thompson sampling. In *IJCAI*, 2016.

Weiwei Shen, Bin Wang, Jian Pu, and Jun Wang. The kelly growth optimal portfolio with ensemble learning. In *AAAI*, 2019.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*, 1999.

Aviv Tamar, Dotan Di Castro, and Shie Mannor. Policy gradients with variance related risk criteria. In *ICML*, 2012.

R.J. Williams. *Toward a Theory of Reinforcement-learning Connectionist Systems*. 1988.

Ronald Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 1991.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.

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 *NeurIPS*. 2018.

Shangtong Zhang, Bo Liu, and Shimon Whiteson. Mean-variance policy iteration for risk-averse reinforcement learning, 2020.

A ALTERNATIVE OBJECTIVE FUNCTIONS OF AC

In this section, we introduce alternative formulations of AC. For instance, we can also train a policy by using the following gradient:

$$\nabla_\theta \log \pi_\theta(S_t^k, A_t^k) \left\{ \left(\alpha \widehat{R}^k - \frac{1}{2}\beta \left(\widehat{R}^k \right)^2 \right) - M_{\hat{\omega}_k}(S_t^k) \right\},$$

where $M_{\hat{\omega}_k}(S_t)$ approximates $\alpha \mathbb{E}[R_{t:\infty}] + \frac{1}{2}\beta \mathbb{E}\left[R_{t:\infty}^2\right]$ with a parameter $\hat{\omega}_k$. For estimating $M^{(1)}_{\hat{\omega}_k^{(1)}}(S_t^k)$ and $M^{(2)}_{\hat{\omega}_k^{(2)}}(S_t^k)$, at a k-th critic step, we can update the parameter by solving the following problem:

$$\hat{\omega}_{k+1} = \underset{\omega \in \Theta_\omega}{\arg\min} \left(\alpha \widehat{R}^k_{t:n-1} - \frac{1}{2}\beta \left(\widehat{R}^k_{t:n-1} \right)^2 - M_\omega(S_t^k) \right)^2$$

Besides, instead of the value function estimations introduced in Section 4.3, we can estimate the parameters $\hat{\omega}^{(1)}_{k+1}$ and $\hat{\omega}^{(2)}_{k+1}$ by solving

$$\hat{\omega}^{(1)}_{k+1} = \underset{\omega \in \Theta_\omega}{\arg\min} \left(\widehat{R}^k_{t:n-1} + \gamma^n M^{(1)}_{\hat{\omega}_k^{(1)}}(S^k_{t+n}) - M^{(1)}_\omega(S_t^k) \right)^2,$$

$$\hat{\omega}^{(2)}_{k+1} = \underset{\omega \in \Theta_\omega}{\arg\min} \left(\left(\widehat{R}^k_{t:n-1} + \gamma^n M^{(1)}_{\hat{\omega}_k^{(1)}}(S^k_{t+n}) \right)^2 - M^{(2)}_\omega(S_t^k) \right)^2,$$

respectively. The main difference is use of $\gamma^n M^{(1)}_{\hat{\omega}_k^{(1)}}(S^k_{t+n})$ and $\gamma^n M^{(1)}_{\hat{\omega}_k^{(1)}}(S^k_{t+n})$. Although the AC formulation in Mnih et al. (2016) is near to the formulation introduced here, we implemented our proposed method in the experiments as introduce in Section 4.3, which follows the formulation in Pytorch example code.

B DETAILS OF EXPERIMENTS

In this section, we describe the details of the experiments.

B.1 DETAILS OF EXPERIMENTS OF EXPERIMENTS OF ATARI GAMES

Here, we show the details and other results of Atari games (Brockman et al., 2016). Among the games, we choose simplified BeamRider, Seaquest, Qbert, and SpaceInvaders in which SPG and AC methods work well, and the observations are the RAM of the Atari machine, consisting of only 128 bytes. We compare our methods with the standard SPG, AC algorithms, and MVLR methods of Tamar et al. (2012) and Xie et al. (2018). We denote the SPG-based methods proposed by Tamar et al. (2012) and Xie et al. (2018) as Tamer and MVP, respectively. We calculate the average reward (AR) at the last episode over 5 trials and the standard deviation (SD). In Table 3, we show the results of SPG-based methods. In Table 4, we show the results of SPG-based methods. We choose the parameter ψ from 0.001, 0.003, 0.005, 0.01, 0.03, and 0.3. We choose the parameter of Tamer from 200 and 2000, which is denoted as b in Tamar et al. (2012). We choose the parameter of MVP from 1 and 10, which is denoted as λ in Tamar et al. (2012). The parameters are denoted as param in Table 3. As shown in Table 4, the EQUM framework with higher ψ tends to achieve the lower SD as we expected.

B.2 DETAILS OF EXPERIMENTS OF PORTFOLIO OPTIMIZATION

We also divide the performance period into two for robustness checks. Table 5 shows the first-half results from July 2000 to June 2010 and the second-half results from July 2010 to June 2020. In almost all cases, the EQUM portfolio achieves the highest R/R.

Table 3: The results on Atari games with SPG-based methods.. The highest ARs and lowest SDs are highlighted in **bold**.

Method	SPG		EQUM with SPG												Tamer				MVP			
			$\psi = 0.001$		$\psi = 0.003$		$\psi = 0.005$		$\psi = 0.01$		$\psi = 0.03$		$\psi = 0.3$		param: 2000		param: 20000		param: 1		param: 10	
Game	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD
BeamRider	99.6	315.0	165.6	358.0	136.8	288.6	265.2	436.7	**397.2**	436.3	165.6	358.0	136.8	288.6	189.6	400.4	99.6	315.0	4.4	**13.9**	180.0	379.5
Seaquest	26.0	59.7	26.0	59.7	26.0	59.7	34.0	61.1	**50.0**	60.6	16.0	**33.7**	26.0	42.2	32.0	41.3	26.0	59.7	42.0	61.4	42.0	61.4
Qbert	40.0	64.8	15.0	47.4	**55.0**	71.5	40.0	64.8	27.5	58.3	15.0	47.4	0.0	**0.0**	40.0	64.8	15.0	47.4	50.0	64.5	50.0	64.5
SpaceInvaders	169.5	146.0	168.0	144.7	**171.0**	147.2	109.5	**141.4**	139.5	147.1	165.0	142.1	139.5	147.1	169.5	146.0	169.5	146.0	169.5	146.0	169.5	146.0

Table 4: The results on Atari games with AC-based methods. The highest ARs and lowest SDs are highlighted in **bold**.

Method	AC		EQUM with AC											
			$\psi = 0.001$		$\psi = 0.003$		$\psi = 0.005$		$\psi = 0.01$		$\psi = 0.03$		$\psi = 0.3$	
Game	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD	AR	SD
BeamRider	325.6	128.2	455.2	203.8	387.6	219.0	432.8	242.1	330.0	**108.3**	453.2	153.9	**476.0**	202.1
Seaquest	310.0	154.7	304.0	123.9	346.0	145.5	354.0	178.9	368.0	157.3	**398.0**	**119.8**	302.0	166.4
Qbert	277.5	197.4	265.0	89.1	222.5	80.3	290.0	120.9	282.5	87.4	**340.0**	185.3	257.5	**78.2**
SpaceInvaders	149.0	121.2	**230.5**	136.2	194.0	115.4	149.0	100.0	199.0	176.8	165.5	132.9	126.5	**99.1**

Table 5: The performance of each portfolio during first half out-of-sample period (from July 2000 to June 2010) and second half out-of-sample period (from July 2010 to June 2020) for FF25 dataset (upper panel) , FF48 (middle panel) and FF100 (lower panel). Among the comparisons of the various portfolios, the best performance within each dataset is highlighted in **bold**.

FF25	EW	MV	EGO	BLD	Tamer	MVP	EQUM		
							$\psi = 1/3$	$\psi = 2/3$	$\psi = 1$
First-Half Period (from July 2000 to June 2010)									
TR↑	69.36	-50.09	86.53	49.76	79.06	159.56	**186.21**	141.17	158.23
RISK↓	19.35	28.85	19.99	12.25	21.32	15.95	**11.58**	11.61	11.69
R/R↑	0.36	-0.17	0.43	0.41	0.37	1.00	**1.61**	1.22	1.35
MaxDD↓	0.54	0.75	0.57	0.37	0.33	0.31	**0.13**	0.14	0.18
Second-Half Period (from July 2000 to June 2010)									
TR↑	121.99	75.71	123.23	84.45	**126.73**	83.16	113.97	52.96	85.70
RISK↓	17.64	21.14	18.30	**11.73**	23.71	23.43	14.97	12.62	17.56
R/R↑	0.69	0.36	0.67	0.72	0.53	0.35	**0.76**	0.42	0.49
MaxDD↓	0.31	0.50	0.31	**0.22**	0.58	0.57	0.31	0.30	0.36

FF48	EW	MV	EGO	BLD	Tamer	MVP	EQUM		
							$\psi = 1/3$	$\psi = 2/3$	$\psi = 1$
First-Half Period (from July 2000 to June 2010)									
TR↑	72.02	49.11	89.90	41.26	64.99	**185.42**	118.40	127.62	163.23
RISK↓	17.61	26.27	21.72	11.48	9.03	17.21	21.49	**8.83**	8.96
R/R↑	0.41	0.19	0.41	0.36	0.72	1.08	0.55	1.45	**1.82**
MaxDD↓	0.53	0.58	0.53	0.38	0.15	0.24	0.45	0.15	**0.05**
Second-Half Period (from July 2000 to June 2010)									
TR↑	122.71	-14.30	**159.25**	83.19	123.11	148.24	107.99	118.99	73.20
RISK↓	15.45	33.97	17.06	**9.96**	19.10	25.70	17.97	19.88	13.87
R/R↑	0.79	-0.04	**0.93**	0.84	0.64	0.58	0.60	0.60	0.53
MaxDD↓	0.26	0.84	0.28	**0.17**	0.54	0.52	0.32	0.45	0.29

FF100	EW	MV	EGO	BLD	Tamer	MVP	EQUM		
							$\psi = 1/3$	$\psi = 2/3$	$\psi = 1$
First-Half Period (from July 2000 to June 2010)									
RT↑	73.25	-44.84	87.78	48.85	146.59	55.87	**215.59**	142.66	199.92
RISK↓	19.60	30.86	20.53	12.21	**11.21**	31.72	13.03	12.69	13.11
R/R↑	0.37	-0.15	0.43	0.40	1.31	0.18	**1.65**	1.12	1.52
MaxDD↓	0.55	0.73	0.58	0.36	0.18	0.46	**0.11**	0.15	0.16
Second-Half Period (from July 2000 to June 2010)									
TR↑	120.66	79.10	119.59	79.07	91.69	**172.66**	46.95	84.39	127.25
RISK↓	17.87	20.76	18.86	**11.55**	16.88	30.91	13.18	14.47	17.52
R/R↑	0.68	0.38	0.63	0.68	0.54	0.56	0.36	0.58	**0.73**
MaxDD↓	0.32	0.35	0.31	**0.22**	0.49	0.65	0.24	0.31	0.39