STAT 900 Final Report

Supervisors: Prof. Bin Li and Prof. David Saunders

Acknowledgement: Prof. Paul Marriott and Prof. Jock MacKay

Author: Yuling Max Chen

Department of Statistics,
University of Waterloo

Spring 2023

1 Introduction

Stochastic Control (SC) and Reinforcement Learning (RL) are two major streams of research in statistics and computer science respectively, that deal with sequential decision-making problems. This report is motivated by a seminal paper in the RL literature, (Gullapalli, 1990), that introduces a Stochastic RL Algorithm. The spirit of Stochastic RL has extended the classical SC problems to an *exploratory* SC formulation (Wang and Zhou, 2020), which is particularly appreciated in the financial applications. Standing on the shoulders of the giants, this report answers a simple research question via a simulation study, based on the 2 aforementioned papers.

In the following sections, we first summarize the key spirit of (Gullapalli, 1990) in Section 2, and elaborate a literature review in Section 3. Then, in Section 4, we propose a simple research question and answer it through a simulation study based on (Wang and Zhou, 2020). Lastly, we make some concluding remarks in Section 5. Relevant figures, tables, algorithm blocks and math derivations are included in the Appendix. The R source code is developed to a R package and has been made available on Github.

2 Stochastic Reinforcement Learning Algorithm: A Summary

Reinforcement Learning (RL) is a paradigm of machine learning that has only partial access to the target output, in contrast to the supervised learning that fully relies on the target output as the *teacher* of the model's output. The *teacher* of RL only gives evaluative (rather than instructional) feedback on the model's output (Gullapalli, 1990), according to some performance measure.

RL typically solves a sequential decision-making task within a dynamic environment, as illustrated in Figure 1. At time t and *state* S_t, an *Agent* takes an *action* a_t following *policy* π, and the environment (*teacher*) gives feedback to the *Agent* in terms of *reward* R_t. Once receiving R_t, the *Agent* will move to the next state S_{t+1}. The *Agent*'s goal is to maximize the cumulative *reward* over time. Most RL works before (Gullapalli, 1990) focused on the *discrete action* problem, where decisions are chosen from a finite set of actions. However, (Gullapalli, 1990) points out that we may face infinitely many actions in reality. For example, the money a trader decides to invest in the stock market everyday is a continuous variable and can take infinitely many different values.

Motivated by this fact, (Gullapalli, 1990) proposes to use a Gaussian distribution as the policy π. The action a_t is thereby a random variable sampled from this Gaussian

distribution, which is continuous. A stochastic policy also allows for more exploration at the blind points within the action space, leading to potential improvements of the policy. This reflects the *exploitation-exploration trade-off* (Sutton and Barto, 2018), where *exploitation* is simply taking the best known action and *exploration* is attempting new actions that may potentially be even better. Since stochasticity is introduced to the RL policy, this approach is known as **Stochastic Reinforcement Learning** (SRL).

The SRL model proposed by Gullapalli (1990) is illustrated in Figure 2, which is an advance of the previous work (Barto and Anandan, 1985). A Gaussian distribution has 2 parameters, mean (μ) and standard deviation (σ), which respectively account for *exploitation* and *exploration* of the policy. (Gullapalli, 1990) uses 2 neural networks to compute μ and σ respectively, taking the state $\{S_t := (S_1(t), \cdots, S_n(t)) \in \mathbb{R}^n\}_{t=0}^T$ as input. [1] Neural network $\mu_w(t)$ is parameterized by $\boldsymbol{w}(t) = (w_1(t), \cdots, w_n(t), w_{bias}(t))$ and estimates μ:

$$\mu_t := \mu_w(t) = \sum_{i=1}^{n} w_i(t) S_i(t) + w_{bias}(t) \tag{1}$$

On the other hand, neural network, $\hat{R}_v(t)$, is parameterized by $\boldsymbol{v}(t) = (v_1(t), \cdots, v_n(t), v_{bias}(t))$ and estimates the expected reward $\hat{R}_t$ at a given state S_t:

$$\hat{R}_t := \hat{R}_v(t) = \sum_{i=1}^{n} v_i(t) S_i(t) + v_{bias}(t) \tag{2}$$

σ is computed by passing $\hat{R}_t$ to a monotonically decreasing and non-negative function $s(\cdot)$,

$$\sigma_t := \sigma_v(t) = s(\hat{R}_v(t)) \tag{3}$$

This is because we are less incentivized to explore new actions when we are already at a state with high expected reward.

By computing (μ_t, σ_t) via Equation 1-3, the policy is constructed as a Gaussian distribution $\pi_t := N(\mu_t, \sigma_t)$, from which the action a_t is sampled. Then, reward R_t is received from the environment, which is used to update the parameters of the neural networks and improve the policy. The updating scheme for w is as follows, and v is updated similarly:

$$\begin{aligned} w_i(t+1) &= w_i(t) + \alpha \Delta_w(t) S_i(t), \text{ for } i = 1, \cdots, n \\ w_{bias}(t+1) &= w_{bias}(t) + \alpha \Delta_w(t) \end{aligned} \tag{4}$$

$\alpha > 0$ is the learning rate and $\Delta_w(t) := (R_t - \hat{R}_t) \left(\frac{a_t - \mu_t}{\sigma_t} \right)$ is the updating step. The fraction in $\Delta_w(t)$ is the *normalized noise*, which updates mean μ_t in the direction of the realized action a_t, if the realized reward R_t is higher than the expected reward $\hat{R}_t$.

[1] Here $n > 0$ is the dimension of the state space and $T > 0$ is the total time steps.

As time $t \in \{1, 2, \cdots, T\}$ changes, the neural networks are updated at different states $\{S_t\}_{t=1}^T$ (Equation 4) and the policy $\{\pi_t\}_{t=1}^T$ improves correspondingly. This results in different actions $\{a_t\}_{t=1}^T$ sampled from different Gaussian distributions $\{N(\mu_t, \sigma_t)\}_{t=1}^T$.

The rest of (Gullapalli, 1990) focuses on implementation and training of the aforementioned SRL model, and performs convergence analysis. The major contribution is the introduction of Gaussian policy distribution, that outputs continuous actions and allows for exploration in the action space. However, the choice of Gaussian distribution isn't explicitly justified, which motivates us to dive deeper into the literature to find the answer.

3 Literature Review

In this section, we walk through the historical advances in Stochastic Control (SC) and Reinforcement Learning (RL) separately in Section 3.1 and 3.2, and we discuss in Section 3.3 on how they recently merged into a new research direction in the financial application.

Remark *State* and *action* can be viewed as 2 stochastic processes. The *state* is the instantaneous status of the environment. An *Agent* takes an *action* at each given *state* over time. Besides, *control strategy* in SC literature is analogous to *policy* in RL literature, which is the law an *Agent* follows to take *actions*. For instance, a trader (*Agent*) make investment (*action*) every day given the stock prices on the market (*state*), following some trading strategy (*policy*). We hereafter use *state*, *action* and *policy* without further explanation.

3.1 The Development of Stochastic Control (SC)

Control theory traces back to the 19th century, when Maxwell (1868) established the theoretical foundation of control engineering. Pioneered by (Bellman, 1961), SC is a problem of finding a control strategy that optimizes a prespecified objective in a dynamic environment. Based on Dynamic Programming Principle (DPP) (Bellman, 1966), we can solve a SC problem by solving a Hamilton-Jacobi-Bellman (HJB) equation. Under certain convergence conditions, the solution to an HJB equation exists uniquely (Pham, 2009).

In pre-2000, researchers mainly focused on analytical approaches to "simpler" problems, i.e., problems with discrete or lower-dimensional state, discrete finite action and discrete time (Wittenmark, 1975; Kumar, 1985). Post 2000, numerical solvers of Partial Differential Equations (PDE) (Larsson and Thomée, 2003) were developed to tackle "harder" problems (Rao, 2009). Such "harder" problems involve high state dimensionality, continuous and/or stochastic action and continuous time, where an analytical solution to the HJB equation

is usually not attainable. More recently, machine learning techniques and neural networks also play an important role in the SC literature (Hu and Lauriere, 2023).

Textbooks (e.g., Pham (2009)) cover both continuous- and discrete-time SC with financial applications, while others (e.g., Bertsekas (2015)) focus on DPP and neural network approximated dynamic programming techniques in optimal control problems.

3.2 The Development of Reinforcement Learning (RL)

RL dates back to the physiologists' study in the human and animal behavior (Skinner, 1938), and it has attracted increasing research attentions since late 1980s (Kaelbling et al., 1996). Before 2000, most of the classical RL methods (e.g., *Q-Learning*) were developed (Kaelbling et al., 1996). After 2000, the significant influence of deep learning lead to vast advances of deep RL models (e.g., *Deep Q-Network*) (Arulkumaran et al., 2017). This is because deep neural networks are good function approximators and can easily compute gradient in Python packages like PyTorch (Paszke et al., 2019). Besides, practitioners also actively apply RL techniques to solve financial problems (Sato, 2019).

Among various taxonomies of RL models, Sato (2019) categorized RL into *value-based models* and *policy-based models*. *Value-based models* focus on understanding the environment and estimating the expected value of the state-action pairs (s, a). The optimal policy is thereby inferred as taking the action that maximizes the expected value of the state-action pair $Q(s, a)$. For instance, in *Q-learning* (Watkins and Dayan, 1992), the optimal action is $a^* := \arg\max_a Q(s, a)$. *Policy-based models*, however, directly focus on learning the optimal policy. For example, *Policy Gradient RL* (Sutton et al., 1999) conducts a search in the policy space and updates its policy in the direction that lead to higher rewards, via gradient ascent optimization method.

(Sutton and Barto, 2018) is a good textbook for beginners, while other machine learning textbooks also significantly remark RL models (e.g., Dixon et al. (2020)).

3.3 SC under RL Framework: A Financial Application

The two research streams, SC and RL, merged in the financial application. Wang and Zhou (2020) proposed a RL framework for SC problems, that solves an *exploratory, entropy regularized, Mean-Variance (MV) portfolio optimization* problem in continuous time. The MV problem (Markowitz, 1952) is a classical finance problem. It aims to find the trading strategy that achieves the highest expected *wealth* with lowest *risk*, where the expected *wealth* and *risk* are respectively measured by the mean and variance of the portfolio value.

3.3.1 Formulation of the Exploratory Mean-Variance (EMV) Problem

For simplicity, consider a tiny market that consists of only one (risky) stock and one (riskfree) bond. Let $T > 0$ be the investment period and $\{W_t\}_{t=0}^T$ be a 1-dimensional standard Brownian Motion defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, \{\mathcal{F}_t\}_{t\in[0,T]})$ that satisfies the usual conditions. The state of this dynamic environment (i.e., market) involves the price processes of the stock $\{S_t\}_{t=0}^T$ and the bond $\{B_t\}_{t=0}^T$:

$$dS_t = S_t(\mu dt + \sigma dW_t), \quad dB_t = B_t r dt, \quad 0 \le t \le T \tag{5}$$

where $(\mu \in \mathbb{R}, \sigma > 0)$ are the mean and volatility of the annualized stock returns , and $r > 0$ is the riskfree interest rate. The *Sharpe ratio* of the risky asset (i.e., stock) is $\rho := \frac{\mu - r}{\sigma}$, which measures the risk-adjusted return. Larger Sharpe ratio means the investment achieves higher return with lower risk-bearing, which is desirable for investors.

The *wealth* (i.e., portfolio value), is denoted by $\{X_t^u\}_{t=0}^T$, where the superscript u means the trader follows policy $\{u_t\}_{t=0}^T$, i.e., at each time t, the trader invests u_t dollars in stock and $X_t - u_t$ dollars in bond. If $u_t < 0$, the trader is *short-selling*, i.e., sell the stock and buy more bond. If $u_t > X_t$, the trader is *trading with leverage*, i.e., borrowing money and buy more stock. With *initial wealth* $X_0 = x_0$, the wealth process is therefore driven by:

$$dX_t^u = \sigma u_t(\rho dt + dW_t), 0 \le t \le T \tag{6}$$

In the **classical Mean-Variance (MV) problem**, the trader solves a constrained optimization problem, given z as a prespecified target level of the expected terminal wealth:

$$\min_u Var(X_T^u) \text{ subject to } \mathbb{E}[X_T^u] = z \tag{7}$$

Wang and Zhou (2020) reformulates Equation 7 to an unconstrained Lagrangian optimization with Lagrange multiplier ω:

$$\min_u \mathbb{E}[(X_T^u - \omega)^2] - (\omega - z)^2 \tag{8}$$

To enable exploration in the action space, (Wang and Zhou, 2020) introduces the *exploratory policy* $\boldsymbol{\pi} := \{\pi_t\}_{t=0}^T$, where $\pi_t : \mathbb{R} \mapsto \mathbb{P}(\mathbb{R})$ is a probability distribution over all actions $u \in \mathbb{R}$ at time t. The wealth process (Equation 6) is thereby extended to:

$$dX_t^\pi = \left(\int_{\mathbb{R}} \rho\sigma u \pi_t(u) du\right) dt + \left(\sqrt{\int_{\mathbb{R}} \sigma^2 u^2 \pi_t(u) du}\right) dW_t \tag{9}$$

Lastly, to encourage exploration and regularize the family of policy distribution, (Wang and Zhou, 2020) derives Equation 8 to the **entropy-regularized Exploratory Mean-Variance (EMV)** problem that solves for the following optimal value function:

$$V(s,y;\omega) := \inf_{\pi\in\mathbb{P}(\mathbb{R})} \mathbb{E}\left[(X_T^\pi - \omega)^2 + \lambda \int_0^T \int_{\mathbb{R}} \pi_t(u)\log\pi_t(u)dudt \middle| X_s^\pi = y\right] - (\omega - z)^2 \quad (10)$$

for $(s,y) \in [0,T)\times\mathbb{R}$. $\lambda > 0$ is the *exploration weight* that controls the *Exploitation-Exploration trade-off* and $\mathcal{H}(\boldsymbol{\pi}) := -\int_0^T \int_{\mathbb{R}} \pi_t(u)\log\pi_t(u)dudt$ is the *entropy* that measures the amount of uncertainty covered by policy distribution $\boldsymbol{\pi}$ and encourages exploration.

3.3.2 Solving EMV for the Optimal Policy

Following standard DPP argument, (Wang and Zhou, 2020) derives Equation 10 to the following entropy-regularized HJB equation:

$$v_t(t,x;\omega) + \min_{\boldsymbol{\pi}\in\mathbb{P}(\mathbb{R})} \int_{\mathbb{R}} \left(\frac{1}{2}\sigma^2 u^2 v_{xx}(t,x;\omega) + \rho\sigma u v_x(t,x,;\omega) + \lambda\log\boldsymbol{\pi}(u)\right)\boldsymbol{\pi}(u)du = 0 \quad (11)$$

This analytically solves for the **optimal exploratory policy** as a Gaussian distribution:

$$\pi^*(u;t,x,\omega) = N\left(u \middle| -\frac{\rho}{\sigma}(x-\omega), \frac{\lambda}{2\sigma^2}e^{\rho^2(T-t)}\right), \text{ with } \omega = \frac{ze^{\rho^2(T-t)} - x_0}{e^{\rho^2 T} - 1} \quad (12)$$

This is a direct result of adapting the entropy regularization ($\mathcal{H}(\boldsymbol{\pi})$) in the EMV problem. It justifies the choice of Gaussian policy distribution, which is left unclear in (Gullapalli, 1990) in Section 2. The corresponding **optimal value function** is given by:

$$V(t,x;\omega) = (x-\omega)^2 e^{-\rho^2(T-t)} + \frac{\lambda\rho^2}{4}(T^2 - t^2) - \frac{\lambda}{2}\left(\rho^2 T - \log\frac{\sigma^2}{\boldsymbol{\pi}\lambda}\right)(T-t) - (\omega - z)^2 \quad (13)$$

It's noteworthy that the mean of the optimal policy distribution in Equation 12 is exactly the solution u^* of the classical Mean-Variance problem in Equation 8, and the variance is controlled by the exploration weight λ. By passing $\lambda \to 0$, (Wang and Zhou, 2020) shows that the optimal exploratory policy π^* converges to the optimal classical policy $u^* = -\frac{\rho}{\sigma}(x-\omega)$. The detailed derivation of solving the EMV problem is given in Appendix.

The rest of the paper proves the **Policy Improvement Theorem (PIT)** [2], showing the effectiveness of policy learning and objective optimization under their proposed framework. The paper also proposes an iterative RL algorithm that learns the optimal policy

[2] (Sutton and Barto, 2018) clearly states this theorem.

via gradient-based optimization. The major contributions of (Wang and Zhou, 2020) are the adaption of RL in SC problem and the formulation of an exploratory SC framework, which has already influenced the subsequent literature (e.g., Jiang et al. (2022))

(Wang and Zhou, 2020) also conducted a simulation study to compare their proposed RL algorithm with other methods, which we discuss in more detail in the next section.

4 Simulation Study

One naive way to estimate the optimal exploratory policy (Equation 12) is to use Maximum Likelihood Estimator (MLE) to estimate (μ, σ) in Equation 5. However, the well-documented *mean-blur problem* (Luenberger, 2013) indicates that the MLE of mean μ is poor, due to the practical infeasibility of collecting enough historical data to reduce the estimation error. Thus, (Wang and Zhou, 2020) proposes an iterative RL algorithm that skips the estimation of (μ, σ) and directly solves the EMV problem (Equation 10) for the optimal exploratory policy. Since investors care more about about their terminal wealth at the end of a trading period, we propose a **simple research question** as follows:

> **"With constant but unknown (μ, σ), does the RL-based optimal Exploratory Mean-Variance (EMV) policy increases the mean and reduces the standard-deviation of the distribution of terminal wealth X_T, compared to the MLE-based optimal classical Mean-Variance (MV) policy?"**

Following the simulation design pipeline in (Morris et al., 2019), we describe the data-generator in Section 4.1. We introduce the MLE estimation approach in Section 4.2 and the RL Algorithm in 4.3. Lastly, we present the simulation results in 4.4.

4.1 Data Generator: Stock Price Simulator

We consider a $T = 1$ year investment period with 252 trading days. Hence the continuous time is discretized with $\Delta t = \frac{1}{252}$. The stock price paths are simulated from a geometric Brownian Motion (Equation 5) with $(\mu, \sigma) = (-0.3, 0.2)$. They are so-chosen to be closed to the realized mean and volatility of the S&P500 market index annualized returns in year 2008 [3], mimicking a bearish market. The riskfree interest rate is set as $r = 2\%$. See Algorithm 1 for details. A simulation of stock price paths $\{S_t^{(i)}\}_{0 \le t \le T}^{i=1,\cdots,N_{sim}}$ of size $N_{sim} = 100$ are illustrated in Figure 3, with the initial stock price $S_0 = 1$.

[3] According to historical data from Yahoo Finance.

4.2 MLE Approach for Classical MV Problem

Given the simulated stock price paths $\{S_t^{(i)}\}_{0\le t\le T}^{i=1,\cdots,N_{sim}}$, the MLE approach for estimating (μ,σ) is explained in (Campbell et al., 1997). Applying Itô's formula to $\log S_t$ and substituting dS_t (Equation 5) yields:

$$d(\log S_t) = \left(\mu - \frac{1}{2}\sigma^2\right)dt + \sigma dW_t \tag{14}$$

Therefore, the daily log-return $\{R_t := \log\frac{S_{t+\Delta t}}{S_t}\}_{0\le t\le T-\Delta t}$ follows a Normal distribution with mean $(\mu - \frac{1}{2}\sigma^2)\Delta t$ and standard deviation $\sigma\sqrt{\Delta t}$. So, the MLE of (μ,σ) are:

$$\hat{\sigma}_{MLE} = \sqrt{\frac{1}{N_{window}\Delta t}\sum_{t=1}^{N_{window}}\left(R_t - \bar{R}_t\right)^2}, \text{ with } \bar{R}_t = \frac{1}{N_{window}}\sum_{t=1}^{N_{window}} R_t \tag{15}$$

$$\hat{\mu}_{MLE} = \frac{\bar{R}_t}{\Delta t} + \frac{1}{2}\hat{\sigma}_{MLE}^2 \tag{16}$$

where N_{window} is the rolling window size. For each time t, we use the last $N_{window} = 100$ daily returns to estimate parameters $(\hat{\mu}_{MLE}, \hat{\sigma}_{MLE})$ and compute the Sharpe ratio $\hat{\rho} = \frac{\hat{\mu}_{MLE}-r}{\hat{\sigma}_{MLE}}$. Then, substitute them in the optimal classical MV policy gives the investment action $u_t^* = -\frac{\hat{\rho}}{\hat{\sigma}_{MLE}}(X_t - \omega)$, where X_t is the current wealth and $\omega = \frac{ze^{\hat{\rho}^2(T-t)} - x_0}{e^{\hat{\rho}^2 T}-1}$ is the Lagrange multiplier (from Equation 12). Then the next-day-wealth is computed by:

$$X_{t+\Delta t} = u_t^* R_t + (X_t - u_t^*)(1 + r\Delta t) \tag{17}$$

Repeating this iteratively until $t = T - \Delta t$ for N_{sim} times gives me a simulation of terminal wealth $\{X_{T,MLE}^{(i)}\}_{i=1}^{N_{sim}}$. See Algorithm 2 for details.

4.3 RL Algorithm for EMV Problem

The RL algorithm in (Wang and Zhou, 2020) reparametrizes the optimal value function (Equation 13) and the EMV policy (Equation 12) with $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3) \in \mathbb{R}^4$ and $\boldsymbol{\phi} = (\phi_1, \phi_2) \in \mathbb{R} \times \mathbb{R}_+$ respectively:

$$V^{\theta}(t,x) = (x-\omega)^2 e^{-\theta_3(T-t)} + \theta_2 t^2 + \theta_1 t + \theta_0, \text{ with } (t,x) \in [0,T]\times\mathbb{R} \tag{18}$$

$$\pi^{\phi}(u;t,x,\omega) = N\left(\sqrt{\frac{2\phi_2}{\lambda\pi}}e^{\phi_1 - \frac{1}{2}}(x-\omega), \frac{1}{2\pi}e^{2\phi_2(T-t)+2\phi_1-1}\right) \tag{19}$$

While Equation 18 is easy to see from Equation 13, Equation 19 is in fact a result of reparameterizing the entropy by $\mathcal{H}(\boldsymbol{\pi}_t^{\phi}) = \phi_1 + \phi_2(T-t)$.

The objective of this RL algorithm is to minimize the following Temporal Difference (TD) Error (Sutton and Barto, 2018), derived from Equation 10:

$$TD(\theta, \phi) = \frac{1}{2} \sum_{t=0}^{T-\Delta t} \left(\frac{V^{\theta}(t+\Delta t, X_{t+\Delta t}) - V^{\theta}(t, X_t)}{\Delta t} - \lambda(\phi_1 + \phi_2(T-t)) \right)^2 \Delta t \qquad (20)$$

A detailed RL algorithm is presented in Algorithm 3 and derivation is in Appendix. Once the training loop converges, we substitute the latest updated $\boldsymbol{\phi}$ in Equation 19 to construct the optimal exploratory policy, and sample investment actions from it. Then, following the same iterative procedure as in Equation 17, we can simulate terminal wealth $\{X_{T,RL}^{(i)}\}_{i=1}^{N_{sim}}$.

4.4 Performance Evaluation

The performance evaluation is two-folded. We first examine the convergence of the RL algorithm described in Section 4.3. Figure 4 shows that the TD Error converges within 10000 training epochs. This is consistent to the convergence of parameters as illustrated in Figure 5 and 6. While most of the parameters converges within 10000 epochs, θ_2 does not show a sign of stopping. However, we still consider this as a "good convergence", because we only care about the optimal exploratory policy π^{ϕ}, which only depends on $\boldsymbol{\phi}$. Figure 5 shows that both (ϕ_1, ϕ_2) have already converged.

The mean and standard deviation of the terminal wealth simulations are summarized in Table 1, along with their Monte Carlo Simulation Errors (MCSE). With initial wealth $x_0 = 1$ and target terminal wealth $z = 1.4$, we compare 4 trading strategies (in Table 1):

1. Strategy *"MLE"* follows the MLE-based classical MV policy ($u_t^* = -\frac{\hat{\rho}}{\hat{\sigma}_{MLE}}(X_t - \omega)$) with $(\hat{\mu}_{MLE}, \hat{\sigma}_{MLE})$ defined in Equation 16 and 15 respectively;

2. Strategy *"MLE (constrained)"* also follows the MLE-based classical MV policy, but with a constraint: $|u_t^*| < 5$. Because in practice, traders cannot invest wildly in the market. The constraint means that the trader can at most *leverage* (borrow) $\$4x_0$ from the bank to buy stock, or *short* (sell) $\$4x_0$ worth of stock to buy bond;

3. Strategy *"MLE (exploratory)"* follows the optimal EMV policy (Equation 12), with the parameters estimated by their MLEs $(\hat{\mu}_{MLE}, \hat{\sigma}_{MLE})$ (in Equation 15-16);

4. Strategy *"RL"* follows the RL-based reparameterized EMV policy (Equation 19), with $\boldsymbol{\phi}$ optimized via the RL algorithm (Algorithm 3).

For each trading strategy, we conduct 50 runs of simulations with each simulation-run of size $N_{sim} = 100$, resulting in 5000 terminal wealths.

Strategy "*MLE*" performs very poor, with the mean terminal wealth at -6.533 and standard-deviation at 12613.83%, indicating a negative annualized return of $-553.3\% = (-6.533 - 1) \times 100\%$ and a Sharpe ratio closed to 0. This is because the *mean-blur* problem causes very volatile estimates of $\hat{\mu}_{MLE}$, which further leads to large fluctuation in the optimal classical policy u_t^*. The resulting MCSE is extremely large.

Strategy "*MLE (constrained)*" and "*MLE (exploratory)*" give similar and improved performance —— higher mean annualized return $(10\% - 30\%)$, higher Sharpe ratio $(1.5 - 2.6)$ and lower standard deviation $(50\% - 75\%)$. Strategy "*RL*" achieves the best performance, as the mean annualized return $(36\% = (1.36 - 1) \times 100\%)$ is the highest and closest to the target return $(40\% = \frac{z - x_0}{x_0} \times 100\%)$. Its performance is also the most stable with the lowest standard-deviation (25.4%), resulting in the highest Sharpe ratio (5.3). The resulting MCSEs of the 3 strategies are all reasonably small (< 0.08).

All of the boxplot (Figure 7), empirical CDF (ECDF) plot (Figure 9) and density plot (Figure 10) are consistent with our observation. While the MLE-based strategies can achieve a relatively higher mean terminal wealth with the constraint on the daily investment, the distribution of simulated terminal wealth still has fat tails compared to that of the RL-based strategy. All MLE-based strategies are more likely to take extreme investment actions. Table 2 contains the quantiles of the terminal wealth simulations. Strategy *"MLE"* gives the most turbulent performance. In the downside market scenarios (i.e., at quantiles 1%, 5%, 10%), strategy *"RL"* always outperforms the others. For instance, in the worst 1% scenario, strategy *"RL"* only lost $25\% = (0.75 - 1) \times 100\%$ of the initial wealth, while all other strategies have lost all the money (i.e., terminal wealth < 0).

5 Concluding Discussion

Based on the simulation study, we can answer a big "YES" to the simple research question in Section 4. That is, the RL-based EMV investment policy can achieve higher expected terminal wealth with lower volatility, compared to the MLE-based classical policy.

With unknown (μ, σ), the RL algorithm bypasses the parameter estimation and avoids the *mean-blur* problem, resulting in more reliable investment performance. However, this benefit comes at a cost — the RL training algorithm requires researchers to be particularly cautious in the choice of initial parameter values, learning rates and gradient computation techniques, which Wang and Zhou (2020) didn't fully disclose in their paper. Empirically, a "good" choice will lead to faster convergence and avoid returning Not-a-Number errors.

References

Arulkumaran, K., M. P. Deisenroth, M. Brundage, and A. A. Bharath (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine 34*(6), 26–38.

Barto, A. G. and P. Anandan (1985). Pattern-recognizing stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics* (3), 360–375.

Bellman, R. (1961). Adaptive control processes-a guided tour. *Princeton Legacy Library 2045*, 1–258.

Bellman, R. (1966). Dynamic programming. *Science 153*(3731), 34–37.

Bertsekas, D. P. (2015). Dynamic programming and optimal control 4th edition, volume ii. *Athena Scientific*.

Campbell, J. Y., A. W. Lo, and A. MacKinlay (1997). *The Econometrics of Financial Markets*. Princeton University Press.

Dixon, M. F., I. Halperin, and P. Bilokon (2020). *Machine learning in Finance*, Volume 1170. Springer.

Gullapalli, V. (1990). A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural networks 3*(6), 671–692.

Hu, R. and M. Lauriere (2023). Recent developments in machine learning methods for stochastic control and games. *arXiv preprint arXiv:2303.10257*.

Jiang, R., D. Saunders, and C. Weng (2022). The reinforcement learning kelly strategy. *Quantitative Finance 22*(8), 1445–1464.

Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research 4*, 237–285.

Kumar, P. R. (1985). A survey of some results in stochastic adaptive control. *SIAM Journal on Control and Optimization 23*(3), 329–380.

Larsson, S. and V. Thomée (2003). *Partial differential equations with numerical methods*, Volume 45. Springer.

Luenberger, D. G. (2013). *Investment Science* (Second ed.). Oxford University Press.

Markowitz, H. (1952). Portfolio selection*. *The Journal of Finance 7*(1), 77–91.

Maxwell, J. C. (1868). I. on governors. *Proceedings of the Royal Society of London* (16), 270–283.

Morris, T. P., I. R. White, and M. J. Crowther (2019). Using simulation studies to evaluate statistical methods. *Statistics in medicine 38*(11), 2074–2102.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc.

Pham, H. (2009). *Continuous-time stochastic control and optimization with financial applications*, Volume 61. Springer Science & Business Media.

Rao, A. V. (2009). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences 135*(1), 497–528.

Sato, Y. (2019). Model-free reinforcement learning for financial portfolios: a brief survey. *arXiv preprint arXiv:1904.04973*.

Skinner, B. F. (1938). *The behavior of organisms: An experimental analysis*. Appleton-Century.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement Learning: An Introduction* (Second ed.). The MIT Press.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour (1999). Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems 12*.

Wang, H. and X. Y. Zhou (2020). Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance 30*(4), 1273–1308.

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

Wittenmark, B. (1975). Stochastic adaptive control methods: a survey. *International Journal of Control 21*(5), 705–730.

I. Figures

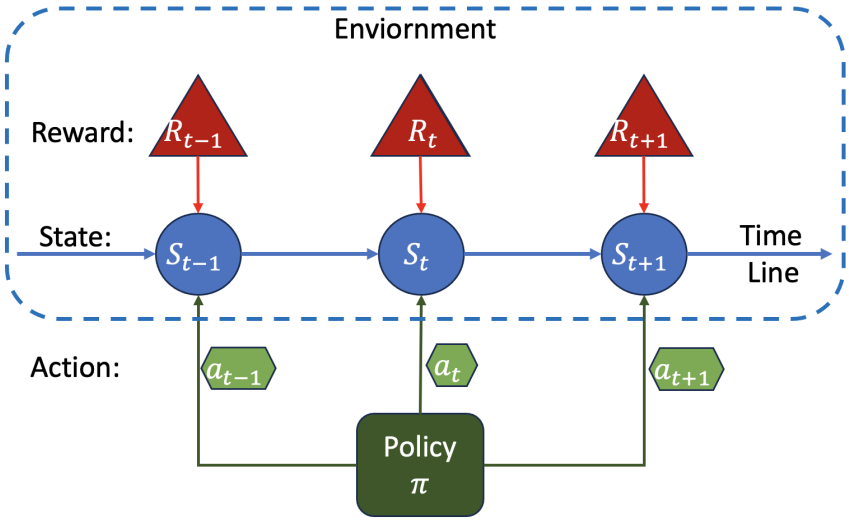

Figure 1: Reinforcement Learning (RL) workflow

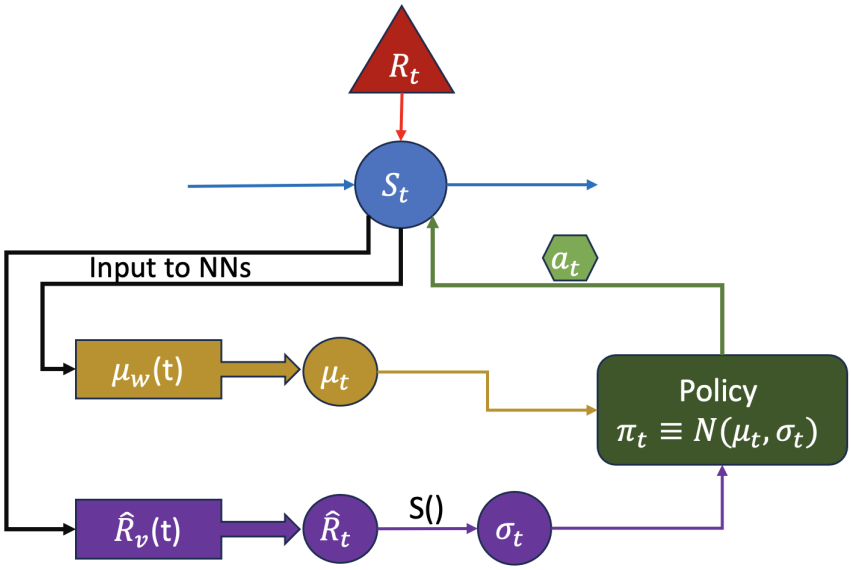

Figure 2: Stochastic Reinforcement Learning (SRL) model proposed in Gullapalli (1990)

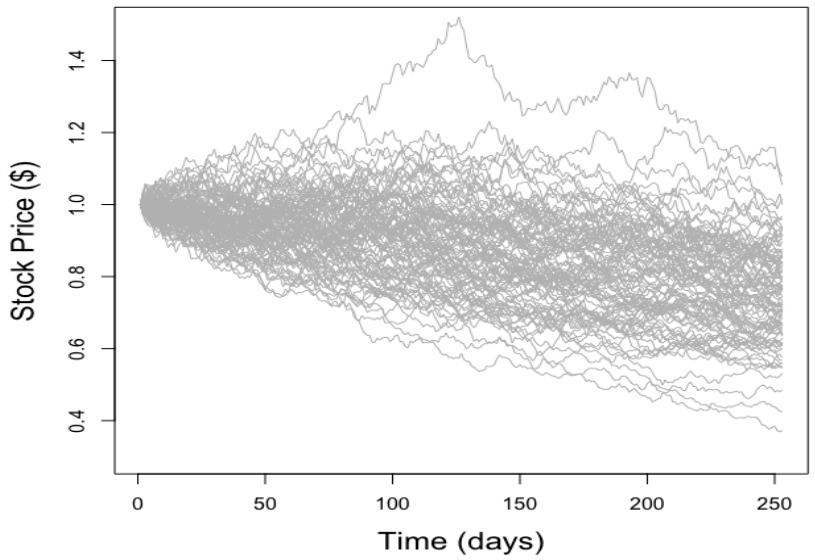

Figure 3: Simulated stock price path following Algorithm 1 with $S_0 = 1$ and $N_{window} = 0$

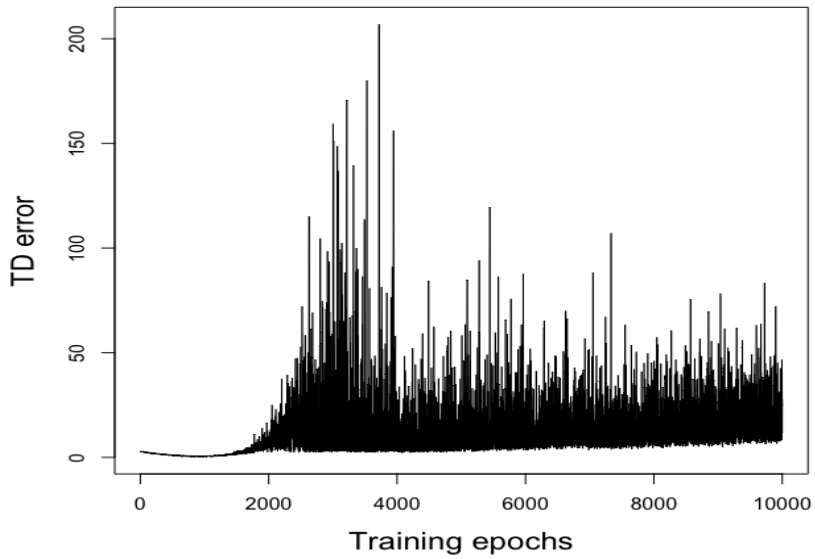

Figure 4: Temporal Difference (TD) error over training epochs following Algorithm 3.

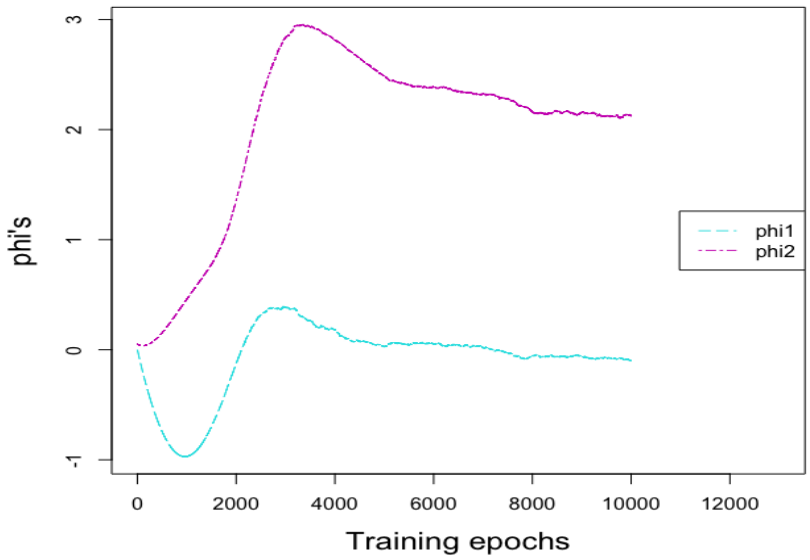

Figure 5: Convergence of the exploratory policy (Equation 19) parameters $\boldsymbol{\phi} = (\phi_1, \phi_2)$.

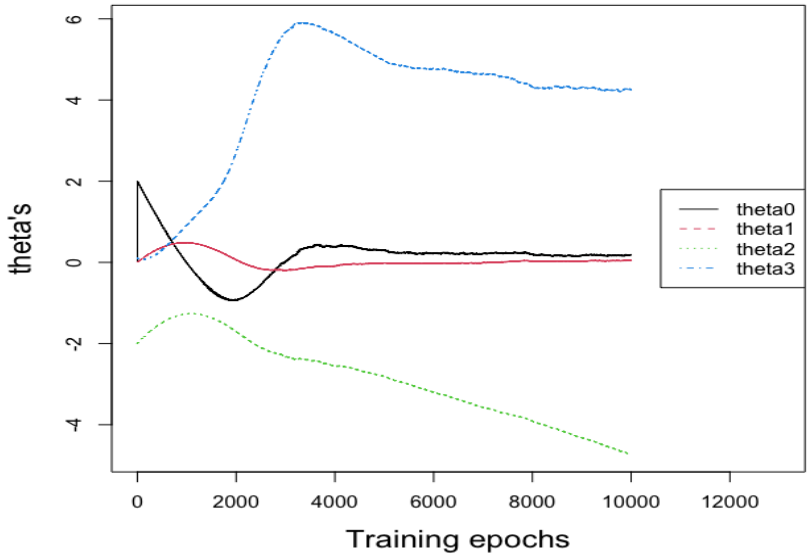

Figure 6: Convergence of the value function (Equation 18) parameters $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$.

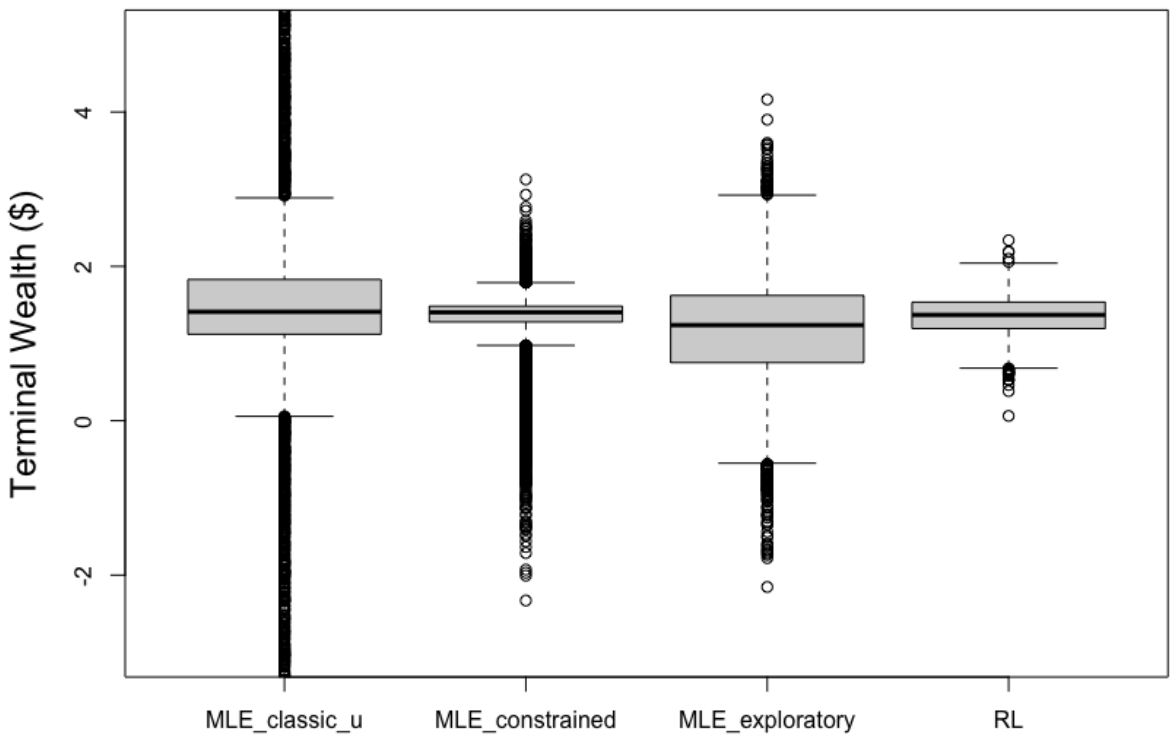

Figure 7: Boxplot of the terminal wealth simulations following the MLE approach and the RL algorithm. As described in Section 4.4, 'MLE_classic_u' is the Strategy *"MLE"*, 'MLE_constrained' is the Strategy *"MLE (constrained)"*, 'MLE_exploratory' is the Strategy *"MLE (exploratory)"* and 'RL' is the Strategy *"RL"*.

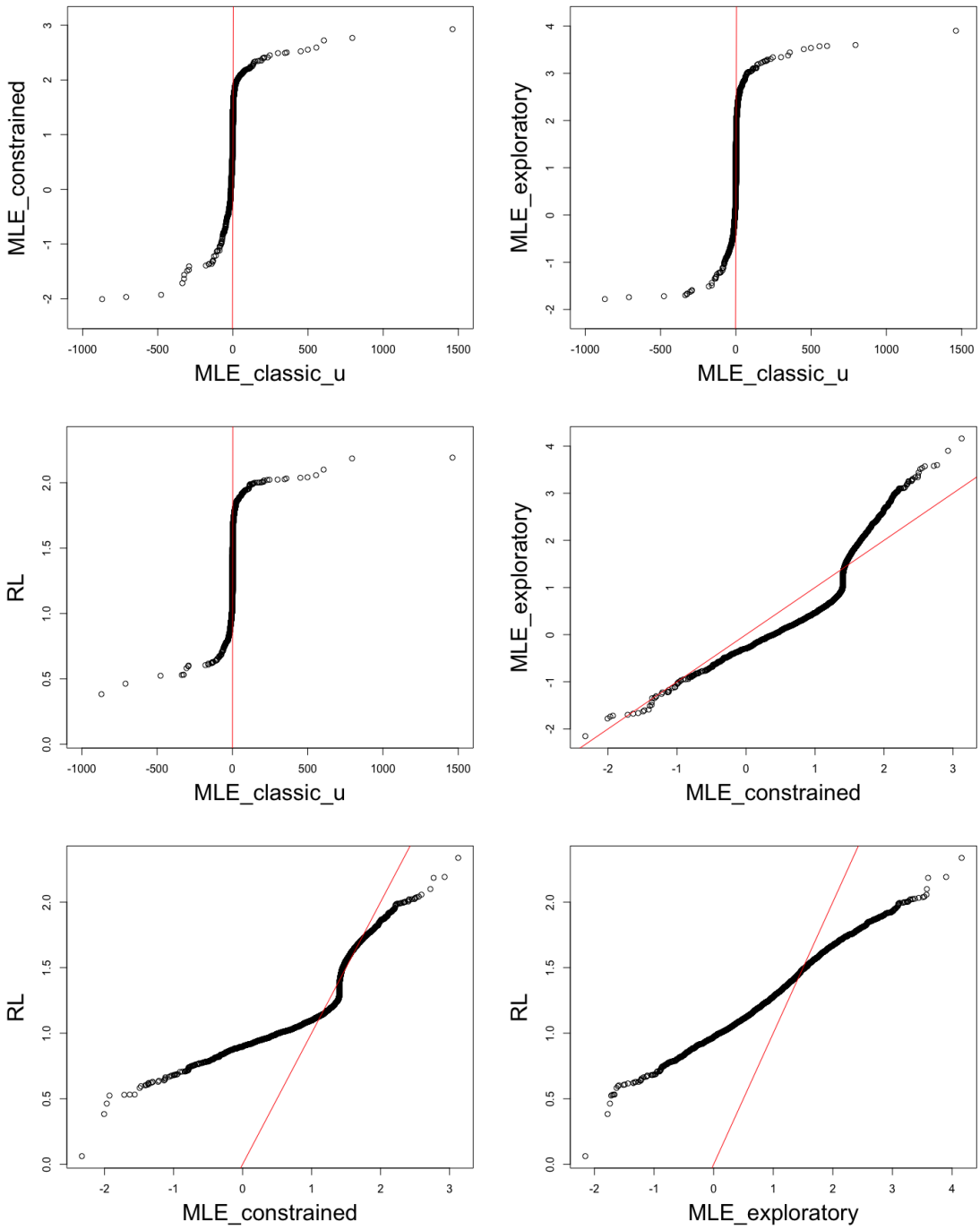

Figure 8: QQplots of the terminal wealth simulations following the MLE approach and the RL algorithm.

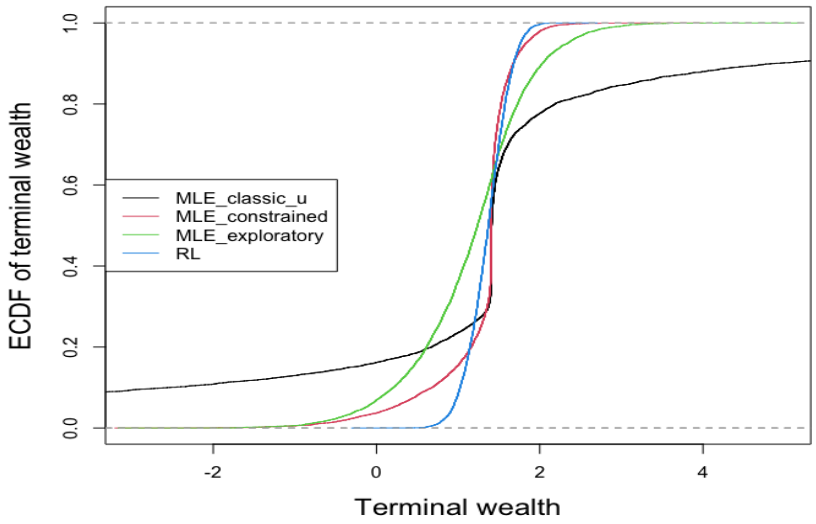

Figure 9: ECDF of the terminal wealth simulations following the MLE approach and the RL algorithm.

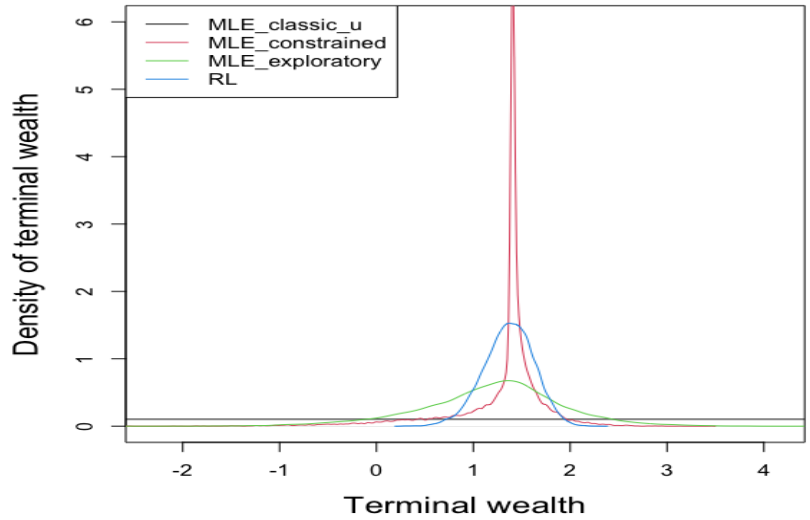

Figure 10: Density of the terminal wealth simulations following the MLE approach and the RL algorithm.

II. Tables

	Mean	MCSE	Stddev	MCSE	Sharpe Ratio	MCSE
MLE	-6.533	64.670	12613.83%	642.302	0.091	0.112
MLE (constrained)	1.286	0.044	50.412%	0.068	2.567	0.430
MLE (exploratory)	1.172	0.078	73.965 %	0.054	1.567	0.166
RL	**1.360**	0.0243	**25.382%**	0.015	**5.300**	0.337

Table 1: Mean and standard-deviation of the simulated terminal wealth, following the MLE approach and the RL algorithm

	1%	5%	10%	25%	50%	75%	95%	99%
MLE	-54.041	-9.729	-2.534	1.122	**1.411**	**1.830**	**11.719**	**61.960**
MLE (constrained)	-0.731	0.196	0.680	**1.280**	1.404	1.484	1.829	2.113
MLE (exploratory)	-0.833	-0.140	0.210	0.753	1.239	1.623	2.345	2.886
RL	**0.750**	**0.935**	**1.024**	1.194	1.371	1.537	1.763	1.913

Table 2: Quantiles of the simulated terminal wealth, following the MLE approach and the RL algorithm

III. Algorithm

Algorithm 1: Simulating a stock price path

Initialize: $\mu = -30\%, \sigma = 20\%$, investment period $T = 1$, time discretization $\Delta t = \frac{1}{252}$, rolling window size $N_{window} = 100$, initial stock price $S_0 = 1$.
Define $N := \frac{T}{\Delta t} + N_{window}$.
Define $t_{N_{window}} = 0$ as the initial time.
Define $t_N = T$ as the terminal time.
Define $\{t_i\}_{i=1}^{N_{window}}$ as the historical time points.
for $i = 1, \cdots, N$ **do**
  Sample $dW_i \sim N(0, \Delta t)$
  $S_{t_i} = S_{t_{i-1}} \cdot \exp((\mu - \frac{1}{2}\sigma^2)\Delta t + \sigma dW_i)$
end
Output: $\{S_{t_i}\}_{i=1}^{N}$

Algorithm 2: Simulating terminal wealth under the classical Mean-Variance policy with MLE approach

Initialize: Number of simulations $N_{sim} = 1000$, target expected terminal wealth $z = 1.4$, investment period $T = 1$, time discretization $\Delta t = \frac{1}{252}$, rolling window size $N_{window} = 100$, initial wealth $X_0 = 1$, riskfree interest rate $r = 0.02$, MLE policy constraint $C = 5$.

for $n = 1, \cdots, N_{sim}$ **do**

Simulate a stock price path $\{S_{t_i}\}_{i=0}^{N}$ via Algorithm 1.

Compute daily log-returns $R_{t_i} := \log \frac{S_{t_{i+1}}}{S_{t_i}}$, for $i = 1, \cdots, N-1$.

for $i = N_{window}, \cdots, N-1$ **do**

Compute the annualized rolling volatility $\hat{\sigma}_{t_i}$ and the annualized mean log-return $\hat{\mu}_{t_i}$ using Equation 15 and 16, with the summation replaced by $\sum_{t=t_{i-N_{window}}}^{t_i}$.

Compute the Sharpe ratio as $\hat{\rho}_{t_i} = \frac{\hat{\mu}_{t_i} - r}{\hat{\sigma}_{t_i}}$.

Compute the Lagrange multiplier $w_{t_i} = \frac{z e^{\hat{\rho}_{t_i}^2 (T-t)} - x_0}{e^{\hat{\rho}_{t_i}^2 T} - 1}$.

Compute the optimal classical Mean-Variance policy $u_{t_i}^* = -\frac{\hat{\rho}_{t+i}}{\hat{\sigma}_{t_i}}(X_{t_i} - w_{t_i})$.

if *constrain the policy* [a] **then**

$u_{t_i}^* = \text{sign}(u_{t_i}^*) \cdot \min(|u_{t_i}^*|, C)$

end

Compute the next-day wealth: $X_{t_{i+1}} = u_{t_i}^* R_{t_i} + (X_{t_i} - u_{t_i}^*)(1 + r\Delta t)$ by Equation 17

end

Get the terminal wealth of the n-th simulation: $X_T^{(n)} := X_{t_N} = X_T$.

end

Output: Terminal wealth of n simulations: $\{X_T^{(n)}\}_{n=1}^{N_{sim}}$.

[a] We specify in advance whether we want to constrain the daily investment u_t or not.

Algorithm 3: Solving the Exploratory Mean-Variance (EMV) problem via Reinforcement Learning (Wang and Zhou, 2020)

Initialize: Parameters of the value function $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3) = (0, 0, -2, 0.1)$, parameters of the exploratory policy $\boldsymbol{\phi} = (\phi_1, \phi_2) = (0, 0.05)$, Lagrange multiplier $\omega = 1.4$, learning rates $(\alpha, \eta_\theta, \eta_\phi) = (0.05, 0.0005, 0.0005)$, investment period $T = 1$, time discretization $\Delta t = \frac{1}{252}$, sample size $N = \frac{T}{\Delta t} = 252$, target expected terminal wealth $z = 1.4$, exploration weight $\lambda = 2$, number of training epochs $N_{train} = 10000$, number of steps to update Lagrange multiplier $w_{step} = 20$.

for $k = 1, \cdots, N_{train}$ **do**

Simulate a stock price path $\{S_{t_i}\}_{i=1}^N$ via Algorithm 1 with $N_{window} = 0$.

for $i = 1, \cdots, N$ **do**

Sample investment action u_{t_i} from the exploratory policy π^ϕ in Equation 19.

Compute the next-day wealth $X_{t_{i+1}} = u^*_{t_i} R_{t_i} + (X_{t_i} - u^*_{t_i})(1 + r\Delta t)$

end

Collect the time and wealth process $\mathcal{D} := \{(t_i, X_{t_i})\}_{i=1}^N$.

Compute the following partial derivatives of objective TD error in Equation 20:

$$\frac{\partial TD}{\partial \theta_1}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{(t_i, X_{t_i}) \in \mathcal{D}} \left(\frac{V^\theta(t_{i+1}, X_{t_{i+1}}) - V^\theta(t_i, X_{t_i})}{\Delta t} - \lambda(\phi_1 + \phi_2(T - t_i)) \right) \Delta t$$

$$\frac{\partial TD}{\partial \theta_2}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{(t_i, X_{t_i}) \in \mathcal{D}} \left(\frac{V^\theta(t_{i+1}, X_{t_{i+1}}) - V^\theta(t_i, X_{t_i})}{\Delta t} - \lambda(\phi_1 + \phi_2(T - t_i)) \right) (t_{i+1}^2 - t_i^2)$$

$$\frac{\partial TD}{\partial \phi_1}(\boldsymbol{\theta}, \boldsymbol{\phi}) = -\lambda \sum_{(t_i, X_{t_i}) \in \mathcal{D}} \left(\frac{V^\theta(t_{i+1}, X_{t_{i+1}}) - V^\theta(t_i, X_{t_i})}{\Delta t} - \lambda(\phi_1 + \phi_2(T - t_i)) \right) \Delta t$$

$$\begin{aligned} \frac{\partial TD}{\partial \phi_2}(\boldsymbol{\theta}, \boldsymbol{\phi}) = & \sum_{(t_i, X_{t_i}) \in \mathcal{D}} \left(\frac{V^\theta(t_{i+1}, X_{t_{i+1}}) - V^\theta(t_i, X_{t_i})}{\Delta t} - \lambda(\phi_1 + \phi_2(T - t_i)) \right) \Delta t \\ & \times \left(- \frac{2(X_{t_{i+1}} - \omega)^2 e^{-2\phi_2(T - t_{i+1})}(T - t_{i+1}) - 2(X_{t_i} - \omega)^2 e^{-2\phi_2(T - t_i)}(T - t_i)}{\Delta t} - \lambda(T - t_i) \right) \end{aligned}$$

Update $(\theta_1, \theta_2) \leftarrow (\theta_1, \theta_2) - \eta_\theta(\frac{\partial TD}{\partial \theta_1}, \frac{\partial TD}{\partial \theta_2})$.

Update $\theta_0 \leftarrow -\theta_2 T^2 - \theta_1 T - (\omega - z)^2$ and $\theta_3 \leftarrow 2\phi_2$.

Update $(\phi_1, \phi_2) \leftarrow (\phi_1, \phi_2) - \eta_\phi(\frac{\partial TD}{\partial \phi_1}, \frac{\partial TD}{\partial \phi_2})$, update policy π^ϕ via Equation 19.

Collect the terminal wealth of the k-th epoch: $X_T^{(k)} := X_{t_N}$.

Update $\omega \leftarrow \omega - \alpha \left(\frac{1}{w_{step}} \sum_{j=k-w_{step}+1}^{k} X_T^{(j)} - z \right)$, if $k \mod w_{step} = 0$.

end

IV. Derivations

Under the same setup as described in Section 3.3.1, let $T > 0$ be the investment period and $\{W_t\}_{t=0}^T$ be a 1-dimensional standard Brownian Motion defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}, \{\mathcal{F}_t\}_{t\in[0,T]})$ that satisfies the usual conditions. The price processes of the stock $\{S_t\}_{t=0}^T$ and the bond $\{B_t\}_{t=0}^T$ are:

$$dS_t = S_t(\mu dt + \sigma dW_t), \quad 0 \le t \le T \tag{21}$$

$$dB_t = B_t r dt, 0 \le t \le T \tag{22}$$

where $(\mu \in \mathbb{R}, \sigma > 0)$ are the mean and volatility of the stock log-return $\{R_t := \log \frac{S_{t+1}}{S_t}\}_{t=0}^T$ respectively, and $r > 0$ is the riskfree interest rate. *Sharpe ratio* is $\rho := \frac{\mu - r}{\sigma}$.

(A) Classical Continuous-Time Mean-Variance Problem

The action $u_t \in \mathbb{R}$ is the dollar value that the trader puts in the stock at time t, and $X_t - u_t$ is thereby the money spent in the bond. The *wealth* (i.e., portfolio value), $\{X_t^u\}_{t=0}^T$, is computed by

$$X_{t+\Delta t} = \frac{u_t}{S_t}(S_{t+\Delta t} - S_T) + (X_t - u_t) r \Delta t$$

Taking $\Delta t \to 0$ and applying Equation 21 give the following dynamic:

$$dX_t^u = \sigma u_t(\rho dt + dW_t), 0 \le t \le T \tag{23}$$

The Classical Continuous-Time Mean-Variance (MV) Problem is:

$$\min_u Var(X_T^u) \text{ subject to } \mathbb{E}[X_T^u] = z \tag{24}$$

where z is a prespecified target level of the expected terminal wealth. Using Lagrangian optimization with Lagrange multiplier 2ω, we can reformulate it to:

$$\min_u \mathbb{E}[(X_T^u)^2] - z^2 - 2\omega(\mathbb{E}(X_T^u) - z) \implies \min_u \mathbb{E}[(X_T^u - \omega)^2] - (\omega - z)^2 \tag{25}$$

(B) Deriving Solution to the Exploratory Mean-Variance (EMV) Problem

The action u_t is extended to a stochastic action $\pi_t : \mathbb{R} \mapsto \mathbb{P}(\mathbb{R})$, which is a probability distribution over all the actions u_t at time t. At each time, an action is in fact sampled from this distribution, i.e., $u \sim \pi_t$.

This extends the wealth dynamic (Equation 23) to:

$$dX_t^\pi = \underbrace{\left(\int_{\mathbb{R}} \rho\sigma u \pi_t(u) du\right)}_{\tilde{b}(\pi_t)} dt + \underbrace{\left(\sqrt{\int_{\mathbb{R}} \sigma^2 u^2 \pi_t(u) du}\right)}_{\tilde{\sigma}(\pi_t)} dW_t \tag{26}$$

where $\tilde{b}(\pi_t) = \mathbb{E}_\pi(\rho\sigma u)$ and $\tilde{\sigma}(\pi_t) = \mathbb{E}_\pi(\sigma^2 u^2)$.

The objective function Equation 25 is also extended to:

$$\min_{\pi \in \mathbb{P}(\mathbb{R})} \mathbb{E}\left[(X_T^\pi - \omega)^2 + \lambda \int_0^T \int_{\mathbb{R}} \pi_t(u) \log \pi_t(u) du dt \middle| X_s^\pi = y\right] - (\omega - z)^2 \tag{27}$$

with $\lambda > 0$ and the entropy regularization:

$$\mathcal{H}(\boldsymbol{\pi}) := -\int_0^T \int_{\mathbb{R}} \pi_t(u) \log \pi_t(u) du dt \tag{28}$$

The entropy term forces exploration in the action space, because any distribution that puts zero mass on some actions in the action space will lead to $\mathcal{H}(\boldsymbol{\pi}) = \infty$. For a fixed ω, we define the value function:

$$V(s, y; \omega) := \inf_{\pi \in \mathbb{P}(\mathbb{R})} \mathbb{E}\left[(X_T^\pi - \omega)^2 + \lambda \int_0^T \int_{\mathbb{R}} \pi_t(u) \log \pi_t(u) du dt \middle| X_s^\pi = y\right] - (\omega - z)^2 \tag{29}$$

for $(s, y) \in [0, T] \times \mathbb{R}$.

By Bellman's Principle of Optimally, for $0 \leq t \leq s \leq T$, we can rewrite Equation 29 in a recursive form:

$$V(t, x; \omega) = \inf_{\pi \in \mathbb{P}(\mathbb{R})} \mathbb{E}\left[V(s, X_s^{\pi,t,x}; \omega) + \lambda \int_t^s \int_{\mathbb{R}} \pi_k(u) \log \pi_k(u) du dk \middle| X_t^\pi = x\right] \tag{30}$$

where $X_s^{\pi,t,x}$ is the wealth at time s given that it started from $X_t^\pi = x$ at time t, following policy π. Then, moving the left-hand-side (LHS) to the right and taking $s \to t$ yield:

$$\begin{aligned}
&\lim_{s \to t} \inf_{\pi \in \mathbb{P}(\mathbb{R})} \mathbb{E}\left[V(s, X_s^{\pi,t,x}; \omega) - V(t, x; \omega) + \lambda \int_t^s \int_{\mathbb{R}} \pi_k(u) \log \pi_k(u) du dk \middle| X_t^\pi = x\right] = 0 \\
&\implies \inf_{\pi \in \mathbb{P}(\mathbb{R})} d(V(t, x; \omega)) + \lambda \int_{\mathbb{R}} \pi_t(u) \log \pi_t(u) du = 0
\end{aligned}$$

Applying Itô's formula, we can derive the Hamilton-Jacobi-Bellman (HJB) equation:

$$v_t(t,x;\omega)+\min_{\boldsymbol{\pi}\in\mathbb{P}(\mathbb{R})}\int_{\mathbb{R}}\left(\frac{1}{2}\sigma^2u^2v_{xx}(t,x;\omega)+\rho\sigma uv_x(t,x,;\omega)+\lambda\log\boldsymbol{\pi}(u)\right)\boldsymbol{\pi}(u)du=0 \quad (31)$$

with the terminal condition $v(T,x;\omega)=(x-\omega)^2-(\omega-z)^2$.

Solving Equation 31 for π gives the optimal policy that follows Gaussian distribution:

$$\pi^*(u;t,x,\omega)=\frac{\exp\left(-\frac{1}{\lambda}\left(\frac{1}{2}\sigma^2u^2v_{xx}(t,x;\omega)+\rho\sigma uv_x(t,x;\omega)\right)\right)}{\int_{\mathbb{R}}\exp\left(-\frac{1}{\lambda}\left(\frac{1}{2}\sigma^2u^2v_{xx}(t,x;\omega)+\rho\sigma uv_x(t,x;\omega)\right)\right)du} \quad (32)$$

$$=N\left(u\,\middle|\,-\frac{\rho}{\sigma}\frac{v_x(t,x;\omega)}{v_{xx}(t,x;\omega)},\sqrt{\frac{\lambda}{\sigma^2v_{xx}(t,x;\omega)}}\right) \quad (33)$$

Substituting $\pi^*(u;t,x,\omega)$ back to the HJB Equation 31, we get:

$$v_t-\frac{\rho^2}{2}\frac{v_x^2(t,x;\omega)}{v_{xx}(t,x;\omega)}+\frac{\lambda}{2}\left(1-\log\frac{2\pi e\lambda}{\sigma^2v_{xx}(t,x;\omega)}\right)=0 \quad (34)$$

because by the Gaussian policy distribution (Equation 33),

$$\int_{\mathbb{R}}\rho\sigma v_xu\pi(u)du=\rho\sigma v_x\cdot\mathbb{E}_\pi(u)=\frac{\rho^2}{2}\frac{v_x^2}{v_{xx}}$$

$$\int_{\mathbb{R}}\frac{1}{2}\sigma^2u^2v_{xx}\pi(u)du=\frac{1}{2}\sigma^2v_{xx}\mathbb{E}_\pi(u^2)=\frac{1}{2}\sigma^2v_{xx}\left(\frac{\lambda}{\sigma^2v_{xx}}+\frac{\rho^2v_x^2}{\sigma^2v_{xx}^2}\right)=\frac{\lambda}{2}+\frac{\rho^2v_x^2}{2v_{xx}}$$

$$\mathcal{H}(\pi)=-\int_{\mathbb{R}}(\log\pi(u))\pi(u)du=\frac{1}{2}\log\left(\frac{2\pi e\lambda}{\sigma^2v_{xx}}\right),\ \text{by the entropy of Gaussian.}$$

One can check that the solution to Equation 34 is:

$$V(t,x;\omega)=(x-\omega)^2e^{-\rho^2(T-t)}+\frac{\lambda\rho^2}{4}(T^2-t^2)-\frac{\lambda}{2}\left(\rho^2T-\log\frac{\sigma^2}{\boldsymbol{\pi}\lambda}\right)(T-t)-(\omega-z)^2 \quad (35)$$

which we call it the **optimal value function** in Equation 13. Then, substitute it in Equation 33 yields the **optimal exploratory policy**:

$$\pi^*(u;t,x,\omega)=N\left(u\,\middle|\,-\frac{\rho}{\sigma}(x-\omega),\frac{\lambda}{2\sigma^2}e^{\rho^2(T-t)}\right),\ \text{with}\ (t,x)\in[0,T]\times\mathbb{R} \quad (36)$$

The Lagrangian multiplier can be solved by the terminal condition. Following the optimal exploratory policy (Equation 36), the wealth process (Equation 26) now becomes:

$$dX_t^* = -\rho^2(X_t^* - \omega)dt + \sqrt{\rho^2(X_t^* - \omega)^2 + \frac{\lambda}{2}e^{\rho^2(T-t)}}dW_t$$

So, by Fubini's Theorem,

$$\begin{aligned}
&\mathbb{E}[X_t^*] = x_0 + \mathbb{E}\left[\int_0^t -\rho^2(X_s^* - \omega)ds\right] = x_0 + \int_0^t -\rho^2(\mathbb{E}[X_s^*] - \omega)ds \\
&\implies \underbrace{\mathbb{E}[X_t^*] - \omega}_{M_t} = \underbrace{x_0 - \omega}_{M_0} + \int_0^t -\rho^2 \underbrace{(\mathbb{E}[X_s^*] - \omega)}_{M_s} ds \\
&\implies dM_t = -\rho^2 M_t dt \implies M_t = M_0 e^{-\rho^2 t} \\
&\implies \mathbb{E}[X_t^*] = (x_0 - \omega)e^{-\rho^2 t} + \omega
\end{aligned}$$

Then, the terminal condition, $E[X_T^*] = (x_0 - \omega)e^{-\rho^2 T} + \omega = z$, implies $\omega = \frac{ze^{\rho^2(T-t)} - x_0}{e^{\rho^2 T} - 1}$.

(C) Parameterization for the Reinforcement Learning (RL) Algorithm

By Equation 35, the value function can be reparameterized by $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$:

$$V^\theta(t,x) = (x - \omega)^2 e^{-\theta_3(T-t)} + \theta_2 t^2 + \theta_1 t + \theta_0, \text{ with } (t,x) \in [0,T] \times \mathbb{R} \tag{37}$$

Note that the entropy takes the following form and can be parameterized by $\boldsymbol{\phi} = (\phi_1, \phi_2)$:

$$\begin{aligned}
H(\boldsymbol{\pi}) &= \frac{1}{2}\log\left(\frac{2\pi e\lambda}{\sigma^2 v_{xx}}\right) \\
&= \frac{1}{2}\left(\log\left(\frac{2\pi e\lambda}{\sigma^2}\right) + \log\frac{1}{v_{xx}}\right) = \frac{1}{2}\log\left(\frac{2\pi e\lambda}{\sigma^2}\right) + \frac{1}{2}\log\left(\frac{1}{2}e^{\theta_3(T-t)}\right) \\
&= \frac{1}{2}\log\left(\frac{\pi e\lambda}{\sigma^2}\right) + \frac{1}{2}\theta_3(T-t) \\
&= \phi_1 + \phi_2(T-t)
\end{aligned}$$

Equating the last 2 lines gives:

$$\sigma^2 = \lambda\boldsymbol{\pi}e^{1-2\phi_1} \quad \text{and} \quad \theta_3 = 2\phi_2 = \rho^2$$

Substituting this into the optimal exploratory policy (Equation 36) yields the **reparameterized RL-based optimal exploratory policy**:

$$\pi^{\phi}(u;t,x,\omega) = N\left(\sqrt{\frac{2\phi_2}{\lambda\pi}}e^{\phi_1-\frac{1}{2}}(x-\omega), \frac{1}{2\pi}e^{2\phi_2(T-t)+2\phi_1-1}\right) \tag{38}$$

where we assume the true (unknown) ρ is negative.

(D) Deriving the Temporal Difference (TD) Error

Denote V^{π} as the value function follow policy $\boldsymbol{\pi}$. Then, from the recursive form of the value function (Equation 30), we have:

$$V^{\pi}(t,x) = \mathbb{E}\left[V^{\pi}(s,X_s) + \lambda\int_t^s\int_{\mathbb{R}}\pi_k(u)\log\pi_k(u)dudk\middle| X_t = x\right], \quad s\in[t,T] \tag{39}$$

Moving the left-hand-size to the right and dividing both sizes by $s-t$ yields:

$$\mathbb{E}\left[\frac{V^{\pi}(s,X_s)-V^{\pi}(t,x)}{s-t} + \frac{\lambda}{s-t}\int_t^s\int_{\mathbb{R}}\pi_k(u)\log\pi_k(u)dudk\middle| X_t = x\right] = 0$$

Taking $s\to t$, then the left-hand-side (inside the expectation) is the TD error:

$$TD_t := \dot{V}_t^{\pi} + \lambda\int_{\mathbb{R}}\pi_k(u)\log\pi_k(u)du, \text{ with } \dot{V}_t^{\pi} = \frac{V^{\pi}(t+\Delta t, X_{t+\Delta t})-V^{\pi}(t,x)}{\Delta t}$$

The RL algorithm as described in Section 4.3 was designed to minimize the TD error.