Executive Summary:

In this project, we split into the analytical part and reinforcement learning (RL) part. The analytical solution is deriving the result from dynamic programming approach, which serves as a benchmark for us to crosscheck whether the result in the RL part. Whereas in the RL part, our team focuses on 2 branches of Temporal-difference learning, Q-learning and SARSA learning for allocation of wealth.

We also compared Q-learning and SARSA learning to traditional benchmarks and considered that both ML policies yields a similar result to optimal strategies, but more robust to changing environments.

Analytical Solution:

Starting from the Constant Absolute Risk-aversion (CARA) function, we have the below utility function for the last step $U: \mathbb{R} \to \mathbb{R}$, parameterized by $\mathcal{A} \in \mathbb{R}$, where a is a constant:

$$U(W_T) = \begin{cases} \dfrac{1 - e^{-\mathcal{A}W_T}}{\mathcal{A}}, & for\ a\ \neq 0 \\ W_T, & for\ a = 0 \end{cases}$$

We will focus on the case $\mathcal{A} \neq 0$.

Our goal is to maximize, for each $t = 0,1, \dots, T-1$, over choices of $x_t \in \mathbb{R}$, the value of

$$\mathbb{E}[\gamma^{T-t} \times \frac{1 - e^{-\mathcal{A}W_T}}{\mathcal{A}} \mid (t, W_t)]$$

Which is equivalent to maximizing the below value since discount factor γ and $\mathcal{A}$ are constant

$$\mathbb{E}[\frac{-e^{-\mathcal{A}W_T}}{a} \mid (t, W_t)]$$

Following the setting of formulating the above problem as a continuous state and continuous actions discrete-time finite-horizon MDP, we follow the notations:

- State $s_t \in S_t$ at any time step $t = 0,1, \dots, T-1$ consist of the wealth W_t
- Action $a_t \in A_t$ at any time step $t = 0,1, \dots, T-1$ indicates the quantity of

investment in risky asset $(= x_t)$, quantity of riskless asset $= W_t - x_t$

- Deterministic policy at time t as $\pi_t(W_t) = x_t$, with the optimal policy denoted as π_t^*, and $\pi_t^*(W_t) = x_t^*$
- Random Variable Y_t for the single-time-step return of the risky return, where Y_t has the below binomial distribution:

$$Y_t = \begin{cases} a, & probability\ of\ p \\ b, & probability\ of\ (1-p) \end{cases}$$

- Riskless return of r which is constant throughout the steps
- MDP reward for $t = T$ is $\frac{-e^{-\mathcal{A}W_T}}{\mathcal{A}}$, and 0 for all other time steps

We then have the wealth W_{t+1} has the below form:

$$W_{t+1} = (W_t - x_t) \cdot (1 + r) + x_t \cdot (1 + Y_t) = x_t \cdot (1 - r) = x_t \cdot (Y_t - r) + W_t \cdot (1 + r)$$

For all $t = 0,1, \dots, T-1$

We denote Value function at time t for all t, $t = 0,1, \dots, T-1$ for a given policy π as:

$$V_t^{\pi}(W_t) = \mathbb{E}_{\pi}[\frac{-e^{-\mathcal{A}W_T}}{\mathcal{A}} \mid (t, W_t)]$$

And the optimal value function for all time t as:

$$V_t^*(W_t) = \max_{\pi} V_t^{\pi}(W_t) = \max_{\pi} \mathbb{E}_{\pi}[\frac{-e^{-\mathcal{A}W_T}}{\mathcal{A}} \mid (t, W_t)]$$

The Bellman Optimality Equation is:

$$V_t^*(W_t) = \max_{x_t} Q_t^*(W_t, x_t) = \max_{x_t}\{\mathbb{E}_{Y_t}[\frac{-e^{-\mathcal{A}W_T}}{\mathcal{A}}]\}$$

for all $t = 0,1, \dots, T-2$, and

$$V_{T-1}^*(W_{T-1}) = \max_{x_{T-1}} Q_{T-1}^*(W_{T-1}, x_{T-1}) = \max_{x_{T-1}}\{\mathbb{E}_{Y_{T-1}}[\frac{-e^{-\mathcal{A}W_T}}{\mathcal{A}}]\}$$

Where Q_t^* is the Optimal Action-Value function.

An educated guess for the Optimal Value function $V_t^*(W_t)$ have the below functional form:

$$V_t^*(W_t) = -c_t \cdot e^{-d_t \cdot W_t}$$

where b_t, c_t are independent of the wealth W_t for all $t = 0,1, \dots, T-1$. Then the Bellman Optimality Equation can be expressed as:

$$V_t^*(W_t) = \max_{x_t}\{\mathbb{E}_{Y_t}[-c_{t+1} \cdot e^{-d_{t+1} \cdot W_{t+1}}]\} = \max_{x_t}\{\mathbb{E}_{Y_t}[-c_{t+1} \cdot e^{-d_{t+1} \cdot (x_t \cdot (Y_t - r) + W_t \cdot (1+r))}]\}$$

$$= \max_{x_t}\{-c_{t+1} \cdot e^{-d_{t+1} \cdot (W_t \cdot (1+r) - r \cdot x_t)}\left(p \cdot e^{-d_{t+1} \cdot x_t \cdot a} + (1-p) \cdot e^{-d_{t+1} \cdot x_t \cdot b}\right)\}$$

$$= \max_{x_t}\{-c_{t+1} \cdot e^{-d_{t+1} \cdot (W_t \cdot (1+r) - r \cdot x_t)} \cdot e^{-d_{t+1} x_t \cdot b}\,(1 - p + pe^{-d_{t+1} \cdot x_t (a-b)})\}$$

Which we can infer the functional form for $Q_t^*(W_t)$

$$Q_t^*(W_t) = -c_{t+1} \cdot e^{-d_{t+1} \cdot (W_t \cdot (1+r) - r \cdot x_t + x_t \cdot b)}(1 - p + pe^{-d_{t+1} \cdot x_t (a-b)})$$

The maximum is of $Q_t^*(W_t, x_t)$ is reached when we set the partial derivative of $Q_t^*(W_t, x_t)$ w.r.t. x_t = 0

$$\frac{\partial Q_t^*(W_t)}{\partial x_t} = 0$$

$$\Rightarrow -c_{t+1} \cdot e^{-d_{t+1} \cdot (W_t \cdot (1+r) - r \cdot x_t + x_t \cdot b)} \cdot d_{t+1}(r-b) \cdot (1-p) - c_{t+1} \cdot e^{-d_{t+1} \cdot (W_t \cdot (1+r) - r \cdot x_t + x_t \cdot a)} \cdot d_{t+1}(r-a) \cdot p = 0$$

$$\Rightarrow e^{-d_{t+1} \cdot x_t \cdot b}(r-b) \cdot (1-p) + e^{-d_{t+1} \cdot x_t \cdot a}(r-a)p = 0$$

$$\Rightarrow x_t^* = \frac{1}{d_{t+1}(a-b)} \ln\left(\frac{p(a-r)}{(1-p)(r-b)}\right)$$

We substitute x_t^* into the Bellman Optimality equation and we have

$$V_t^*(W_t) = -c_{t+1} \cdot e^{-d_{t+1} \cdot (W_t \cdot (1+r) - (r-b) \cdot x_t^*)}\left(1 - p + pe^{-d_{t+1} \cdot x_t^* (a-b)}\right)$$

$$= -c_{t+1} \cdot \left(\frac{p(a-r)}{(1-p)(r-b)}\right)^{\frac{r-b}{a-b}} \cdot \frac{(1-p)(a-b)}{a-r} \cdot e^{-d_{t+1}(W_t(1+r))}$$

And we also have the functional form of

$$V_t^*(W_t) = -c_t \cdot e^{-d_t \cdot W_t}$$

By comparing match terms, we have

$$c_t = c_{t+1} \cdot \left(\frac{p(a-r)}{(1-p)(r-b)}\right)^{\frac{r-b}{a-b}} \frac{(1-p)(a-b)}{a-r}$$

$$d_t = d_{t+1} \cdot (1+r)$$

Recursively, we can calculate c_{T-1} and d_{T-1} based on the knowledge of the final step MDP reward of $\frac{-e^{-\mathcal{A}W_T}}{\mathcal{A}}$ at time = T.

$$V_{T-1}^*(W_{T-1}) = \max_{x_{T-1}}\left\{\mathbb{E}_{Y_{T-1}}\left[\frac{-e^{-\mathcal{A}W_T}}{\mathcal{A}}\right]\right\} = \max_{x_{T-1}}\left\{\mathbb{E}_{Y_{T-1}}\left[\frac{-e^{-\mathcal{A}(x_{T-1}\cdot(Y_{T-1}-r)+W_{T-1}\cdot(1+r))}}{\mathcal{A}}\right]\right\}$$

Consider the minimizer of MGF of binomial distribution

$Z_t = \begin{cases} a-r, & probability\ of\ p \\ b-r, & probability\ of\ (1-p) \end{cases}$, we have

$$\min_{t\in\mathbb{R}} f_Z(t) = \min_{t\in\mathbb{R}} \mathbb{E}_Z[e^{tx}] = \min_{t\in\mathbb{R}} pe^{(a-r)t} + (1-p)e^{(b-r)t}$$

$$f_Z'(t) = (a-r)pe^{(a-r)t} + (1-p)(b-r)e^{(b-r)t}$$

We try to set $f_Z'(t) = 0$, we have

$(a-r)pe^{(a-r)t^*} + (1-p)(b-r)e^{(b-r)t^*} = 0$

$\Rightarrow (a-r)pe^{(a-r)t^*} = (p-1)(b-r)e^{(b-r)t^*}$

$$\Rightarrow \frac{(a-r)p}{(b-r)(p-1)} = e^{(b-a)t^*}$$

$$\Rightarrow t^* = \frac{\ln\left(\frac{(a-r)p}{(b-r)(p-1)}\right)}{b-a}$$

$f_Z''(t) = (a-r)^2pe^{(a-r)t} + (1-p)(b-r)^2e^{(b-r)t} > 0$ for all $t \in \mathbb{R}$ as $p \in [0,1]$

Thus, we have t^* as the minima. Substituting $t = t^*$ in $f_Z(t)$ yields us:

$$\min_{t\in\mathbb{R}} f_Z(t) = p\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{a-r}{b-a}} + (1-p)\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{b-r}{b-a}}$$

$$V_{T-1}^*(W_{T-1}) = \frac{-(p\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{a-r}{b-a}}+(1-p)\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{b-r}{b-a}})e^{-\mathcal{A}\cdot(1+\mathrm{r})\cdot W_{T-1}}}{\mathcal{A}}$$

Therefore, $c_{T-1} = \frac{p\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{a-r}{b-a}}+(1-p)\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{b-r}{b-a}})}{\mathcal{A}}$

$d_{T-1} = \mathcal{A}\cdot(1+\mathrm{r})$

The recursion gives us the general form for c_t and d_t for all $T \le T-2$ as:

$$c_t = \left(\frac{(p\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{a-r}{b-a}} + (1-p)\left(\frac{(a-r)p}{(r-b)(1-p)}\right)^{\frac{b-r}{b-a}})}{\mathcal{A}}\right)$$

$$\cdot (\left(\frac{p(a-r)}{(1-p)(r-b)}\right)^{\frac{r-b}{a-b}}\frac{(1-p)(a-b)}{a-r})^{T-1-t}$$

$$d_t = \mathcal{A}\cdot(1+\mathrm{r})^{\mathrm{T-t}}$$

Thus the solution for d_{t+1} gives us the solution for the optimal policy:

$$\pi_t^*(W_t) = x_t^* = \frac{1}{\mathcal{A}\cdot(1+\mathrm{r})^{\mathrm{T-t-1}}(a-b)}\ln(\frac{p(a-r)}{(1-p)(r-b)})$$

$$V_t^*(W_t) = -\left(\frac{-(p\left(\frac{(a-r)p}{(b-r)(p-1)}\right)^{\frac{a-r}{b-a}} + (1-p)\left(\frac{(a-r)p}{(b-r)(p-1)}\right)^{\frac{b-r}{b-a}})}{\mathcal{A}}\right)$$

$$\cdot (\left(\frac{p(a-r)}{(1-p)(r-b)}\right)^{\frac{r-b}{a-b}}(\frac{(1-p)(a-b)}{a-r}))^{T-1-t}\cdot e^{-\mathcal{A}\cdot(1+\mathrm{r})^{\mathrm{T-t}}\cdot W_t}$$

Substituting the solutions for b_{t+1} and c_{t+1} gives us the solution for the Optimal Action-Value Function:

$$Q_t^*(W_t) = -\left(\frac{-(p\left(\frac{(a-r)p}{(b-r)(p-1)}\right)^{\frac{a-r}{b-a}} + (1-p)\left(\frac{(a-r)p}{(b-r)(p-1)}\right)^{\frac{b-r}{b-a}})}{\mathcal{A}}\right)$$

$$\cdot (\left(\frac{p(a-r)}{(1-p)(r-b)}\right)^{\frac{r-b}{a-b}} \frac{(1-p)(a-b)}{a-r}) \qquad \cdot e^{-\mathcal{A}\cdot(1+\mathrm{r})\cdot(W_t\cdot(1+r)-r\cdot x_t+x_t\cdot b)}(1$$

$$-p+pe^{-\mathcal{A}\cdot(1+\mathrm{r})\cdot x_t(a-b)})^{T-1-t}$$

Reinforcement Learning:

Overview:

In the temporal-difference algorithms, the simplest 2 algorithms are Q-learning and SARSA algorithms.

The universal assumptions of the report are as stated below:

1. The risk averse governing factor is set as $\mathcal{A} \in [0,1]$, as the sensitivity is too low for $\mathcal{A} \geq 1$ given we have a 10-step process
2. Discrete state $S_t = (W_t, t)$ and action space instead of continuous state and action space
3. Short selling and borrowing are not permitted, i.e. $\frac{x_t}{W_t} \in [0,1]$

To generalize the results, we will fit our model into 3 scenarios w.r.t. optimal asset allocation x_t^* as in the Bellman close form solution

$$\begin{cases} \frac{x_t^*}{W_t} = 1, \ \ case\ 1\ (High\ Risk\ Allocation) \\ \frac{x_t^*}{W_t} = 0, \ \ case\ 2\ (Low\ Risk\ Allocation) \\ 0 < \frac{x_t^*}{W_t} < 1, \ \ case\ 3\ (Dynamic\ Risk\ Allocation) \end{cases}$$

Parameters	Probability of high return	High return	Low return	Risk-free rate	Risk averse
Case 1	90%	+10%	-1%	1%	0.2
Case 2	10%	+1.1%	-10%	1%	0.5
Case 3	70%	+30%	-30%	6%	0.4

The agent's performance and convergence were evaluated using:

1. Difference from Theoretical Optimal Allocation $|\frac{x_t^*}{W_t} - a_t|$

2. Final Utility

3. TD Error

Q-Learning

Q-Learning was employed to estimate the action-value function $Q(S_t, A_t)$, which represents the expected utility of selecting action A_t (allocation to the risky asset, $\frac{x_t}{W_t}$) in state S_t (current wealth index and time step) and following an optimal policy thereafter.

Environment Setup:

The environment models wealth dynamics over 10 steps, with:

1. A risk-free rate
2. A risky asset with binomial returns
3. $50 \cdot N$ Discretized wealth levels. (N is the maximum wealth possible)
4. 11 allocation actions over $[0,1]$.

Updating Policy:

Q-learning has the following formula for updating Q-values initially:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_A Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

We set the discount factor γ = 1.0, and $\alpha = 0.003$.

To handle the large state space (discretized wealth levels × time steps) and sparse rewards, a smoothing technique was introduced. The TD error is propagated to neighboring state-action pairs within a fixed window (e.g., ±5 wealth indices, ±1 action index), specifically targeting wealth indices and action indices, while the time index remains fixed. The state in this context is defined as $S_t = (W_t, t)$, but smoothing is applied only across wealth levels and actions, reflecting the assumption that nearby wealth levels and allocation actions have correlated Q-values, whereas the time step's effect is preserved exactly.

Rewriting the above Q-learning formula, we have:

$$Q((W_k, t), A_m) \leftarrow Q((W_k, t), A_m) + \alpha \cdot w_k \cdot w_m[R_{t+1} + \gamma \max_A Q((W_{t+1}, t+1), a) - Q((W_t, t), A_t)]$$

where w_k and w_m are Gaussian decay functions of wealth and action index respectively:

$$w_k = \begin{cases} \exp(-|W_t - W_k|^2 \cdot Decay_A), & k \neq t \\ 1, & k = t \end{cases}$$

$$w_m = \begin{cases} \exp(-|A_t - A_k|^2 \cdot Decay_B), & m \neq t \\ 1, & m = t \end{cases}$$

The $Decay_A$ and $Decay_B$ are a tuning rate of decay $\in [0, \infty)$. The decay rate is established at 0.1 for wealth indices and 0.8 for action indices, indicating a steeper decay for action indices compared to wealth indices.

This generalization, inspired by kernel-based RL, accelerates convergence by sharing information across similar states, compensating for the agent's limited exploration in a high-dimensional space.

Action Policy:

We adopted a conventional form of ϵ-greedy as below form:

$$\pi(A|S) = \begin{cases} arg\max_{A'} Q(S', A'), & probability\ 1-\epsilon \\ Uniform(A), & probability\ \epsilon \end{cases}$$

Where we set ϵ to be a linearly decaying function. Q-learning agent starts $\epsilon = 1$ and decays to 0.001 over 1 million episodes in *Case 1* and *Case 2*, while decays to 0.05 over 1.5 million episodes in *Case 3* to balance exploration and exploitation.

Results and Analysis:

Cases 1 and 2: High Risk Allocation and Low Risk Allocation

In *Case 1* (*Figure 1.1*), with an attractive risk-return profile, the theoretical optimal allocation is 100% risky, while in *Case 2* (*Figure 1.2*), it is 0% risky due to the negative expected return of the risky asset. The Q-Learning agent exhibits similar convergence patterns over 1 million episodes in both cases, with the difference from theoretical optimal allocation decreasing steadily from around 0.5 to below 0.1. The final utility in *Case 1* plateaus around 2.75, showing high returns from the risky asset's 10% expected return, while in *Case 2* it stabilizes near 1.05, consistent with the risk-free rate of 1%. TD error in both cases peaks early and declines to near 0, indicating stable learning. Therefore, the agent showed effective performance and convergence in trivial *Case 1* and *Case 2*.

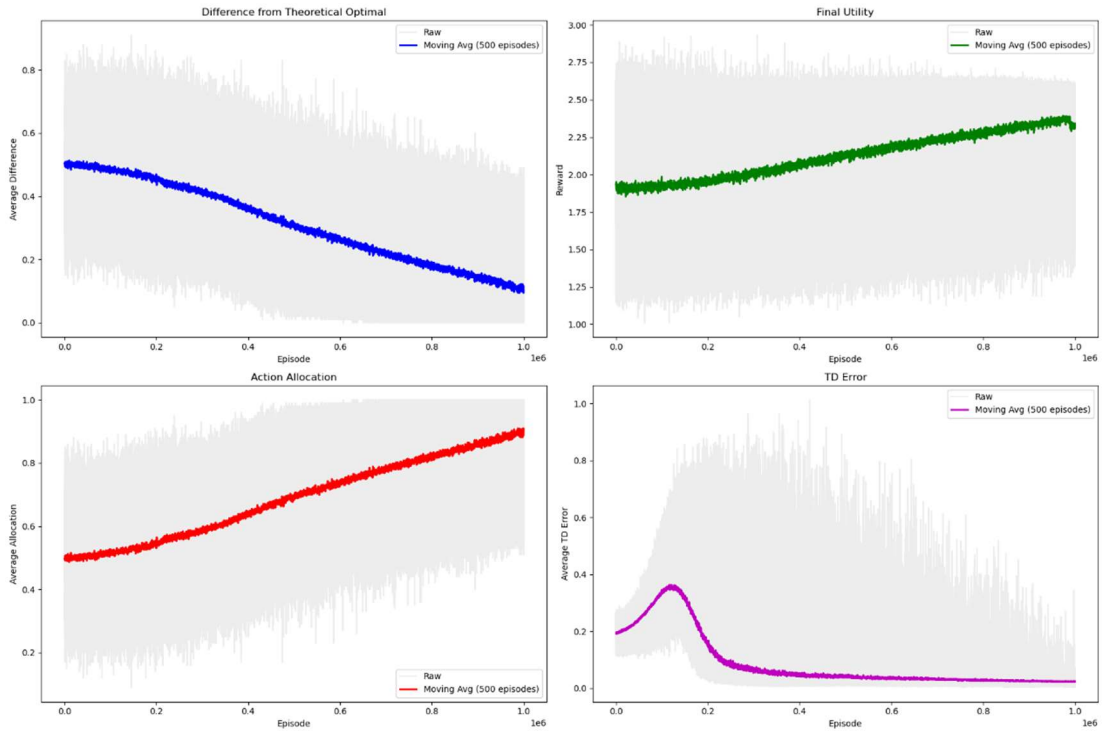

Figure 1.1. The convergence results of Q-Learning Agent in *Case 1 (High Risk Allocation)*.

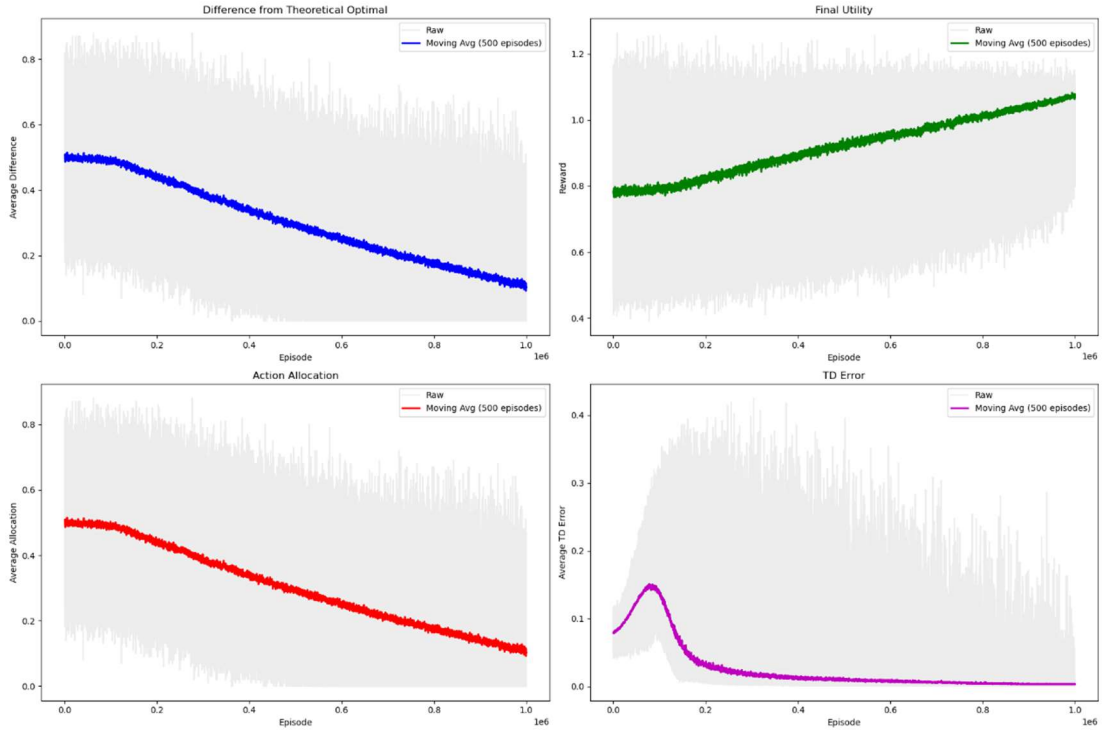

Figure 1.2. The convergence results of Q-Learning Agent in *Case 2 (Low Risk Allocation)*.

Case 3: Dynamic Risk Allocation

In *Case 3* (*Figure 1.3*), where the risky asset exhibits significant potential for both substantial gains and losses with a risk aversion of 0.5, the agent is designed to dynamically adjust its allocation strategy based on the current state to optimize the final utility. Over 1.5 million training episodes, the average deviation from the theoretical optimal allocation gradually decreased from approximately 0.35 to 0.25, while the TD error stabilized below 0.1, indicating robust learning, though convergence is less pronounced than in *Cases 1* and *2*. This deviation may be attributed to the limited sensitivity of the final utility to allocation changes, as a 0.1 reduction in the average difference from the theoretical optimum only increases the average final utility from around 1.74 to 1.76, hindering the agent's ability to identify the optimal allocation. To improve performance, incorporating intermediate utility signals through reward shaping may enhance the agent's sensitivity to dynamic adjustments.

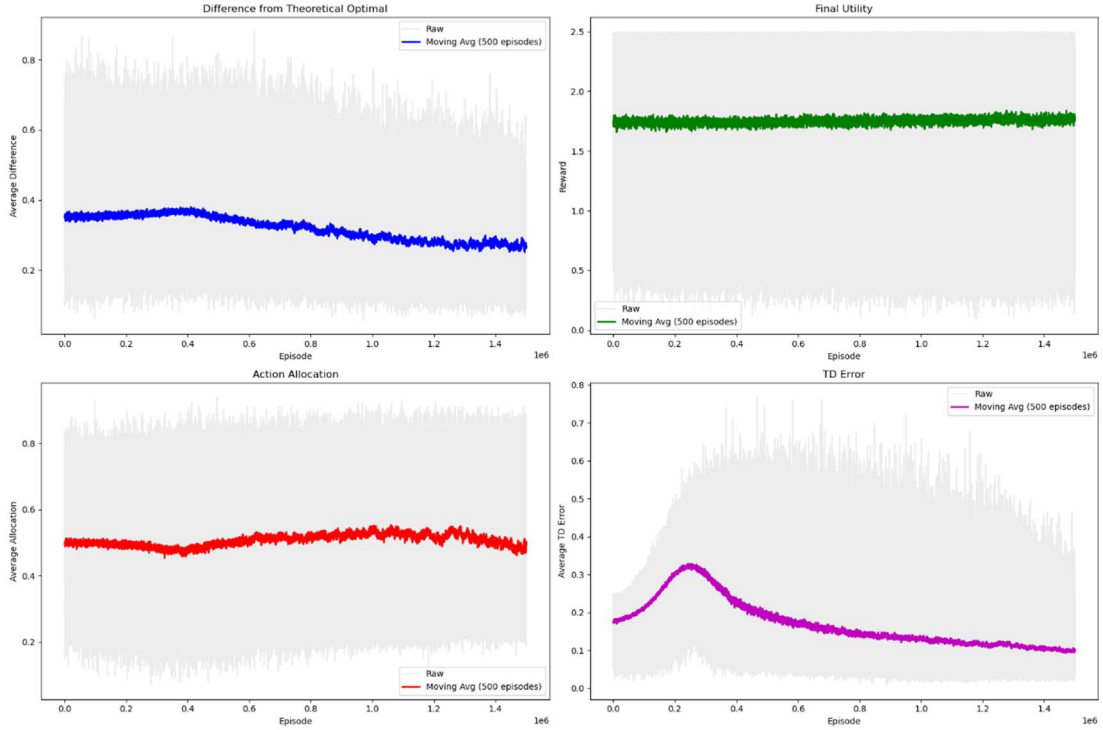

Figure 1.3. The convergence results of Q-Learning Agent in *Case 3 (Dynamic Risk Allocation)*.

SARSA

Our SARSA model adopted a discrete tabular method, which is maintaining a 3-dimensional table of Q-values $Q(t, W_t, A_t)$ and updating on number visit. We tried to limit our action and wealth space so that our training process can update every part of the Q-table. We also monitor the frequency of visits to the state denoted as $Visit(S, A)$.

Environment Setup:

The environment models wealth dynamics over 10 steps, with:

1. A risk-free rate
2. A risky asset with binomial returns
3. 25 Discretized wealth levels.
4. 21 allocation actions over $[0,1]$.

Action Policy:

We adopted an extension of ϵ-greedy as below form:

$$\pi(A|S) = \begin{cases} arg\max_{A'} Q(S, A'), & probability\ 0.2 + 0.6 \cdot (1 - \epsilon) \\ Uniform(A), & probability\ 0.4 \cdot \epsilon \\ \arg\min Visit(S', A'), & probability\ 0.4 \cdot \epsilon + 0.2 \cdot (1 - \epsilon) \end{cases}$$

Where we set ϵ to be a linearly decaying function from $1 \rightarrow 0.05$.

Updating Policy:

SARSA has the following formula for updating Q-values:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \sum_A \pi(A|S_{t+1}) Q(S_{t+1}, A) - Q(S_t, A_t)]$$

Reward Policy:

We followed the textbook's reward function but adding a multiplier as below:

$$R_t = \begin{cases} U(W_T) \cdot 100, & t = T \\ 0, & t \neq T \end{cases}$$

Since the reward is only distributed in the last step, we have the below setup facilitating the agent training process

- $\gamma = 0.99$ for transmission of last step reward to previous steps

- Adaptive $\alpha \propto \frac{1}{episodes\ trained} \cdot \frac{1}{Visit(s',a')}$ for encouraging trying out new states for better outcomes

Results and Analysis:

Due to super lengthy training time driven by our greedy-explore $\pi(a,s)$, we tried to limit the training episodes to 3 million iterations (despite not being ideal we do admit that), thus most results do show a sign of convergence but not to optimal stages.

Despite Slow convergence, we do see the agent does successfully learn from our metrics (Difference from theoretical optimal allocation, final utility and TD error). It is worth noting that theres a initial spike in the first 200,000 iterations, it is completely in line with our policy that encourages exploration and more frequently focused on states that we have visited fewer in the past. But as training goes by, the policy slowly goes to choosing the optimal action by design, thus we do see the result gradually converge.

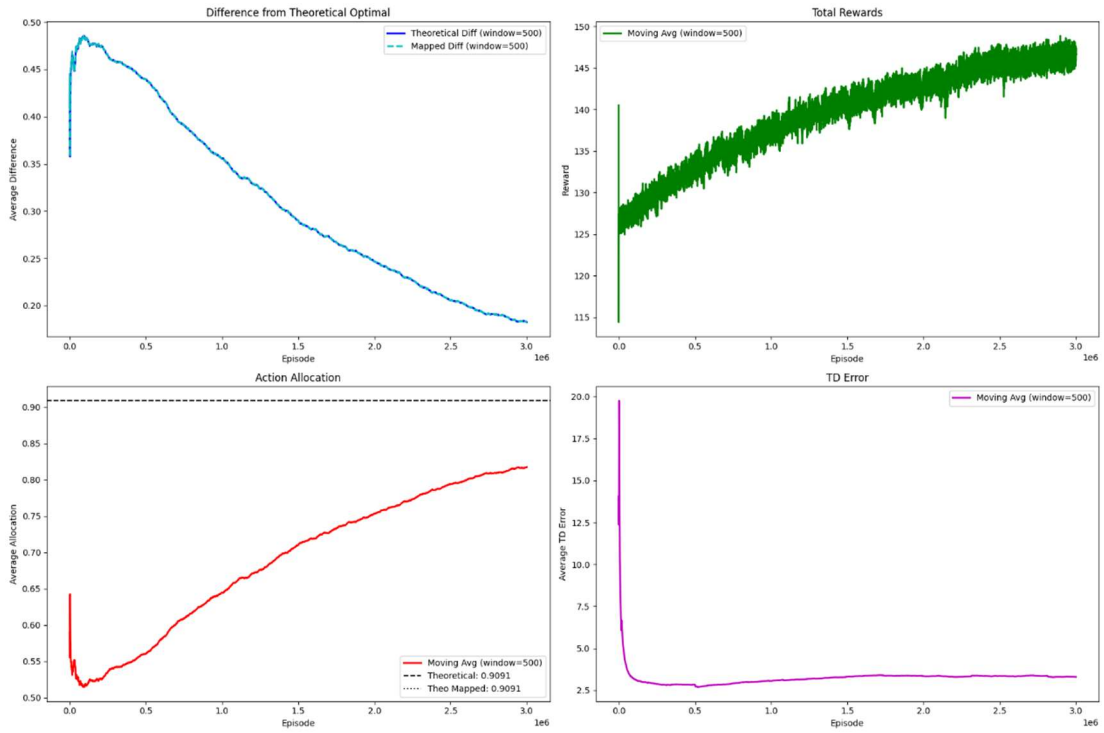

Figure 2.1. The convergence results of SARSA Agent in *Case 1 (High Risk Allocation)*.

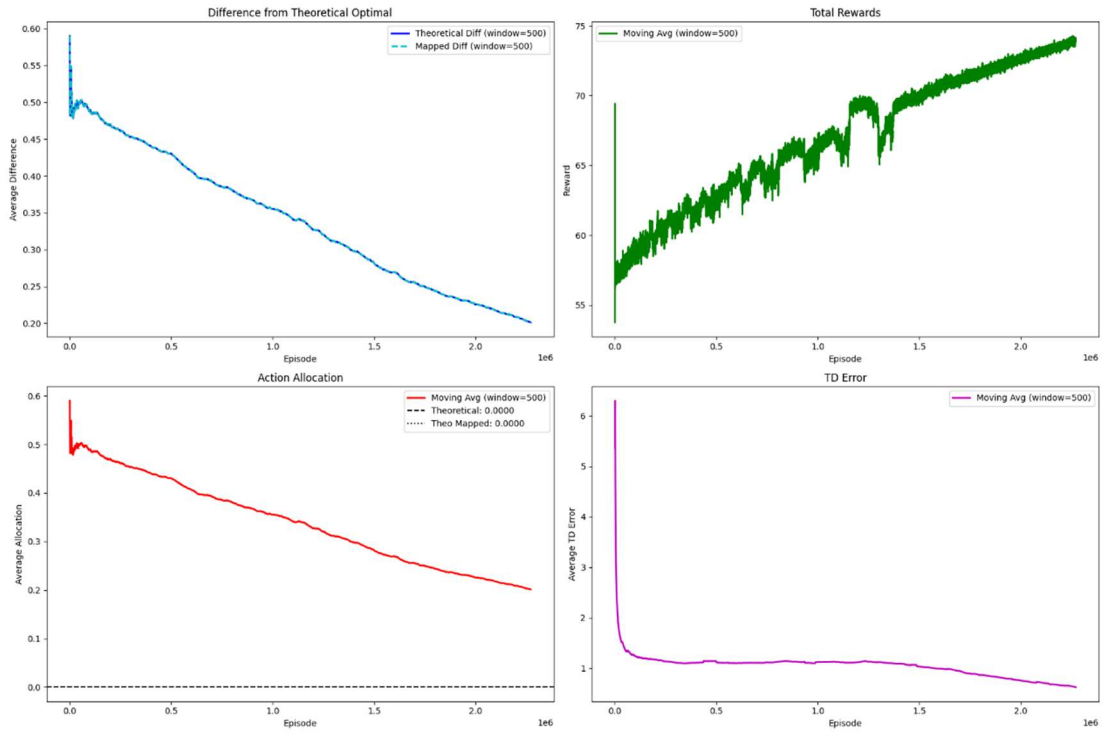

Figure 2.2. The convergence results of SARSA Agent in *Case 2 (Low Risk Allocation)*.

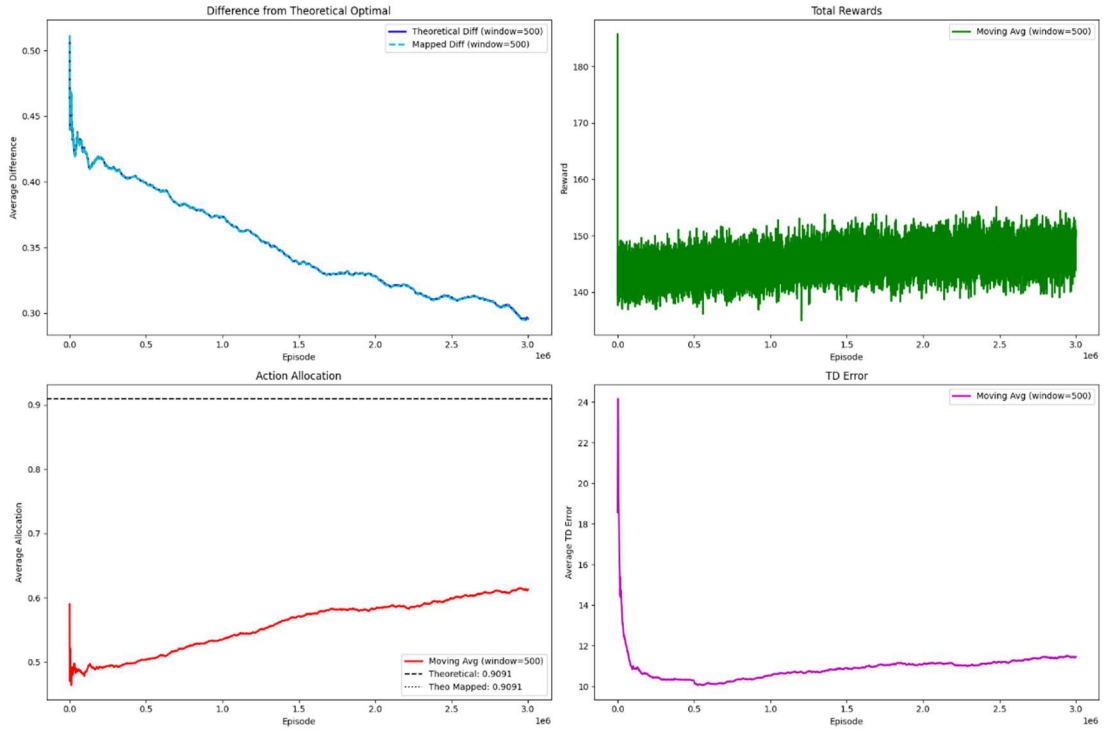

Figure 2.3. The convergence results of SARSA Agent in *Case 3 (Dynamic Risk Allocation)*.

Comparative Analysis:

Following an assessment of convergence performance during the training phase, the policies derived from Q-Learning and SARSA were subjected to testing in order to evaluate their efficacy in determining optimal allocations to risky assets and maximizing the expected final utility. Given that Case 3 necessitates dynamic risky allocation—thereby effectively demonstrating the capabilities of reinforcement learning (RL)—simulations comprising 10,000 wealth growth trajectories were conducted under the RL policies within the Case 3 framework. A summary of the resulting statistical analysis is presented below:

Policy	*Q-Learning*	*SARSA*	*100_risky*	*100_riskfree*	*50_50*	*Random*
Mean of final wealth	*2.4949*	*2.4319*	*3.0930*	*1.7908*	*2.3533*	*2.3592*
S.D. of final wealth	*1.1620*	*1.0496*	*2.779*	*0*	*0.9753*	*1.1471*
Mean of utility	*1.4782*	*1.4695*[*]	*1.4598*	*1.2787*	*1.4554*	*1.4334*

[*] *SARSA utility is boosted by 100 in SARSA environment, thus we manually convert back to the same scale*

In addition to the RL policies, it was observed that the static policy of allocating 100% to risky assets (*100_risky*) yielded the highest average utility among all static and random policies evaluated. Nonetheless, this policy merely outperformed the equal-weighting policy (*50_50*) by an increment of 0.0044 in average final utility, showing the relative insensitivity of utility to variations in actions.

Additionally, the RL policies, Q-Learning and SARSA, demonstrated superior performance by achieving the highest average final utilities of 1.4782 and 1.4695, respectively. This outcome underscores the capability of RL agents to identify optimal policies effectively, not only in straightforward scenarios such as *Case 1* and *Case 2*, but also in the more volatile market conditions represented by *Case 3*.

Summary:

This report evaluates Q-Learning and SARSA, two reinforcement learning (RL) models designed to optimize wealth allocation policies for maximizing expected final utility. Both models demonstrated convergence during training, evidenced by a decreasing gap between theoretical and agent-selected actions, diminishing temporal difference errors, and rising average utility. In simulated environments, particularly under volatile market conditions, these RL policies outperformed static and random strategies, achieving higher average final utilities, underscoring their adaptability and effectiveness.

However, challenges emerged, including difficulty converging to the theoretical optimum when utility was insensitive to actions and limitations imposed by finite training episodes. Potential enhancements involve extending the models to continuous state and action spaces, though this would require more advanced models, increased tuning, and longer training times. Overall, Q-Learning and SARSA exhibit strong potential for dynamic allocation tasks, with room for refinement to address these limitations and broaden their applicability.

Task Attribution:

Q-learning & Policy evaluation: Leung Ching Fung (21133252)

SARSA & Analytical Solution: Au Man Yi Sigmund (20504636)